Whoever Said Money Won't Solve All Your Problems?: Weighted Envy-free Allocation with Subsidy

Noga Klein Elmalem
The Open University of Israel

Haris, Aziz
UNSW Sydney

Rica, Gonen
The Open University of Israel

Huang, Xin
Kyushu University

他

出版情報：Computing Research Repository (CoRR). 2025-02, 2025-02. Cornell University
バージョン：

Whoever Said Money Won't Solve All Your Problems?
Weighted Envy-free Allocation with Subsidy*

Noga Klein Elmalem[1], Haris Aziz[3], Rica Gonen[1], Xin Huang[4], Kei Kimura[4], Indrajit Saha[4], Erel Segal-Halevi[2], Zhaohong Sun[4], Mashbat Suzuki[3], and Makoto Yokoo[4]

[1]The Open University of Israel
[2]Ariel University, Israel
[3]UNSW Sydney, Australia
[4] Kyushu University, Japan

March 7, 2025

Abstract

We explore solutions for fairly allocating indivisible items among agents assigned weights representing their entitlements. Our fairness goal is **weighted-envy-freeness (WEF)**, where each agent deems their allocated portion relative to their entitlement at least as favorable as any other's relative to their own. Often, achieving WEF necessitates monetary transfers, which can be modeled as third-party subsidies. The goal is to attain WEF with bounded subsidies.

Previous work relied on characterizations of unweighted envy-freeness (EF), that fail in the weighted setting. This makes our new setting challenging. We present polynomial-time algorithms that compute WEF allocations with a guaranteed upper bound on total subsidy for monotone valuations and various subclasses thereof.

We also present an efficient algorithm to compute a fair allocation of items and money, when the budget is not enough to make the allocation WEF. This algorithm is new even for the unweighted setting.

1 Introduction

The mathematical theory of fair item allocation among agents has practical applications, such as inheritance and partnership dissolutions. When agents have equal entitlements,

*This article combines two works (Aziz et al. (2024); Klein Elmalem et al. (2024)), both set to appear as extended abstracts at AAMAS.

each expects a share at least as good as others', known as an *envy-free (EF)* allocation. For indivisible items, an EF allocation might not exist. A common solution is using *money* to compensate for envy. Recent studies assume a hypothetical third-party provides a non-negative *subsidy* for each agent, and focus on minimizing the *total subsidy* needed for envy-freeness.

Halpern and Shah (2019) showed that for any allocation, there exists a permutation of bundles that is *envy-freeable (EF-able)*, meaning it can be made EF with subsidies. They proved the required subsidy is at most $(n-1)mV$, where m is the number of items, n the number of agents, and V the maximum item value, and this bound is tight when the allocation is given. Brustle et al. (2020) presented an algorithm using iterative maximum matching, that computes an EF-able allocation with subsidy at most $(n-1)V$, also tight. We extend the study to agents with different entitlements, or *weights*, as in partnership dissolutions, where agents hold varying numbers of shares. For example, an agent with twice the entitlement of another expects a bundle worth at least twice as much.

Formally, an allocation is *weighted envy-free (WEF)* (see e.g., Robertson and Webb (1998); Zeng (2000); Chakraborty et al. (2021a)) if, for any two agents i and j, $\frac{1}{w_i}$ times the utility i assigns to their own bundle is at least $\frac{1}{w_j}$ times the utility i assigns to j's bundle, where w_i and w_j are their entitlements.

We define *weighted envy-freeability (WEF-ability)* analogously to EF-ability: An allocation is WEF-able if it can be made WEF with subsidies. Specifically, for any two agents i and j, $\frac{1}{w_i}$ times the sum of the utility that i assigns to his own bundle and the subsidy he receives is at least as high as $\frac{1}{w_j}$ times the sum of the utility that i assigns to the bundle of j and the subsidy j receives. Here, we assume quasi-linear utilities.

To illustrate the challenges in the generalized setting of unequal entitlements, we demonstrate that the results from Halpern and Shah (2019); Brustle et al. (2020) for additive valuations fail when agents have different entitlements.

Example 1.1 (No permutation of bundles is WEF-able)**.** There are two items o_1, o_2 and two agents i_1, i_2, with weights $w_1 = 1, w_2 = 10$ and valuation functions

$$\begin{bmatrix} & o_1 & o_2 \\ i_1 & 5 & 7 \\ i_2 & 10 & 8 \end{bmatrix}$$

Consider the bundles $X_1 = \{o_1\}$ and $X_2 = \{o_2\}$, where i_1 receives X_1 and i_2 receives X_2. Let p_1 and p_2 represent the subsidies for i_1 and i_2, respectively. The utility of i_1 for their own bundle is $5 + p_1$, and for i_2's bundle, it is $7 + p_2$. To satisfy WEF, we need: $\frac{5+p_1}{1} \geq \frac{7+p_2}{10}$, which implies $p_2 \leq 43 + 10p_1$. Similarly, for agent i_2, WEF requires: $\frac{8+p_2}{10} \geq \frac{10+p_1}{1}$, which implies $p_2 \geq 92 + 10p_1$. These two conditions are contradictory, so no subsidies can make this allocation WEF. Next, consider the permutation where i_1

receives X_2 and i_2 receives X_1. In this case, WEF requires $\frac{7+p_1}{1} \geq \frac{5+p_2}{10}$, which implies $p_2 \leq 65 + 10p_1$, and for i_2: $\frac{10+p_2}{10} \geq \frac{8+p_1}{1}$, which implies $p_2 \geq 70 + 10p_1$. Again, these conditions are contradictory, proving that no permutation of bundles is WEF-able.

Example 1.1 also implies that the *Iterated Maximum Matching algorithm* (Brustle et al. (2020)) does not guarantee WEF, as that algorithm yields an allocation where all agents receive the same number of items.

When valuations are not additive, even more results from the unweighted setting fail to hold.

Example 1.2 (Welfare-maximizing allocation is not WEF-able)**.** There are two agents with weights $w_1 = 1, w_2 = 3$. There are two identical items. The agents have unit demand: agent 1 values any bundle with at least one item at 30, and agent 2 at 90. We show that, contrary to the result in Halpern and Shah (2019), the allocation maximizing the social welfare (sum of utilities) is not WEF-able.

The social welfare is maximized by allocating one item to each agent. Note that this is also the only allocation that is non-wasteful (Definition 2.3). WEF requires $\frac{30+p_1}{1} \geq \frac{30+p_2}{3}$, which implies $p_2 \leq 3p_1 + 60$; and $\frac{90+p_2}{3} \geq \frac{90+p_1}{1}$, which implies $p_2 \geq 3p_1 + 180$ — a contradiction.

These examples demonstrate that the weighted case is more challenging than the unweighted case and requires new ideas.

Of course, since the unweighted case is equivalent to the weighted case where each weight $w_i = 1/n$, all negative results from the unweighted setting extend to the weighted case. In particular, it is NP-hard to compute the minimum subsidy required to achieve (weighted) envy-freeness, even in the binary additive case, assuming the allocation is non-wasteful (as shown in (Halpern and Shah, 2019, Corollary 1)). Thus, following previous work, we develop polynomial-time algorithms that, while not necessarily optimal, guarantee an upper bound on the total subsidy.

1.1 Our Results

Our subsidy bounds for achieving WEF allocations are expressed as functions of n (number of agents), m (number of items), $w_{\min}$ (smallest agent weight), and V (maximum item value); see Section 2 for definitions.

For *general monotone valuations*, we show that a total subsidy of $\left(\frac{W}{w_{\min}} - 1\right) mV$ is sufficient for WEF, and that this bound is tight in the worst case (Section 3).

For *supermodular* and *superadditive* valuations, we further show that WEF can be attained simultaneously with maximizing the social welfare and truthfulness (Section 4).

For *additive valuations*, assuming all weights are integers, we further show an upper bound that is independent of m: it is $\frac{W-w_{\min}}{\gcd(\mathbf{w})}V$, where $\gcd(\mathbf{w})$ is the greatest common divisor of all weights — largest number d such that w_i/d is an integer for all $i \in N$ (Section 5). Our algorithm extends the one in Brustle et al. (2020); when all entitlements are equal it guarantees the same upper bound $(n-1)V$.

We also study WEF relaxations introduced for the setting without subsidy: WEF(x, y) Chakraborty et al. (2022) and WWEF1 Chakraborty et al. (2021b). We prove that WEF-ability is incompatible with WWEF1 or with WEF(x, y) for $x + y < 2$, but show an algorithm for two agents that finds a WEF-able and WEF$(1, 1)$ allocation (Section 5.1).

For *identical additive valuations* (Section 6), we compute a WEF-able and WEF$(0, 1)$ allocation with total subsidy at most $(n-1)V$, which is tight even in the unweighted case.

For *binary additive valuations*, we adapt the General Yankee Swap algorithm (Viswanathan and Zick (2023a)) to compute a WEF-able and WEF$(0, 1)$ allocation with total subsidy at most $\frac{W}{w_{\min}} - 1$, reducing to $n-1$ for equal weights (Section 7). For matroidal valuations, we show a linear lower bound in m (Section 7.1).

For identical items, we derive an almost tight bound of $V \sum_{2 \leq i \leq n} \left(w_i \sum_{1 \leq j \leq i} \frac{1}{w_j} \right)$, where agents are sorted by descending order of their value for a single-item (Section 8). In particular, with nearly equal but different weights, the required subsidy may be in $\Omega(n^2 V)$, unlike the $O(nV)$ bound for equal weights. Additionally, we present a polynomial-time algorithm for computing a WEF-able allocation that requires the smallest possible amount of subsidy for each specific instance. This is in contrast to the other sub-cases of additive valuations (binary additive and identical additive), in which computing the minimum subsidy per instance is known to be NP-hard.

Finally, we address scenarios with limited subsidy budgets (e.g., leftover cash in partnership dissolutions), exploring relaxed fairness under subsidy constraints (Section 9).

Our contributions are summarized in Table 1.

In Section 10, we present preliminary experiments on the required subsidy in random instances and compare our algorithms with theoretical bounds.

1.2 Related Work

Equal entitlements. Steinhaus (1948) initiated fair allocation with the cake-cutting problem, followed by Foley (1966) advocacy for envy-free resource allocation. Challenges with indivisible items were outlined by Schmeidler and Yaari (1971).

Fair allocation with Monetary Transfers. The concept of compensating an indivisible resource allocation with money has been explored in the literature ever since Demange et al. (1986) introduced an ascending auction for envy-free allocation using monetary payments for unit demand agents.

Recently, the topic of multi-demand fair division with subsidies has attracted significant attention (see, e.g., a survey article Liu et al. (2024)). Halpern and Shah (2019) showed

Table 1: Upper and lower bounds on worst-case total subsidy in weighted envy-freeable allocations. All subsidy upper bounds are attainable by polynomial-time algorithms. Here, w_2 represents the second-smallest weight.

Valuation	Lower bound	Upper bound
General, superadditive, supermodular	$\left(\frac{W}{w_{\min}} - 1\right) mV$ [Theorem 3.8]	$\left(\frac{W}{w_{\min}} - 1\right) mV$ [Theorem 3.8]
Additive	$\left(\frac{W}{w_{\min}} - 1\right) V$ [Theorem 5.2]	$\frac{W - w_{\min}}{\gcd(\mathbf{w})} V$ [Lemma 5.10]
Identical additive	$(n-1)V$ [Theorem 6.1]	$(n-1)V$ [Theorem 6.6]
Binary additive	$\frac{W}{w_2} - 1$ [Proposition 7.1]	$\frac{W}{w_{\min}} - 1$ [Theorem 7.12]
Matroidal	$\frac{m}{n}\left(\frac{W}{w_{\min}} - n\right)$ [Theorem 7.14]	$\left(\frac{W}{w_{\min}} - 1\right) m$ [Theorem 3.8]
Additive, identical items	$\sum_{2 \le i \le n} \left(V w_i \sum_{1 \le j < i} \frac{1}{w_j} \right)$ [Theorem 8.2]	$\sum_{2 \le i \le n} \left(V w_i \sum_{1 \le j \le i} \frac{1}{w_j} \right)$ [Theorem 8.4]

that an allocation is envy-freeable with money if and only if the agents cannot increase social welfare by permuting bundles. Brustle et al. (2020) study the more general class of monotone valuations. They demonstrate that a total subsidy of $2(n-1)^2V$ is sufficient to guarantee the existence of an envy-freeable allocation. Moreover, they showed that for additive valuations where the value of each item is at most V, giving at most V to each agent (i.e., a total subsidy of at most $(n-1)V$) is sufficient to eliminate envies. Kawase et al. (2024) improved this bound to $\frac{n^2-n-1}{2}$.

Caragiannis and Ioannidis (2021) developed an algorithm that approximates the minimum subsidies with any required accuracy for a constant number of agents, though with increased running time. However, for a super-constant number of agents, they showed that minimizing subsidies for envy-freeness is both hard to compute exactly and difficult to approximate.

Aziz (2021) presented a sufficient condition and an algorithm to achieve envy-freeness and equitability (every agent should get the same utility) when monetary transfers are allowed for agents with quasi-linear utilities and superadditive valuations (positive or negative).

Barman et al. (2022) studied agents with dichotomous valuations (agents whose marginal value for any good is either zero or one), without any additivity requirement. They proved that, for n agents, there exists an allocation that achieves envy-freeness with total required subsidy of at most $n - 1$, which is tight even for additive valuations.

Goko et al. (2024) study an algorithm for an envy-free allocation with subsidy, that is also *truthful*, when agents have submodular *binary* valuations. The subsidy per agent is at most $V = 1$. Their algorithm works only for agents with equal entitlements.

The case where multiple items can be allocated to each agent while the agents pay some amount of money to the mechanism designer, is extensively studied in combinatorial auctions (Cramton et al., 2006). A representative mechanism is the well-known Vickrey-Clarke-Groves (VCG) mechanism (Clarke (1971); Groves (1973); Vickrey (1961)), which is truthful and maximizes social welfare. Envy-freeness is not a central issue in combinatorial auctions, with a notable exception presented by Pápai (2003).

Different entitlements. In the past few years, several researchers have examined a more general model in which different agents may have different entitlements, included weighted fairness models like *weighted maximin share fairness (WMMS)* and *weighted proportionality up to one item (WPROP1)* (Chakraborty et al. (2021b); Babaioff et al. (2024); Aziz et al. (2019)). Chakraborty et al. (2021a) established *maximum weighted Nash welfare (MWNW)* satisfies *Pareto optimality* and introduced a weighted extension of EF1. Suksompong and Teh (2022) demonstrated MWNW properties under binary additive valuations and its polynomial-time computability. They further extended these findings to various valuation types.

Different entitlements with subsidies. There is a long-standing tradition in fair division to revisit settings and extend them to the case of weighted entitlements. In a classic book by Brams and Taylor (1996), many algorithms and results are extended to the cases of weighted entitlements. This tradition continues in the context of the allocation of indivisible items. Wu et al. (2023) presented a polynomial-time algorithm for computing a PROP allocation of *chores* among agents with additive valuations, with total subsidy at most $\frac{nV}{4}$, which is tight. For agents with different entitlements, they compute a WPROP allocation with total subsidy at most $\frac{(n-1)V}{2}$. In a subsequent work (Wu and Zhou (2024)), they further improved this bound to $(\frac{n}{3} - \frac{1}{6})V$.

As far as we know, weighted envy-freeness with subsides has not been studied yet. Our paper aims to fill this gap.

Matroid-rank valuations. Recent studies have considered *matroid-rank* valuation (binary submodular). Montanari et al. (2024) introduce a new family of weighted envy-freeness notions based on the concept of *transferability* and provide an algorithm for computing transferable allocations that maximize welfare. Babaioff et al. (2021) design truthful allocation mechanisms that maximize welfare and are fair. Particularly relevant to our work is a recent work by Viswanathan and Zick (2023b), who devised a fair allocation method inspired by Yankee Swap, achieving efficient and fair allocations when agents have submodular binary valuations. We use some of their techniques in our algorithms. Later, Viswanathan and Zick (2023a) generalize the Yankee Swap algorithm to efficiently compute allocations that maximize any fairness objective, called General Yankee Swap.

2 Model

Agents and valuations. Let $[t] = \{1, 2, \dots, t\}$ for any positive integer t. We consider n agents in N and m items in M. Each agent $i \in N$ has a valuation function $v_i : 2^M \to \mathbb{R}_0^+$, specifying a non-negative real value $v_i(A)$ for a given bundle $A \subseteq M$. We write $v_i(o_1, ..., o_t)$ instead of $v_i(\{o_1, ..., o_t\})$.

We assume that the valuation functions v_i are *normalized*, that is $v_i(\emptyset) = 0$ for all $i \in N$ (Suksompong and Teh (2023)), and *monotone*, i.e., for each $i \in N$ and $A \subseteq B \subseteq M$, $v_i(A) \leq v_i(B)$. We denote $V := \max\limits_{i \in N, A \subseteq M, A \neq \emptyset} \dfrac{v_i(A)}{|A|}$. Note that when the valuations are additive, V equals to $\max\limits_{i \in N, o \in M} v_i(o)$.[1] Also, $\max\limits_{i \in N} v_i(M) \leq m \cdot V$.

An *allocation* $X = (X_1, \dots, X_n)$ is a partitioning of the items into n bundles where X_i is the bundle allocated to agent i. We assume allocation X must be *complete*, i.e., $\bigcup_{i \in N} X_i = M$ holds;

An *outcome* is a pair consisting of the allocation and the subsidies received by the agents, that is, a pair $(X, \mathbf{p})$ is the allocation that specifies bundle $X_i \subseteq M$ for agent i and $\mathbf{p} \in (\mathbb{R}_0^+)^n$ specifies the subsidy p_i received by agent i.

An agent i's *utility* for a bundle-subsidy pair (X_j, p_j) is $v_i(X_j) + p_j$. In other words, we assume quasi-linear utilities.

[1] In previous work it was assumed that $V = 1$.

Envy. An outcome $(X, \mathbf{p})$ is *envy-free (EF)* if for all $i, j \in N$, it holds that

$$v_i(X_i) + p_i \geq v_i(X_j) + p_j.$$

An allocation X is *envy-freeable (EF-able)* if there exists a subsidy vector $\mathbf{p}$ such that $(X, \mathbf{p})$ is EF.

Entitlements. Each agent $i \in N$ is endowed with a fixed *entitlement* $w_i \in \mathbb{R}_{>0}$. We also refer to entitlement as *weight*. We assume, without loss of generality, that the entitlements are ordered in increasing order, i.e., $w_{\min} = w_1 \leq w_2 \leq \ldots \leq w_n = w_{\max}$. We denote $W := \sum_{i \in N} w_i$.

Definition 2.1 (Weighted envy-freeability)**.** An outcome $(X, \mathbf{p})$ is *weighted envy-free (WEF)* if for all $i, j \in N$:

$$\frac{v_i(X_i) + p_i}{w_i} \geq \frac{v_i(X_j) + p_j}{w_j}.$$

An allocation X is *weighted envy-freeable (WEF-able)* if there is subsidy vector $\mathbf{p}$, such that $(X, \mathbf{p})$ is WEF.

The term $\frac{v_i(X_i)}{w_i}$ represents the value per unit entitlement for agent i in their allocation. The WEF condition ensures that this value is at least as high as $\frac{v_i(X_j)}{w_j}$, denoting the corresponding value per unit entitlement for agent j in the same allocation. WEF seamlessly reduces to envy-free (EF) concept when entitlements are equal, i.e., $w_i = w_j$ for all $i, j \in N$.

Efficiency concepts.

Definition 2.2 (Pareto efficiency)**.** We say allocation X dominates another allocation X' if $\forall i \in N, v_i(X_i) \geq v_i(X'_i)$ and $\exists j \in N$, $v_j(X_j) > v_j(X'_j)$ hold. We say X is Pareto efficient if it is not dominated by any other allocation.

Definition 2.3 (Non-wastefulness)**.** We say allocation X is *non-wasteful* if $\forall i \in N$, $\forall o \in X_i$, if $v_i(X_i) = v_i(X_i \setminus \{o\})$ holds, then $v_j(X_j \cup \{o\}) = v_j(X_j)$ for all $j \neq i$.

In other words, no item can be transferred from one agent to another, and the transfer results in a Pareto improvement.

For $S \subseteq N$ and $A \subseteq M$, let $\mathcal{X}^{S,A}$ denote all possible allocations of A among S. For an allocation X and a subset $S \subseteq N$ of agents, the *social welfare* among the agents in S is defined as $SW^S(X) := \sum_{i \in S} v_i(X_i)$.

Definition 2.4 (Maximizing social welfare allocation (MSW))**.** We say allocation $X^{S,A} \in \mathcal{X}^{S,A}$ maximizing social welfare with respect to S and A ($MSW^{S,A}$) if $SW^S(X^{S,A}) \geq SW^S(X)$ holds for any $X \in \mathcal{X}^{S,A}$ [2].

Any MSW allocation is Pareto efficient, but not vice versa.

Weighted envy-freeability and non-wastefulness are generally incompatible (Example 1.2). To address this, we propose a weaker efficiency property:

Definition 2.5 (Non-zero social welfare)**.** We say allocation X satisfies non-zero social welfare property if $SW^N(X) = 0$, then for any other allocation X', $SW^N(X') = 0$ holds.

This property allows choosing an allocation X such that $SW^N(X) = 0$ only if social welfare is zero for all allocations.

Pareto efficiency implies both non-wastefulness and non-zero social welfare, but the reverse is not true. Non-wastefulness and non-zero social welfare are independent properties.

3 Characterization for General Monotone Valuations

In this section we allow agents to have arbitrary monotone valuations. We give a characterization of WEF-able allocations. For the characterization, we generalize a couple of previously studied mathematical objects to the weighted case.

Definition 3.1 (Weighted reassignment-stability)**.** We say that an allocation X is *weighted reassignment-stable* if

$$\sum_{i \in N} \frac{v_i(X_i)}{w_i} \geq \sum_{i \in N} \frac{v_i(X_{\pi(i)})}{w_{\pi(i)}} \tag{1}$$

for all permutations π of N.

With equal weights, reassignment-stability implies that X maximizes the sum of utilities. But with different weights, it does *not* imply that X maximizes any function.

Definition 3.2 (Weighted envy-graph)**.** For any given allocation X, the corresponding *weighted envy-graph*, $G_{X,w}$, is a complete directed graph with vertex set N, each assigned a weight w_i.

For any pair of agents $i, j \in N$, $cost_X(i, j)$ is the cost of edge (i, j) in $G_{X,w}$, which presents the fact that agent i has envy toward agent j under the allocation X:

$$cost_X(i,j) \;=\; \frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i}.$$

[2]If $S = N$ and $A = M$, we omit "with respect to N and M" and just say a maximizing social welfare (MSW) allocation.

Note that $cost_X$ can take negative values. For any path or cycle C in the graph, $cost_X(C)$ is the cost of the C under allocation X, which is the sum of costs of edges along C in $G_{X,w}$.

With these definitions, $\ell_{i,j}(X)$ represents the cost of the maximum-cost path from i to j in $G_{X,w}$, and $\ell_i(X) = cost_X(P_i(X))$ represents cost of the maximum-cost path in $G_{X,w}$ starting at i, denoted as $P_i(X)$.

Similar to the previous work of Halpern and Shah (2019) (in the unweighted setup), we provide necessary and sufficient conditions for a WEF-able allocation:

Theorem 3.3. *The following are equivalent for allocation X:*

1. *X is WEF-able;*
2. *X is weighted reassignment-stable;*
3. *The graph $G_{X,w}$ has no positive-cost cycle.*

Proof. $1 \Rightarrow 2$. Suppose the allocation X is WEF-able. Then, there exists a subsidy vector $\mathbf{p}$ such that for all agents i, j $\frac{v_i(X_i)+p_i}{w_i} \geq \frac{v_i(X_j)+p_j}{w_j}$. Equivalently, $\frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i} \leq \frac{p_i}{w_i} - \frac{p_j}{w_j}$. Consider any permutation π of N. Then,

$$\sum_{i \in N} \left(\frac{v_i(X_{\pi(i)})}{w_{\pi(i)}} - \frac{v_i(X_i)}{w_i} \right) \leq \sum_{i \in N} \left(\frac{p_i}{w_i} - \frac{p_{\pi(i)}}{w_{\pi(i)}} \right) = 0.$$

The last entry is zero as all the weighted subsidies are considered twice, and they cancel out each other. Hence the allocation X is weighted reassignment-stable.

$2 \Rightarrow 3$. Suppose some allocation X has a corresponding weighted envy-graph with a cycle $C = (i_1, \ldots, i_r)$ of strictly positive cost. Then consider a permutation π, defined for each agent $i_k \in N$ as follows:

$$\pi(i_k) = \begin{cases} i_k, & i_k \notin C \\ i_{k+1}, & k \in \{1, \ldots, r-1\} \\ i_1 & k = r \end{cases}.$$

In that case $0 < cost_X(C) \Leftrightarrow \sum_{i \in N} \frac{v_i(X_i)}{w_i} < \sum_{i \in N} \frac{v_i(X_{\pi(i)})}{w_{\pi(i)}}$, which means that X is not weighted reassignment-stable.

$3 \Rightarrow 1$. Suppose (3) holds. As there are no positive-weight cycles in $G_{X,w}$, we can define, for each agent i, the maximum cost of any path in the weighted envy-graph that starts at i. We denote this path by $\ell_i(X)$. Let each agent i's subsidy be $p_i = \ell_i(X) \cdot w_i$. Then for any other agent $j \neq i \in N$, $\frac{p_i}{w_i} = \ell_i(X) \geq cost_X(i,j) + \ell_j(X) = \frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i} + \frac{p_j}{w_j}$. This implies that $(X, \mathbf{p})$ is WEF, and hence X is WEF-able. ∎

Theorem 3.3 presents an effective method for verifying whether a given allocation is WEF-able.

Proposition 3.4. *Given an allocation X, it can be checked in polynomial time whether X is WEF-able.*

Proof. According to Theorem 3.3, determining whether X is is WEF-able is equivalent to verifying if X is weighted reassignment-stable. To analyze this, consider a complete bipartite graph G with N nodes on each side. The weight of each edge connecting i on the left to j on the right is defined as $\frac{v_i(X_j)}{w_j}$. This graph can be constructed in $O(mn+n^2)$ time. Next, the maximum-value matching for G can be computed in $O(n^3)$ time (Edmonds and Karp (1972)). If the value of the matching that contains edge (i,i) for all $i \in N$ is equal to the maximum value, it follows that $\sum_{i \in N} \frac{v_i(X_i)}{w_i} \geq \sum_{i \in N} \frac{v_i(X_\pi(i))}{w_{\pi(i)}}$ for any permutation $\pi : N \to N$. This condition aligns with the definition of weighted reassignment-stability. The total running time is $O(mn + n^3)$.

Another approach involves verifying the absence of positive-cost cycles in the weighted envy-graph $G_{X,w}$. This can be achieved by transforming the graph by negating all edge weights and then checking for the presence of negative-cost cycles. Using the Floyd-Warshall algorithm (Weisstein (2008); Wimmer and Lammich (2017)) on the graph obtained by negating all edge cost in $G_{X,w}$ requires $O(n^3)$ time. Constructing the initial graph $G_{X,w}$ takes $O(mn)$ time, resulting in an overall complexity of $O(mn + n^3)$. □

Deciding that an allocation X is WEF-able is not enough. we also need to find a minimal subsidy vector, $\mathbf{p}$, that ensures WEF for $(X, \mathbf{p})$. The following theorem is similar to (Halpern and Shah, 2019, Theorem 2), which states the minimum subsidy required when given a WEF-able allocation.

Theorem 3.5. *For any WEF-able allocation X, let $\mathbf{p}^*$ be a subsidy vector defined by $p_i^* := w_i \ell_i(X)$, for all $i \in N$. Then*

1. *$(X, \mathbf{p}^*)$ is WEF;*
2. *Any other subsidy vector $\mathbf{p}$, such that $(X, \mathbf{p})$ is WEF, satisfies $p_i^* \leq p_i$ for all $i \in N$;*
3. *$\mathbf{p}^*$ can be computed in $O(nm + n^3)$ time;*
4. *There exists at least one agent $i \in N$ for whom $p_i^* = 0$.*

Proof. 1. The establishment of condition 3 implying condition 1 in Theorem 3.3 has already demonstrated that $(X, \mathbf{p}^*)$ is WEF.

2. Let $\mathbf{p}$ be a subsidy vector, such that $(X, \mathbf{p})$ is WEF, and $i \in N$ be fixed. Consider the highest-cost path originating from i in the graph $G_{X,w}$, $P_i(X) = (i = i_1, ..., i_r)$, with $cost_X(P_i(X)) = \ell_i(X) = \frac{p_i^*}{w_i}$.

   Due to the WEF nature of $(X, \mathbf{p})$, it follows that for each $k \in [r-1]$, the following inequality holds:

$$\frac{v_{i_k}(X_{i_k}) + p_{i_k}}{w_{i_k}} \geq \frac{v_{i_k}(X_{i_{k+1}}) + p_{i_{k+1}}}{w_{i_{k+1}}} \Rightarrow \frac{p_{i_k}}{w_{i_k}} - \frac{p_{i_{k+1}}}{w_{i_{k+1}}} \geq \frac{v_{i_k}(X_{i_{k+1}})}{w_{i_{k+1}}} - \frac{v_{i_k}(X_{i_k})}{w_{i_k}} = cost_X(i_k, i_{k+1}).$$

   Summing this inequality over all $k \in [r-1]$, the following relation is obtained:

$$\frac{p_i}{w_i} - \frac{p_{i_r}}{w_{i_r}} = \frac{p_{i_1}}{w_{i_1}} - \frac{p_{i_r}}{w_{i_r}} \geq cost_X(P_i(X)) = \frac{p_i^*}{w_i} \Rightarrow \frac{p_i}{w_i} \geq \frac{p_i^*}{w_i} + \frac{p_{i_r}}{w_{i_r}} \geq \frac{p_i^*}{w_i}.$$

   The final transition is valid due to the non-negativity of subsidies and weights, that is, $\frac{p_{i_r}}{w_{i_r}} \geq 0$.

3. The computation of $\mathbf{p}^*$ can be executed as follows: Initially, the Floyd-Marshall algorithm (Weisstein (2008), Wimmer and Lammich (2017)) is applied to the graph derived by negating all edge costs in $G_{X,w}$ (This has a linear time solution since there are no cycles with positive costs in the graph). Hence, determining the longest path cost between any two agents, accomplished in $O(nm + n^3)$ time. Subsequently, the longest path starting at each agent is identified in $O(n^2)$ time.

4. Assume, for the sake of contradiction, that $p_i^* > 0$ for every agent $i \in N$. This implies that $\ell_i(X) > 0$, since $w_i > 0$. Starting from an arbitrary agent i_1, we trace the highest-cost path starting at i_1 and arrive at some agent i_2; then we trace the highest-cost path starting at i_2 and arrive at some agent i_3; and so on. Because the number of agents is finite, eventually we will arrive at an agent we already visited. We will then have a cycle with positive cost in $G_{X,w}$, contradicting Theorem 3.3.

   Therefore, there must be at least one agent whose subsidy under $\mathbf{p}^*$ is 0.

□

Now we can find the minimum subsidy needed in the worst-case scenario for agents with different entitlements, whether the allocation is given or can be chosen.

Theorem 3.6. *For every weight vector and every given WEF-able allocation X, letting $p_i := p_i^* = w_i \ell_i(X)$, the total subsidy $\sum_{i \in N} p_i$ is at most $\left(\frac{W}{w_{\min}} - 1\right) mV$. This bound is tight in the worst case.*

Proof. The proofs extend those of Halpern and Shah (2019). By Theorem 3.5, to bound the subsidy required for i, we bound the highest cost of a path starting at i. We prove that, for every WEF-able allocation X and agent i, the highest cost of a path from i in $G_{X,w}$ is at most $\frac{mV}{w_{\min}}$.

For every path P in $G_{X,w}$,

$$cost_X(P) = \sum_{(i,j)\in P} cost_X(i,j) = \sum_{(i,j)\in P} \frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i} \leq$$
$$\sum_{(i,j)\in P} \frac{v_i(X_j)}{w_{\min}} \leq \sum_{(i,j)\in P} \frac{V\cdot|X_j|}{w_{\min}} \leq \frac{mV}{w_{\min}}.$$

Therefore, the cost of every path is at most $\frac{mV}{w_{\min}}$, so agent i needs a subsidy of at most $w_i\frac{mV}{w_{\min}}$. By part (4) of Theorem 3.5, at least one agent has a subsidy of 0. This implies a total subsidy of at most $\frac{W-w_{\min}}{w_{\min}}mV = \left(\frac{W}{w_{\min}}-1\right)mV$.

To establish tightness, let us assume all agent has an all-or-nothing valuation for M, such that such that $v_1(M) = mV$ and $v_i(M) = mV - \varepsilon$ for $i \neq 1$. Consider the allocation X, which assigns all items to a single agent 1 (with the minimum entitlement $w_{\min}$). It's evident that: (1) X is WEF-able, (2) satisfies non-zero social welfare property, (3) X is MSW (and hence, Pareto efficient) and (4) its optimal subsidy vector $\mathbf{p}$ satisfies $p_1 = 0$ and $p_i = \frac{w_i}{w_{\min}}(mV-\varepsilon)$ for $i \neq 1$. Therefore, we require $\frac{W-w_{\min}}{w_{\min}}(mV-\varepsilon) = \left(\frac{W}{w_{\min}}-1\right)(mV-\varepsilon)$ in total. As ε can be arbitrarily small, we get a lower bound of $\left(\frac{W}{w_{\min}}-1\right)mV$. □

In the unweighted case $W/w_{\min} = n$, so the upper bound on the subsidy becomes $(n-1)mV$. This is the same upper bound proved by Halpern and Shah (2019) for the unweighted case and additive valuations.

The following lemma is useful for showing weighed envy-freeability and the subsidy bounds.

Lemma 3.7. *For an allocation X, if for all $i,j \in N$, $v_i(X_i) \geq v_j(X_i)$ holds, then X is WEF-able, and the cost of any path from agent i to j is at most $\frac{v_j(X_j)}{w_j} - \frac{v_i(X_i)}{w_i}$.*

Proof.
1. It is sufficient to show that X satisfies reassignment-stability. Indeed, for any permutation π, let π^{-1} be the inverse permutation. Then the condition in the lemma implies

$$\sum_{i\in N} \frac{v_i(X_i)}{w_i} \geq \sum_{i\in N} \frac{v_{\pi^{-1}(i)}(X_i)}{w_i}.$$

By re-ordering the summands in the right-hand side we get the reassignment-stability condition:

$$\sum_{i \in N} \frac{v_i(X_i)}{w_i} \geq \sum_{i \in N} \frac{v_i(X_{\pi(i)})}{w_{\pi(i)}}.$$

2. For any path P from i to j in the weighted envy-graph,

$$cost_X(P) = \sum_{(h,k) \in P} cost_X(h,k) = \sum_{(h,k) \in P} \frac{v_h(X_k)}{w_k} - \frac{v_h(X_h)}{w_h} \leq \sum_{(h,k) \in P} \frac{v_k(X_k)}{w_k} - \frac{v_h(X_h)}{w_h}.$$

The latter expression is a telescopic sum that simplifies to the difference of two elements:

$$\frac{v_j(X_j)}{w_j} - \frac{v_i(X_i)}{w_i}.$$

□

We now prove the main positive result of this section.

Theorem 3.8. *For every instance with monotone valuations, there exists a WEF-able, non-zero-social-welfare allocation with total subsidy at most* $\left(\frac{W}{w_{\min}} - 1\right) mV$*. This bound is tight in the worst case.*

Proof. Let i^* be an an agent for whom $v_{i^*}(M)$ is largest. Let X be the allocation in which all items are allocated to i^*. It is clear that this allocation satisfies non-zero social welfare property. Also, $v_i(X_i) \geq v_j(X_i)$ holds for any $i, j \in N$. Thus, by Lemma 3.7, the allocation is WEF-able. To eliminate envy, agent i^* should receive no subsidy, while any other agent $i \neq i^* \in N$ should receive a subsidy of $\frac{w_i}{w_{i^*}} v_j(M) \leq \frac{w_i}{w_{\min}} v_j(M)$. For any agent j, $v_j(M) \leq mV$ by definition of V. Therefore, the total subsidy is at most $\left(\frac{W}{w_{\min}} - 1\right) mV$.

To prove tightness, we use the same example as in the tightness proof of Theorem 3.6. Every non-zero-social-welfare allocation must allocate all items to a single agent. Therefore, the situation is identical to allocating a single item. The **only** WEF-able allocation is the one giving all items to agent assigning the highest value to M, who is agent 1 (with the lowest entitlement). Therefore, the subsidy must be $\left(\frac{W}{w_{\min}} - 1\right)(mV - \epsilon)$ for any $\epsilon > 0$. □

Importantly, the maximum worst-case subsidy in the weighted setting depends on the proportion of $w_{\min}$ in the total weight W, which can be much larger than the number of agents n used in the unweighted setting bounds.

Theorem 3.8 guarantees only a very weak efficiency notion: non-zero-welfare. We do not know if WEF is compatible with Pareto-efficiency for general monotone valuations. However, for the large sub-class of *superadditive valuations*, we prove in the following section that every allocation maximizing the sum of utilities (*maximizing social welfare* or MSW) is WEF-able. In particular, a Pareto-efficient WEF allocation always exists. Moreover, we prove that such an allocation can always be attained by a *truthful mechanism* — which induces agents to reveal their true utility functions.

4 Superadditive and Supermodular Valuations

A monotone valuation v_i is called

- *Superadditive* — if for any $X, Y \subseteq M$ with $X \cap Y = \emptyset$, $v_i(X) + v_i(Y) \leq v_i(X \cup Y)$.
- *Supermodular* — if for any $X, Y \subseteq M$ holds $v_i(X) + v_i(Y) \leq v_i(X \cup Y) + v_i(X \cap Y)$.

Every additive valuation is supermodular, and every supermodular valuation is superadditive.

The lower bound for monotone valuations (in the proof of Theorem 3.8) is attained by supermodular valuations, so it applies to supermodular and superadditive valuations too.

In this section we prove that, for superadditive valuations (hence also for supermodular and additive valuations), the same upper bound of Theorem 3.8 can be attained by an allocation that maximizes the social welfare.

Theorem 4.1. *When valuations are superadditive, any MSW allocation is WEF-able.*

Proof. We show that for an MSW allocation X, $v_i(X_i) \geq v_j(X_i)$ holds for any $i, j \in N$. For the sake of contradiction, assume $v_i(X_i) < v_j(X_i)$ holds. In this case, we can construct another allocation X', where for all $k \neq i, j$, $X'_k = X_k$, $X'_i = \emptyset$, $X'_j = X_j \cup X_i$. In other words, we reassign X_i from i to j. By the definition of superadditivity, we have $v_j(X'_j) = v_j(X_i \cup X_j) \geq v_j(X_i) + v_j(X_j)$. Therefore, the total social welfare under allocation X' is

$$SW^N(X') = v_j(X'_j) + \sum_{k \neq i,j} v_k(X_k) > v_i(X_i) + v_j(X_j) + \sum_{k \neq i,j} v_k(X_k) = SW^N(X).$$

However, this contradicts the fact that X is a MSW allocation. From Lemma 3.7, it follows that X is WEF-able. Theorem 3.6 implies that a subsidy of $\left(\frac{W}{w_{\min}} - 1\right) mV$ is sufficient. □

Example 1.2 shows that the theorem does not hold without the superadditivity assumption.

A *mechanism* is a function from the profile of declared agents' valuation functions to an outcome. We say a mechanism is truthful if no agent can obtain a strictly better outcome by misreporting its valuation function.

Definition 4.2 (VCG mechanism (Vickrey (1961); Clarke (1971); Groves (1973))**.** The VCG mechanism chooses an allocation X which maximizes $SW^N(X)$ among all allocations of M to N.

Agent i, who is allocated X_i, pays a price equal to $SW^{N\setminus\{i\}}(X') - SW^{N\setminus\{i\}}(X)$, where X' maximizes $SW^{N\setminus\{i\}}$ among all allocations of M to $N \setminus \{i\}$.

Theorem 4.3. *When valuations are superadditive, the VCG mechanism with a large up-front subsidy (i.e.,we first distribute $C \cdot w_i$ to agent i, and if agent i obtains a bundle, it pays the VCG payment from $C \cdot w_i$) is WEF, Pareto efficient, and truthful.*

Proof. Truthfulness and Pareto efficiency are clear. We show that it is WEF. We first show that in the VCG, for each agent i who obtains X_i and pays q_i, $q_i \geq v_j(X_i)$ holds for any $j \neq i$. For the sake of contradiction, assume $q_i = SW^{N\setminus\{i\}}(X') - SW^{N\setminus\{i\}}(X) < v_j(X_i)$ holds. Then, $SW^{N\setminus\{i\}}(X') < v_j(X_i) + SW^{N\setminus\{i\}}(X)$ holds. However, if we consider an allocation of M to agents except for i, we can first allocate $M \setminus X_i$ optimally among $N \setminus \{i\}$, then allocate X_i additionally to agent j. Then, the total valuation of this allocation is at least $v_j(X_i) + SW^{N\setminus\{i\}}(X)$ due to superadditivity. This contradicts the fact that $SW^{N\setminus\{i\}}(X')$ is the total valuation when allocating M optimally among agents except for i. Also, it is known that VCG is individually rational, which means that $v_i(X_i) \geq q_i$ holds for all $i \in N$.

Combining both inequalities leads to

$$\frac{v_j(X_j) + C \cdot w_j - q_j}{w_j} \geq \frac{C \cdot w_j}{w_j} = C = \frac{C \cdot w_i}{w_i} \geq \frac{v_j(X_i) + C \cdot w_i - q_i}{w_i},$$

which is the WEF condition. □

A similar mechanism is presented by Goko et al. (2024) for the unweighted case.

To guarantee that all subsidies are non-negative, C should be an upper bound on the payment of each agent. As the payment of each agent is at most the social welfare in an allocation, which is at most mV, we can simply take $C := mV/w_{\min}$.

5 Additive Valuation

The valuation function of an agent i is called *Additive* if for each $i \in N$ and $A, B \subseteq M$ such that $A \cap B = \emptyset$, $v_i(A \cup B) = v_i(A) + v_i(B)$. Without loss of generality, we assume that each item is valued positively by at least one agent; items that are valued at 0 by all agents can be allocated arbitrarily without affecting the envy. Halpern and Shah (2019) prove that, with additive valuations, the minimum subsidy required in the worst case is at least $(n-1)V$, even for binary valuations, when the allocation can be chosen. We generalize their results as follows.

Lemma 5.1. *Suppose there are n agents, and only one item o which the agents value positively. Then an allocation is WEF-able iff o is given to an agent i with the highest $v_i(o)$.*

Proof. By Theorem 3.3, it is sufficient to check the cycles in the weighted envy-graph. If o is given to i, then i's envy is $-\frac{v_i(o)}{w_i}$, and the envy of every other agent j in i is $\frac{v_j(o)}{w_i}$. All other envies are 0. The only potential positive-weight cycles are cycles of length 2 involving agent i. The weight of such a cycle is positive iff $\frac{v_j(o)}{w_i} - \frac{v_i(o)}{w_i} > 0$, which holds iff $v_j(o) > v_i(o)$. Therefore, there are no positive-weight cycles iff $v_i(o)$ is maximum. ☐

Theorem 5.2. *For every weights $\mathbf{w}$ and $n \geq 2$, no algorithm can guarantee a total subsidy smaller than $\left(\frac{W}{w_{\min}} - 1\right) V$.*

Proof. Consider an instance with n agents and one item o with valuations $v_1(o) = V$ and $v_i(o) = V - \epsilon$ for $i \geq 2$.

By Lemma 5.1, the only WEF-able allocation is to give o to agent 1. In this case, the minimum subsidy is $p_1 = 0$ and $p_i = w_i \frac{v_i(o)}{w_{\min}} = w_i \frac{(V-\epsilon)}{w_{\min}}$. Summing all subsidies leads to

$$\sum_{i \geq 2} w_i \frac{(V-\epsilon)}{w_{\min}} = \frac{W - w_{\min}}{w_{\min}} (V-\epsilon) = \left(\frac{W}{w_{\min}} - 1\right)(V-\epsilon).$$

As ϵ can be arbitrarily small, we get a lower bound of $\left(\frac{W}{w_{\min}} - 1\right) V$. ☐

In Example 1.1, we showed that the iterated-maximum-matching algorithm (Brustle et al. (2020)) might produce an allocation that is not WEF-able.

We now introduce a new algorithm, Algorithm 1, which extends the iterated-maximum-matching approach to the weighted setting, assuming all weights are integers. The algorithm finds a one-to-many maximum matching between agents and items, ensuring that each agent $i \in N$ receives exactly w_i items. If the number of items remaining in a round is less than W, we add dummy items (valued at 0 by all agents) so that the total number of items becomes W. In Example 1.1, we add 9 dummy items, and perform a one-to-many maximum-value matching between agent and items, resulting in a WEF-able allocation: $X_1 = \emptyset, X_2 = \{o_1, o_2\}$.

The algorithm runs in $\lceil m/W \rceil$ rounds. In each round t, the algorithm computes a one-to-many maximum-value matching $\{X_i^t\}_{i \in N}$ between all agents and unallocated items O_t, where each agent $i \in N$ receives exactly w_i items.

To achieve this, we reduce the problem to the *minimum-cost network flow problem* (Goldberg et al. (1989)) by constructing a flow network and computing the maximum integral flow of minimum cost. The flow network is defined as follows:

- **Layer 1 (Source Node).** a single source node s.
- **Layer 2 (Agents).** a node for each agent $i \in N$, with an arc from s to i, having cost 0 and capacity w_i.
- **Layer 3 (Unallocated Items).** a node for each unallocated item $o \in O_t$, with an arc from each agent $i \in N$ to item o, having cost $-v_i(o)$ and capacity 1.
- **Layer 4 (Sink Node).** a single sink node t, with an arc from each item $o \in O_t$ to t, having 0 cost and capacity 1.

Any integral maximum flow in this network corresponds to a valid matching where each agent $i \in N$ receives exactly w_i items from O_t, and each item is assigned to exactly one agent, the result is a minimum-cost one-to-many matching based on the costs in the constructed network. Because we negate the original costs in our construction, the obtained matching $\{X_i^t\}_{i \in N}$ maximizes the total value with respect to the original valuations. After at most $\lceil m/W \rceil$ valuations, all items are allocated.

Proposition 5.3. *For each round t in Algorithm 1, X^t is WEF-able.*

Proof. We prove that, in every round t, the total cost added to any directed cycle in the weighted-envy graph is non-positive. Combined with Theorem 3.3, this shows that X^t is WEF-able for every round $t \in T$.

Let X^t the allocation computed by Algorithm 1 at iteration t. Note that Algorithm 1 is deterministic. Let C be any directed cycle in $G_{X^t,w}$, and denote $C = (i_1, ..., i_r)$. To simplify notation, we consider i_1 as i_{r+1}.

Given the allocation X^t and the cycle C, we construct a random alternative allocation B^t as follows: for each agent $i_j \in C$, we choose one item $o^t_{i_{j+1}}$ uniformly from i_{j+1}'s bundle and transfer it to i_j's bundle [3]. The expected value of $v_{i_j}(o^t_{i_{j+1}})$, the value of the item removed from i_j's bundle, can be computed as the average value of all items in $X^t_{i_j}$: $\frac{\sum_{o \in X^t_{i_j}} v_{i_j}(o)}{w_{i_j}} = \frac{v_{i_j}(X^t_{i_j})}{w_{i_j}}$. Similarly, the expected value of $v_{i_j}(o^t_{i_j})$, the value of the item added the i_j's bundle, is $\frac{v_{i_j}(X^t_{i_{j+1}})}{w_{i_{j+1}}}$. Thus, the expected change in value between B^t and X^t

[3] Recall that at each iteration, each agent i_j receives exactly w_{i_j} items.

ALGORITHM 1: Weighted Sequence Protocol For Additive Valuations and Integer weights

Input: Instance $(N, M, v, \mathbf{w})$ with additive valuations.
Output: WEF-able allocation X with total required subsidy of at most $(W - w_{\min})V$.
$X_i \leftarrow \emptyset, \forall i \in N$;
$t \leftarrow 1$; $O_1 \leftarrow M$;
while $O_t \neq \emptyset$ **do**
  Construct the flow network $G' = (V', E')$:
  - define $V' = N \cup O_t \cup \{s, t\}$.
  - Add arcs with the following properties:
    - From s to each agent $i \in N$ with cost 0 and capacity w_i.
    - From each agent $i \in N$ to each unallocated item $o \in O_t$, with cost $-v_i(o)$ and capacity 1.
    - From each unallocated item $o \in O_t$ to t with cost 0 and capacity 1.

  Compute an *integral maximum flow of minimum cost* on G', resulting in the one to many matching $\{X_i^t\}_{i \in N}$;
  Set $X_i \leftarrow X_i \cup X_i^t$, for all $i \in N$;
  Set $O_{t+1} \leftarrow O_t \setminus \cup_{i \in N} X_i^t$;
  $t \leftarrow t + 1$
end
return X

is

$$\mathbb{E}\left[\sum_{i \in N} \left(v_i(B_i^t) - v_i(X_i^t)\right)\right] = \sum_{i_j \in C} \left(\mathbb{E}\left[v_{i_j}(o_{i_{j+1}}^t)\right] - \mathbb{E}\left[v_{i_j}(o_{i_j}^t)\right]\right) =$$
$$\sum_{i_j \in C} \frac{v_{i_j}(X_{i_{j+1}}^t)}{w_{i_{j+1}}} - \frac{v_{i_j}(X_{i_j}^t)}{w_{i_j}}.$$

This is exactly the total cost of cycle C. According to Algorithm 1, X^t maximizes the total value among all allocations in which each agent i receives exactly w_i items. Therefore, the left-hand side of the above expression, which is the difference between the sum of values in B^t and the sum of values in X^t, must be at most 0. But the right-hand side of the same expression is exactly the total cost of C. Therefore,

$$0 \geq \mathbb{E}\left[\sum_{i \in N} \left(v_i(B_i^t) - v_i(X_i^t)\right)\right] = cost_{X^t}(C),$$

so the cost of every directed cycle is at most 0, as required. □

As the allocation in each iteration is WEF-able, the output allocation X is WEF-able too.

To compute an upper bound on the subsidy, we adapt the proof technique in Brustle et al. (2020).

Lemma 5.4. *Let X be a WEF-able allocation. For any positive number z, if $cost_X(i,k) \geq -z$ for every edge (i,k) in $G_{X,w}$, then the maximum subsidy required is at most $w_i z$ per agent $i \in N$.*

Proof. Assume $P_i(X) = (i \dots j)$ is the highest-cost path from i in $G_{X,w}$. Note that $\ell_i(X) = cost_X(P_i(X))$. Then it holds for the cycle $C = (i \dots j)$ that $cost_X(C) = \ell_i(X) + cost_X(j,i)$. By Theorem 3.3, $cost_X(C) \leq 0$, thus, $\ell_i(X) \leq -cost_X(j,i) \leq z$. Therefore, $p_i = w_i \ell_i(X) \leq w_i z$. □

In most allocations, including the one resulting from Algorithm 1, it is not always true that $cost_X(i,k) \geq -V$ for every edge (i,k). We use Lemma 5.4 with a modified valuation function $\bar{v}_i$, derived from the weighted valuation $\frac{v_i(X_i)}{w_i}$.

We prove that an allocation that is WEF-able for the original valuations is also WEF-able for the modified valuations (Proposition 5.6), and that the maximum subsidy required by each agent for the original valuations is bounded by the subsidy required for the modified valuations (Proposition 5.7).

Next, we demonstrate that under the modified valuations, the cost of each edge is at least $-V$. Finally, by Lemma 5.4, we conclude that the maximum subsidy required for any agent $i \in N$ is $w_i V$ for the modified valuation $\bar{v}$ (Proposition 5.8) and for the original valuations v as well.

Let X^t be the output allocation from Algorithm 1, computed in iteration t. For each $i \in N$, we define the modified valuation function as follow:

$$\bar{v}_i(X_j^t) = \begin{cases} \frac{v_i(X_i^t)}{w_i} & j = i \\ \frac{v_i(X_j^T)}{w_j} & j \neq i, t = T \\ \max\left(\frac{v_i(X_j^t)}{w_j}, \frac{v_i(X_i^{t+1})}{w_i}\right) & j \neq i, t < T \end{cases}$$

Under the modified valuations, for any two agents $i, j \in N$, the modified-cost assigned to the edge (i,j) in the envy graph (with unit weights) is defined as $\overline{cost}_X(i,j) = \bar{v}_i(X_i) - \bar{v}_i(X_j)$. Moreover, the modified-cost of a path $(i_1, ..., i_k)$ is

$$\overline{cost}_X(i_1, ..., i_k) = \sum_{j=1}^{k-1} \overline{cost}_X(i_j, i_{j+1}).$$

observation 5.5. *For agent $i \in N$ and round $t \in [T]$ it holds that:*

1. $\bar{v}_i(X_i^t) = \frac{v_i(X_i^t)}{w_i}$.

2. For agent $j \neq i \in N$, $\bar{v}_i(X_j^t) \geq \frac{v_i(X_j^t)}{w_j}$*; hence* $\bar{v}_i(X_j^t) - \bar{v}_i(X_i^t) \geq \frac{v_i(X_j^t)}{w_j} - \frac{v_i(X_i^t)}{w_i}$.

Proposition 5.6. *Assume X is WEF-able under the original valuations v. Then, X is EF-able (i.e., WEF-able with unit weights) under the modified valuations $\bar{v}$.*

Proof. By Theorem 3.3, it is sufficient to prove that all directed cycles in the envy graph (with the modified valuations and unit weights) have non-positive total cost.

We prove a stronger claim: in every round t, the total modified-cost added to every directed cycle C is non-positive.

Let X^t be the allocation computed by Algorithm 1 at iteration t. Suppose, contrary to our assumption, that there exists a cycle $C = (i_1, ..., i_r)$ and a round t in which the modified-cost added to C is positive. To simplify notation, we consider i_1 as i_{r+1}. This implies that

$$\sum_{j=1}^{r} \bar{v}_{i_j}(X_{i_{j+1}}^t) > \sum_{j=1}^{r} \bar{v}_{i_j}(X_{i_j}^t). \tag{2}$$

There are several cases to consider.

Case 1: All arcs $i_j \to i_{j+1}$ in C have

$$\bar{v}_{i_j}(X_{i_{j+1}}^t) = \frac{v_{i_j}(X_{i_{j+1}}^t)}{w_{i_{j+1}}}$$

(in particular, this holds for $t = T$). In this case, inequality (2) implies

$$\overline{cost}_{X^t}(C) = \sum_{j=1}^{r} \frac{v_{i_j}(X_{i_j}^t)}{w_{i_{j+1}}} - \frac{v_{i_j}(X_{i_j}^t)}{w_{i_j}} > 0.$$

Combined with Theorem 3.3, this contradicts Proposition 5.3, which states that X^t is WEF-able.

Case 2: All arcs $i_j \to i_{j+1}$ in C have

$$\bar{v}_{i_j}(X_{i_{j+1}}^t) = \frac{v_{i_j}(X_{i_j}^{t+1})}{w_{i_j}}.$$

In this case, inequality (2) implies

$$\sum_{j=1}^{r} \frac{v_{i_j}(X_{i_j}^{t+1})}{w_{i_j}} > \sum_{j=1}^{r} \frac{v_{i_j}(X_{i_j}^t)}{w_{i_j}}.$$

Notice that all the items in $X_{j_1}^{t+1}, ..., X_{j_r}^{t+1}$ are available at iteration t, which contradicts the optimality of $\{X_i^t\}_{i \in N}$.

Case 3: Some arcs $i_j \to i_{j+1}$ in C satisfy Case 1 and the other arcs satisfy Case 2.

Let $l \geq 1$ be the number of arcs in C that satisfy Case 2. We decompose C into a sequence of l edge-disjoint paths, denoted $P_1, ..., P_l$, such that the last node of each path is the first node of the next path, and in each path, only the last edge satisfies Case 2.

Formally, suppose that some path contains $k \geq 1$ agents, denoted as $i_1, ..., i_k$, and $k-1$ arcs. Then for each $1 \leq j \leq k-2$,

$$\bar{v}_{i_j}\left(X_{i_{j+1}}^t\right) = \frac{v_{i_j}\left(X_{i_{j+1}}^t\right)}{w_{i_{j+1}}},$$

and

$$\bar{v}_{i_{k-1}}\left(X_{i_k}^t\right) = \frac{v_{i_{k-1}}\left(X_{i_{k-1}}^{t+1}\right)}{w_{i_{k-1}}}.$$

Since $\overline{cost}_{X^t}(C) > 0$, there exits a path $P = (i_1, ..., i_k)$ where $\overline{cost}_{X^t}(P) > 0$, which implies that:

$$0 < \sum_{j=1}^{k-1}\left(\bar{v}_{i_j}(X_{i_{j+1}}^t) - \bar{v}_{i_j}(X_{i_j}^t)\right) = \sum_{j=1}^{k-2}\left(\frac{v_{i_j}(X_{i_{j+1}}^t)}{w_{i_{j+1}}} - \frac{v_{i_j}(X_{i_j}^t)}{w_{i_j}}\right) + \frac{v_{i_{k-1}}(X_{i_{k-1}}^{t+1})}{w_{i_{k-1}}} - \frac{v_{i_{k-1}}(X_{i_{k-1}}^t)}{w_{i_{k-1}}}.$$

The rest of the proof is similar to the proof of Proposition 5.3. We construct another allocation B^t randomly as follows:

1. For each agent $1 \leq j \leq k-1$, we choose one item $o_{i_{j+1}}^t$ uniformly from i_{j+1}'s bundle and transfer it to i_j's bundle.
2. We choose one item $o_{i_1}^t$ uniformly from i_1's bundle to remove.
3. We choose one item $o_{i_{k-1}}^{t+1}$ uniformly from $X_{i_{k-1}}^{t+1}$ and add it to i_{k-1}'s bundle.

Thus, the expected change in value between B^t and X^t is

$$\mathbb{E}\left[\sum_{i \in N}\left(v_i(B_i^t) - v_i(X_i^t)\right)\right] =$$

$$\sum_{1 \leq j \leq k-2}\left(\mathbb{E}\left[v_{i_j}(o_{i_{j+1}}^t)\right] - \mathbb{E}\left[v_{i_j}(o_{i_j}^t)\right]\right) + \mathbb{E}\left[v_{i_{k-1}}(o_{i_{k-1}}^{t+1})\right] - \mathbb{E}\left[v_{i_{k-1}}(o_{i_{k-1}}^t)\right] =$$

$$\sum_{1 \leq j \leq k-2} \frac{v_{i_j}(X_{i_{j+1}}^t)}{w_{i_{j+1}}} - \frac{v_{i_j}(X_{i_j}^t)}{w_{i_j}} + \frac{v_{i_{k-1}}(X_{i_{k-1}}^{t+1})}{w_{i_{k-1}}} - \frac{v_{i_{k-1}}(X_{i_{k-1}}^t)}{w_{i_{k-1}}}.$$

This is exactly the cost of P which by assumption is greater than 0. However, according to Algorithm 1, X^t maximizes the value of an allocation where each agent i receives w_i items among the set of O^t items. Therefore,

$$0 \leq \mathbb{E}\left[\sum_{i \in N} \left(v_i(B_i^t) - v_i(X_i^t)\right)\right] = cost_{X^t}(P)$$

leading to a contradiction.

To sum up, X^t is WEF-able under the original valuations v (with weights w), and under the modified valuations $\bar{v}$ (with unit weights). □

Proposition 5.7. *For the allocation X computed by Algorithm 1, the subsidy required by an agent given v (with weights w) is at most the subsidy required given $\bar{v}$ (with unit weights).*

Proof. Given Observation 5.5, for each $i, j \in N$,

$$\bar{v}_i(X_j) - \bar{v}_i(X_i) \geq \frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i}.$$

Thus, the cost of any path in the envy graph under the modified function and unit weights is at least the cost of the same path in the weighted envy-graph with the original valuations. □

Proposition 5.8. *For the allocation X computed by Algorithm 1, the subsidy to each agent is at most $w_i V$ for the modified valuation profile $\bar{v}$.*

Proof. By Proposition 5.6, the allocation X is WEF-able under the valuations $\bar{v}$. Together with Lemma 5.4, if for each $i, j \in N$ it holds that $\bar{v}_i(X_j) - \bar{v}_i(X_i) \geq -V$, the subsidy required for agent $i \in N$ is at most $w_i V$ for $\bar{v}$.

$$\begin{aligned}
&\bar{v}_i(X_j) - \bar{v}_i(X_i) = \sum_{t \in [T]} \bar{v}_i(X_j^t) - \sum_{t \in [T]} \bar{v}_i(X_i^t) = \\
&\sum_{t \in [T-1]} \max\left\{\frac{v_i(X_j^t)}{w_j}, \frac{v_i(X_i^{t+1})}{w_i}\right\} + \frac{v_i(X_j^T)}{w_j} - \sum_{t \in [T]} \frac{v_i(X_i^t)}{w_i} \geq \\
&\sum_{t \in [T-1]} \frac{v_i(X_i^{t+1})}{w_i} + \frac{v_i(X_j^T)}{w_j} - \sum_{t \in [T]} \frac{v_i(X_i^t)}{w_i} = \\
&\frac{v_i(X_j^T)}{w_j} - \frac{v_i(X_i^1)}{w_i} \geq -\frac{v_i(X_i^1)}{w_i}.
\end{aligned}$$

Since X_i^1 contains exactly w_i items, $-v_i(X_i^1) \geq -w_i V$. Hence, $\bar{v}_i(X_j) - \bar{v}_i(X_i) \geq -\frac{w_i V}{w_i} = -V$. □

We are now prepared to prove the main theorem.

Theorem 5.9. *For additive valuations and integer entitlements, Algorithm 1 computes in polynomial time a WEF-able allocation where the subsidy to each agent is at most $w_i V$ and the total subsidy is at most $(W - w_{\min})V$.*

Proof. For the runtime analysis, the most computationally intensive step in Algorithm 1 is solving the maximum integral flow of minimum cost in G'. The flow network G' consists of at most $n + m + 2$ nodes and at most $n + m + mn$ arcs. By Goldberg and Tarjan (1989), this can be done in time polynomial in n, m:

$$O\left((n + m + 2)\,(n + m + mn) \log (n + m + 2) \min\{\log\left((n + m + 2)\,V\right), (n + m + mn) \log (n + m + 2)\}\right).$$

By Proposition 5.3, X is WEF-able under the original valuations. Combined with Proposition 5.6 and Proposition 5.8 , X is also WEF-able under the modified valuations and requires a subsidy of at most $w_i V$ for each agent $i \in N$.

Proposition 5.7, implies that under the original valuations, the required subsidy for each agent $i \in N$ is at most $w_i V$. By Theorem 3.5, there is at least one agent who requires no subsidy, so the required total subsidy is at most $(W - w_{\min})V$. □

The WEF condition is invariant to multiplying the weight vector by a scalar. This can be used in two ways:

(1) If the weights are not integers, but their ratios are integers, we can still use Algorithm 1. For example, if $w_1 = 1/3$ and $w_2 = 2/3$ (or even if w_i's are irrational numbers such as $w_1 = \sqrt{2}$ and $w_2 = 2\sqrt{2}$), Algorithm 1 works correctly by resetting $w_1 = 1$ and $w_2 = 2$.

(2) If the weights are integers with greatest common divisor (gcd) larger than 1, we can divide all weights by the gcd to get a better subsidy bound.

Lemma 5.10. *For additive valuations and integer entitlements, there exists an algorithm that computes in polynomial time a WEF-able allocation where the subsidy to each agent is at most $w_i V/\gcd(\mathbf{w})$ and the total subsidy is at most $(W - w_{\min})V/\gcd(\mathbf{w})$, where $\gcd(\mathbf{w})$ is the greatest common divisor of all the w_i.*

Proof. Algorithm 1 works correctly, even if we divide each w_i by the greatest common divisor of w_i's. In other words, letting $d = \gcd(w_1, ..., w_n)$, $w'_i = w_i/d$, $W' = W/d$, and running Algorithm 1 with w'_i's, we get the bound $(W' - w'_{\min})V$ of the total subsidy. □

A discussion about the tightness of the bound can be found in Appendix A.1.

5.1 Combining WEF-able and Approximate-WEF

In the setting without money and additive valuations, WEF can be relaxed by allowing envy up to an upper bound based on item values. We adopt the generalization of allowable envy, *WEF*(x, y), proposed by Chakraborty et al. (2022) [4], as well as another relaxation of WEF, *WWEF1*, introduced in Chakraborty et al. (2021a).

Definition 5.11 (Chakraborty et al. (2022))**.** For $x, y \in [0, 1]$, an allocation X is said to satisfy *WEF(x,y)* if for any $i, j \in N$, there exists $B \subseteq X_j$ with $|B| \leq 1$ such that $\frac{v_i(X_i)+yv_i(B)}{w_i} \geq \frac{v_i(X_j)-xv_i(B)}{w_j}$.

WEF(x, y) captures various conditions related to WEF:

- WEF$(0, 0)$ corresponds to WEF.
- WEF$(1, 0)$ coincides with (strong) weighted envy-freeness up to one item (WEF1) (Chakraborty et al. (2021a)).
- WEF$(1, 1)$ coincides with weighted envy-free up to one item transfer (WEF1-T) (Aziz et al. (2023); Hoefer et al. (2024)).

Definition 5.12 (Chakraborty et al. (2021a))**.** An allocation X is said to be *weakly weighted envy-free up to one item* (WWEF1) if for any pair of agents i, j with $X_j \neq \emptyset$, there exists an item $o \in X_j$ such that either $\frac{v_i(X_i)}{w_i} \geq \frac{v_i(X_j \setminus \{o\})}{w_j}$ or $\frac{v_i(X_i \cup \{o\})}{w_i} \geq \frac{v_i(X_j)}{w_j}$.

Halpern and Shah proved in Halpern and Shah (2019) that, if an allocation X is EF-able and EF1, the total subsidy of at most $(n-1)^2 V$ is sufficient. The following theorem generalizes this result to the weighted setting.

Theorem 5.13. *Let X be both WEF-able and WEF(x, y) for some $x, y \in [0, 1]$. Then there exists a subsidy vector $\mathbf{p}$, such that $(X, \mathbf{p})$ WEF, with total subsidy at most $(x + y)\left(\frac{W}{w_{\min}} - 1\right)(n-1)V$.*

Proof. X is WEF(x, y), so for all $i, j \in N$, there exists $B \subseteq X_j$ with $|B| \leq 1$ where

$$\frac{v_i(X_j)}{w_j} - \frac{v_i(X_i)}{w_i} \leq \frac{yv_i(B)}{w_i} + \frac{xv_i(B)}{w_j} \leq \frac{yv_i(B)}{w_{\min}} + \frac{xv_i(B)}{w_{\min}} = \frac{(x+y)v_i(B)}{w_{\min}} \leq (x+y)\frac{V}{w_{\min}}.$$

Any path contains at most $n - 1$ arcs, that is, $p_i = w_i \ell_i \leq (x + y)\frac{w_i}{w_{\min}}(n-1)V$.

By Theorem 3.5, there is at least one agent that requires no subsidy, so the required total subsidy is at most $(x + y)\frac{W - w_{\min}}{w_{\min}}(n-1)V = (x + y)\left(\frac{W}{w_{\min}} - 1\right)(n-1)V$. ∎

[4] The definition of WEF(x, y) does not apply to non-additive valuations. Montanari et al. (2024) introduced two extensions to this definition; but they are outside the scope of our paper.

Brustle et al. (2020) proved that the allocation resulting from their algorithm satisfies both EF and EF1. However, in the weighted setup, a WEF-able allocation may not satisfy WEF(x, y) for any x, y with $x + y < 2$. This also holds for the allocation produced by Algorithm 1.

Proposition 5.14. *For any $x, y \geq 0$ with $x + y < 2$, there exists a weight vector and an instance with additive valuations in which that every WEF-able allocation fails to satisfy WEF(x, y) or WWEF1.*

Proof. Consider an instance with two identical items and two agents with weights $w_1 < w_2$. Agent 1 values each item at 1 and agent 2 values each item at 2.

The only WEF-able allocation is the one giving both items to agent 2: $X_1 = \emptyset, X_2 = \{o_1, o_2\}$. (If we allocate one item to each agent, the cost of the cycle between those agents would be $\left(\frac{1}{w_2} - \frac{1}{w_1}\right)(v_2(o) - v_1(o))$, which is positive).

For this allocation to be WEF(x, y), it must satisfy

$$\frac{v_1(X_1) + yv_1(o_1)}{w_1} \geq \frac{v_1(X_2) - xv_1(o_1)}{w_2} \iff \frac{y}{w_1} \geq \frac{2 - x}{w_2} \iff x + \frac{w_2}{w_1}y \geq 2.$$

Hence, if $x + y < 2$ and $\frac{w_2}{w_1}$ is sufficiently close to 1, then X fails WEF(x, y).

Moreover, X does not satisfy WWEF1, as $0 = \frac{v_1(X_1)}{w_1} < \frac{v_1(X_2 \setminus \{o\})}{w_2} = 1$ and $\frac{1}{w_1} = \frac{v_1(X_1 \cup \{o\})}{w_1} < \frac{v_1(X_2)}{w_2} = \frac{2}{w_2}$, for any $o \in X_2$, whenever $w_2 < 2w_1$. □

We also observe that even for 2 agents, the *Weighted Picking Sequence Protocol* (Chakraborty et al. (2021a)), which outputs a WEF$(1, 0)$ allocation for any number of agents with additive valuations, is not WEF-able:

Example 5.15. Suppose there is one item o and $v_1(o) = 2$, $v_2(o) = 1$, $w_1 = 1$, and $w_2 = 4$. Agent 2 gets the first turn and gets o. Agent 2 gets the first turn and gets o. The envy of agent 1 toward agent 2 is $2/4$, while the envy of agent 2 toward agent 1 is $-1/4$. Thus, the cycle between these agents has a positive cost: $2/4 - 1/4 = 1/4 > 0$. By Theorem 3.3, the resulting allocation is not WEF-able.

Biased Weighted Adjusted Winner Procedure Proposition 5.14 is partly complemented by the result below (Theorem 5.16) that states that WEF-ability and WEF(1,1) are compatible for two agents having additive valuations. The theorem uses a particular version of the *Weighted Adjusted Winner* procedure (Chakraborty et al. (2021a)). The original procedure finds a WEF$(1, 0)$ and Pareto efficient allocation. We call our variant *Biased Weighted Adjusted Winner Procedure*, as it is biased towards the agent who expresses a higher value for a 'contested' item. The resulting allocation may not be WEF$(1, 0)$, but it is WEF$(1, 1)$ and WEF-able.

We first observe that for two agents, the WEF condition for each agent i is equivalent to the *weighted proportionality* (WPROP) condition: $v_i(X_i) \geq \frac{w_i}{W} \cdot v_i(M)$.

1. Normalize the valuations so that the sum of values over all items is 1 for both agents. Sort the items such that $\frac{v_1(o_1)}{v_2(o_1)} \geq \frac{v_1(o_2)}{v_2(o_2)} \geq \cdots \geq \frac{v_1(o_m)}{v_2(o_m)}$.
2. Let $d \in \{1, 2, ..., m\}$ be the unique number satisfying $\frac{1}{w_1} \sum_{r=1}^{d-1} v_1(o_r) < \frac{1}{w_2} \sum_{r=d}^{m} v_1(o_r)$ and $\frac{1}{w_1} \sum_{r=1}^{d} v_1(o_r) \geq \frac{1}{w_2} \sum_{r=d+1}^{m} v_1(o_r)$. We call o_d the *contested object*.
3. Denote $L := \{o_1, \ldots, o_{d-1}\}$ and $R := \{o_{d+1}, \ldots, o_m\}$ (the "Left" and "Right" parts); note that each of them might be empty. Give L to agent 1 and R to agent 2.
4. Finally, give o_d to the agent i with largest $v_i(o_d)$ (break ties arbitrarily).

Theorem 5.16. *The outcome of the* Biased Weighted Adjusted Winner Procedure *is both WEF*$(1, 1)$ *and WEF-able.*

Proof. The WEF$(1, 1)$ for agent 1 follows immediately from the definition of o_d, as

$$\frac{1}{w_1} v_1(L \cup \{o_d\}) \geq \frac{1}{w_2} v_1(R).$$

If o_d is allocated to 1 then 1 does not envy at all, otherwise 1 stops envying after o_d is transferred to him.

As for agent 2, the item ordering implies that

$$\frac{v_1(L)}{v_2(L)} \geq \frac{v_1(R \cup \{o_d\})}{v_2(R \cup \{o_d\})} \iff \frac{v_2(R \cup \{o_d\})}{v_2(L)} \geq \frac{v_1(R \cup \{o_d\})}{v_1(L)}$$

By the definition of o_d, $\frac{v_1(L)}{v_1(R \cup \{o_d\})} < \frac{w_1}{w_2}$. Combining this with the above inequality implies

$$\frac{v_2(R \cup \{o_d\})}{v_2(L)} > \frac{w_2}{w_1} \iff \frac{1}{w_2} v_2(R \cup \{o_d\}) > \frac{1}{w_1} v_2(L).$$

An analogous argument to that used for agent 1 shows that the WEF$(1, 1)$ condition is satisfied for 2 too.

Next, we prove that the outcome is WEF-able.

Suppose o_d is given to agent 2, that is, $v_2(o_d) \geq v_1(o_d)$. By definition of o_d, there exists a fraction $r \in [0, 2]$ such that

$$\frac{1}{w_1}(v_1(L) + r \cdot v_1(o_d)) = \frac{1}{w_2}(v_1(R \cup \{o_d\}).$$

Let the subsidy to agent 1 be $p_1 := r \cdot v_1(o_d)$, and the subsidy to agent 2 be $p_2 := 0$. The resulting allocation is WEF for agent 1 by definition.

As for agent 2, the item ordering implies

$$\frac{v_1(L) + rv_1(o_d)}{v_2(L) + rv_2(o_d)} \geq \frac{v_1(R \cup \{o_d\})}{v_2(R \cup \{o_d\})} \iff v_2(L) + rv_2(o_d) \leq \frac{v_2(R \cup \{o_d\})}{v_1(R \cup \{o_d\})}(v_1(L) + rv_1(o_d)) = \frac{w_1}{w_2} v_2(R \cup \{o_d\}).$$

As $v_1(o_d) \leq v_2(o_d)$, agent 2 values agent 1's bundle at

$$v_2(L) + p_1 = v_2(L) + r \cdot v_1(o_d) \leq v_2(L) + r \cdot v_2(o_d),$$

which by the above inequality is at most $\frac{w_1}{w_2} v_2(R \cup \{o_d\})$. Hence, the allocation is WEF for agent 2 too.

The case that o_d is allocated to agent 1 can be proved in a similar way. □

It remains open whether weighted envy-freeability and WEF(1,1) are compatible for $n \geq 3$ agents.

6 Identical Additive Valuation

This section deals with the case where all agents have identical valuations, that is, $v_i \equiv v$ for all $i \in N$. With identical valuations, any allocation is WEF-able by Lemma 3.7. Furthermore, all allocations are non-wasteful. We present a polynomial-time algorithm for finding a WEF-able allocation with a subsidy bounded by V per agent and a total subsidy bounded by $(n-1)V$. The following shows that this bound is tight for any weight vector:

Theorem 6.1. *For identical additive valuations, for any integer weights vector, there exists an instance where, for any WEF-able allocation, at least one agent requires subsidy at least V, and the total subsidy is at least $(n-1)V$.*

Proof. Consider n agents with integer weights $w_1 \leq \cdots \leq w_n$ and $m = 1 + \sum_{i \in N} (w_i - 1)$ items all valued at V.

To avoid envy, each agent i should receive a total utility of $w_i V$, so the sum of all agents' utilities would be WV.

As the sum of all values is $(W - (n-1))V$, a total subsidy of at least $(n-1)V$ is required (to minimize the subsidy per agent, each agent $i \in N$ should receive $w_i - 1$ items, except for the agent with the highest entitlement (agent n), who should receive w_n items.

The value per unit entitlement of each agent $i < n$ is $V(w_i - 1)/w_i$, and for agent n it is V. Therefore, to avoid envy, each agent $i < n$ should receive a subsidy of $w_i \left(1 - \frac{w_i - 1}{w_i}\right) V = V$ and the total subsidy required is $(n-1)V$. □

Algorithm 2 presents our WEF-able allocation method with bounded subsidy.

The algorithm traverses the items in an arbitrary order. At each iteration it selects the agent that minimizes the expression $\frac{v(X_i \cup \{o\})}{w_i}$, with ties broken in favor of the agent with the larger w_i, and allocates the next item to that agent. Intuitively, this selection minimizes the likelihood that weighted envy is generated.

ALGORITHM 2: Weighted Sequence Protocol For Additive Identical Valuations

---

Input: Instance $(N, M, v, \mathbf{w})$ with additive identical valuations.
Output: WEF-able allocation X with total required subsidy of at most $(n-1)V$.
$X_i \leftarrow \emptyset, \forall i \in N$;
for $o : 1$ *to* m **do**
    $U \leftarrow \arg\min_{i \in N} \frac{v(X_i \cup \{o\})}{w_i}$;
    $u \leftarrow \max_{i \in U} (i)$;
    Add o to X_u
end
return X

The following example illustrates Algorithm 2:

Example 6.2. Consider two agents, denoted as i_1 and i_2, with corresponding weights $w_1 = 1$ and $w_2 = \frac{7}{2}$, and three items, namely o_1, o_2, o_3, with valuations $v(o_1) = v(o_2) = v(o_3) = 1$, Algorithm 2 is executed as follows:

1. for the first iteration, the algorithm compares $\frac{v(o_1)}{w_1} = 1$ and $\frac{v(o_1)}{w_2} = \frac{2}{7}$. Consequently, the algorithm allocates item o_1 to agent i_2, resulting in $X_1 = \emptyset$ and $X_2 = \{o_1\}$.

2. for the second iteration, the algorithm compares $\frac{v(o_2)}{w_1} = 1$ and $\frac{v(X_2 \cup \{o_2\})}{w_2} = \frac{2}{\frac{7}{2}} = \frac{4}{7}$. Subsequently, the algorithm allocates item o_2 to agent i_2, resulting in $X_1 = \emptyset$ and $X_2 = \{o_1, o_2\}$.

3. for the third iteration, the algorithm compares $\frac{v(o_3)}{w_1} = 1$ and $\frac{v(X_2 \cup \{o_3\})}{w_2} = \frac{3}{\frac{7}{2}} = \frac{6}{7}$, Consequently, item o_3 is allocated to agent i_2, resulting in $X_1 = \emptyset$ and $X_2 = \{o_1, o_2, o_3\}$.

Now, agent i_1 envies agent i_2 by an amount of $\frac{v(X_2)}{w_2} - \frac{v(X_1)}{w_1} = \frac{3}{\frac{7}{2}} = \frac{6}{7}$, and conversely, agent i_2 envies agent i_1 by $\frac{v(X_1)}{w_1} - \frac{v(X_2)}{w_2} = -\frac{6}{7}$ In order to mitigate envy, $p_1 = \frac{6}{7}$ and $p_2 = 0$.

We start by observing that, with identical valuations, the cost of any path in the weighted envy graph is determined only by the agents at the endpoints of that path.

observation 6.3. *Given an instance with identical valuations, let X be any allocation, and denote by P any path in the weighted envy-graph of X between agents $i, j \in N$.*

$$cost_X(P) = \frac{v(X_j)}{w_j} - \frac{v(X_i)}{w_i}$$

This is because the path cost is $\sum_{(h,k)\in P} cost_X(h,k) = \sum_{(h,k)\in P} \frac{v(X_k)}{w_k} - \frac{v(X_h)}{w_h}$, and the latter sum is a telescopic sum that reduces to the difference of its last and first element.

Example 6.2 illustrates that the resulting allocation may not be *WEF(1,0)* — Algorithm 2 might allocate all items to the agent with the highest entitlement. However, the outcome is always WEF$(0,1)$:

Proposition 6.4. *For additive identical valuations, Algorithm 2 computes a WEF$(0,1)$ allocation.*

Proof. We prove by induction that at each iteration, X satisfies WEF$(0,1)$. The claim is straightforward for the first iteration. Assume the claim holds for the $(t-1)$-th iteration, and prove it for the t-th iteration. Let o be the item assigned in this iteration and u be the agent receiving this item. Agent u satisfies WEF$(0,1)$ due to the induction hypothesis. For $i \neq u$, by the selection rule, $\frac{v(X_u)}{w_u} \leq \frac{v(X_i \cup \{o\})}{w_i}$. This is exactly the definition of WEF$(0,1)$. □

Proposition 6.5. *With identical additive valuations, for every WEF$(0,1)$ allocation X, $\ell_i(X) \leq \frac{V}{w_i}$, for all $i \in N$.*

Proof. For each agent $i \in N$, denote the highest-cost path starting at i in that graph by $P_i(X) = (i, ..., j)$ for some agent $j \in N$. Then by Observation 6.3, $\ell_i(X) = cost_X(P_i(X)) = \frac{v(X_j)}{w_j} - \frac{v(X_i)}{w_i}$.

From the definition of WEF$(0,1)$, Proposition 6.4 implies that this difference is at most $\frac{v(o)}{w_i}$ for some object $o \in X_j$. Therefore, the difference is at most $\frac{V}{w_i}$. □

Theorem 6.6. *For identical additive valuation, there exists a polynomial time algorithm to find a WEF-able and non-wasteful allocation such that the subsidy per agent is at most V. Therefore, the total subsidy required is at most $(n-1)V$.*

Proof. It is clear that Algorithm 2 runs in polynomial time.

Let X be the output of Algorithm 2. First notice that X is WEF-able and non-wasteful.

Proposition 6.5 implies that, to achieve weighted-envy-freeness under identical additive valuations for the allocation computed by Algorithm 2, the required subsidy per agent $i \in N$ is at most $w_i \frac{V}{w_i} = V$. In combination with Theorem 3.5, the total required subsidy is at most $(n-1)V$. □

Note that $W \geq nw_{\min}$. Therefore, this bound is better than the one proved in Theorem 5.9 for integer weights: $(W - w_i)\,V \geq (n-1)\,w_i V \geq (n-1)\,V$.

The upper bound of $(n-1)V$ is tight even for equal entitlements (Halpern and Shah (2019)). Interestingly, when either the valuations or the entitlements are identical, the worst-case upper bound depends on n, whereas when both valuations and entitlements are different, the bound depends on W.

7 Binary Additive Valuation

In this section we focus on the special case of agents with binary additive valuations. We assume $v_i(o) \in \{0, 1\}$ for all $i \in N$ and $o \in M$. We start with a lower bound.

Proposition 7.1. *For every $n \geq 2$ and weight vector $\mathbf{w}$, there is an instance with n agents with binary valuations in which the total subsidy in any WEF allocation is at least $\frac{W}{w_2} - 1$.*

Proof. There is one item. Agents 1 and 2 value the item at 1 and the others at 0. If agent $i \in \{1, 2\}$ gets the item, then the other agent $j \neq i \in \{1, 2\}$ must get subsidy $\frac{w_j}{w_i}$. To ensure that other agents do not envy j's subsidy, every other agent $k \notin \{1, 2\}$ must get subsidy $\frac{w_k}{w_i}$. The total subsidy is $\frac{W}{w_i} - 1$. The subsidy is minimized by giving the item to agent 2, since $w_2 \geq w_1$. This gives a lower bound of $\frac{W}{w_2} - 1$. □

Below, we show how to compute a WEF-able allocation where the subsidy given to each agent $i \in N$ is at most $\frac{w_i}{w_{\min}}V = \frac{w_i}{w_{\min}}$, and the total subsidy is at most $\frac{W}{w_{\min}} - 1$.

In the case of binary valuations, Algorithm 1 is inefficient in three ways: (1) the maximum-value matching does not always prioritize agents with higher entitlements, (2) there may be situations where an agent prefers items already allocated in previous iterations, while the agent holding those items could instead take unallocated ones, and (3) it works only for agents with integer weights.

We address these issues by adapting the *General Yankee Swap (GYS)* algorithm introduced by Viswanathan and Zick (2023a).

GYS starts with an empty allocation for all agents. We add a dummy agent i_0 and assume that all items are initially assigned to i_0: $X_{i_0} = M$.

Algorithm 3 presents our approach for finding a WEF-able allocation with a bounded subsidy. The algorithm runs in T iterations. We denote by X^t the allocation at the end of iteration $t \in [T]$. Throughout this algorithm, we divide the agents into two sets:

1. R: The agents remaining in the game at the beginning of the iteration t.

2. $N \backslash R$: The agents who were removed from the game in earlier iteration $t' < t$. Agents are removed from the game when the algorithm deduces that their utility cannot be improved.

As long as not all the objects have been allocated, at every iteration t, the algorithm looks for the agents maximizing the *gain function* (Viswanathan and Zick (2023a)) among R, i.e., the agents remaining in the game at this iteration.

We use the gain function: $\frac{w_i}{v_i(X_i^{t-1})+1}$, which selects agents with the minimal potential for increasing envy. If multiple agents have the same value, we select one arbitrarily.

The selected agent then chooses either to acquire an unallocated item or take an item from another agent. In either case, their utility increases by 1. If the agent takes an item from another, the affected agent must decide whether to take an unallocated item or another allocated item to preserve their utility, and so on. This process creates a *transfer path* from agent i to the dummy agent i_0 , where items are passed until an unallocated item is reached. Formally, we represent this as a directed graph, where nodes are agents, and an edge (i, j) if and only if there exists an item in j's bundle that i values positively. A *transfer path* is any directed path in that graph, that ends at the dummy agent i_0.

When an agent is selected, the algorithm attempts to find a transfer path from that agent, preserving utilities for all agents except the initiator, whose utility increases by 1. If no path is found, the agent is removed from the game. We use the polynomial-time method by Viswanathan and Zick (2023a) to find transfer paths.

Algorithm 3 differs from GYS in the following way: at the beginning of iteration t, the algorithm first removes all agents without a transfer path originating from them (line 3). Then, it selects an agent based on the gain function to allocate a new item to that agent. For convenience, we denote by $R(t)$ the agents who have a transfer path originating from them at the beginning of iteration t (line 3).

The following example demonstrates Algorithm 3:

Example 7.2. Consider two agents with weights $w_1 = 1$ and $w_2 = 2$, and five items. The valuation functions are:

$$\begin{bmatrix} & o_1 & o_2 & o_3 & o_4 & o_5 \\ i_1 & 1 & 1 & 1 & 1 & 1 \\ i_2 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The algorithm is executed as follows:

1. For $t = 1$, the algorithm compares $\frac{1}{w_1} = \frac{1}{1}$, $\frac{1}{w_2} = \frac{1}{2}$. Consequently, the algorithm searches for a transfer path starting at i_2 and ending at i_0, and finds the path (i_2, i_0). The algorithm transfers the item o_1 to agent i_2 from i_0's bundle, resulting in $X_1^1 = \emptyset$ and $X_2^1 = \{o_1\}$.

2. For $t = 2$, the algorithm compares $\frac{1}{w_1} = \frac{1}{1}$ and $\frac{v_2(X_2^1)+1}{w_2} = \frac{2}{2}$. Since those values are equal, the algorithm arbitrarily selects agent i_2 and searches for a transfer path starting at i_2 and ending at i_0, and finds the path (i_2, i_0). The algorithm transfers the item o_2 to agent i_2, yielding $X_1^2 = \emptyset$ and $X_2^2 = \{o_1, o_2\}$.

ALGORITHM 3: Weighted Sequence Protocol For Additive Binary Valuations

Input: Instance $(N, M, v, \mathbf{w})$ with additive binary valuations.
Output: WEF-able allocation X with total required subsidy of at most $\frac{W}{w_{\min}} - 1$.

```
X_{i_0} ← M, and X_i^0 ← ∅ for each i ∈ N /* All items initially are unassigned */
t ← 1;
R ← N;
while R ≠ ∅ do
    Remove from R all agents who do not have a transfer path starting from them ;
    u ← argmax_{i∈R} ( w_i / (v_i(X_i^{t-1})+1) ) /* Choose the agent who maximizes the gain function */
    Find a transfer path starting at u /* For example, one can use the BFS algorithm to find a shortest path from u to i_0. */
    Transfer the items along the path and update the allocation X^t;
    t ← t + 1
end
return X^t
```

3. For $t = 3$, the algorithm compares $\frac{1}{w_1} = 1$ and $\frac{v_2(X_2^2)+1}{w_2} = \frac{3}{2}$. As a result, the algorithm searches for a transfer path starting at i_1 and ending at i_0, and finds the path (i_1, i_0). The algorithm transfers the item o_3 to agent i_1, producing $X_1^3 = \{o_3\}$ and $X_2^3 = \{o_1, o_2\}$.

4. For $t = 4$, the algorithm compares $\frac{v_1(X_1^3)+1}{w_1} = 2$ and $\frac{v_2(X_2^3)+1}{w_2} = \frac{3}{2}$. Thus, the algorithm searches for a transfer path starting at i_2 and ending at i_0, and finds the path (i_2, i_0). The algorithm transfers the item o_4 to agent i_2, leading $X_1^4 = \{o_3\}$ and $X_2^4 = \{o_1, o_2, o_4\}$.

5. For $t = 5$, the algorithm compares $\frac{v_1(X_1^4)+1}{w_1} = 2$ and $\frac{v_2(X_2^4)+1}{w_2} = \frac{4}{2} = 2$. Since those values are equal, the algorithm arbitrarily selects agent i_2 and searches for a transfer path starting at i_2 and ending at i_0, and finds the path (i_2, i_1, i_0). The algorithm transfers the item o_3 to agent i_2 from i_1's bundle and the item o_5 to agent i_1 from i_0's bundle, leading $X_1^5 = \{o_5\}$ and $X_2^5 = \{o_1, o_2, o_3, o_4\}$.

6. Agent 1 envies agent 2 by $\frac{4}{2} - \frac{1}{1} = 1$, while agent 2 envies agent 1 by $0 - \frac{4}{2} < 0$.

7. In order to mitigate envy, $p_1 = 1$ and $p_2 = 0$.

Definition 7.3 (Viswanathan and Zick (2023a)). An allocation X is said to be *non-redundant* if for all $i \in N$, we have $v_i(X_i) = |X_i|$.

That is, $v_j(X_i) \leq |X_i| = v_i(X_i)$ for every $i, j \in N$, and therefore, every non-redundant allocation is also WEF-able by Lemma 3.7. Lemma 3.1 in Viswanathan and Zick (2023a) shows that the allocation produced by GYS is non-redundant. The same is true for our variant:

Lemma 7.4. *At the end of any iteration t of Algorithm 3, the allocation X^t is non-redundant.*

Proof. We prove by induction that at the end of each iteration t, X^t remains non-redundant.

For the base case, X^0 is an empty allocation and is therefore non-redundant. Now, assume that at the end of iteration $t-1$, X^{t-1} is non-redundant.

If $X^t = X^{t-1}$, meaning no agent received a new item, the process is complete. Otherwise, let u be the agent who receives new item. Agent u obtains an item via the transfer path $P = (u = i_1, \ldots, i_k)$. For each $1 \leq j < k$, agent i_j receives the item o_j from the bundle of i_{j+1}, given that $v_{i_j}(o_j) = 1$. Agent i_k receives a new item o_k from the bundle of i_0, with $v_{i_k}(o_k) = 1$.

Additionally, for each $1 < j \leq k$, item o_{j-1} is removed from agent i_j's bundle where $v_{i_j}(o_{j-1}) = 1$, since X^{t-1} is non-redundant.

For agents not on the transfer path P, their bundles remain unchanged. Thus, for each agent $i \in N$, it holds that $v_i(X_i^t) = v_i(X_i^{t-1}) = |X_i^{t-1}| = |X_i^t|$, confirming that X^t is non-redundant. □

Based on Lemma 7.4 it is established that at the end of every iteration $t \in [T]$, X^t is WEF-able. The remaining task is to establish subsidy bounds.

We focus on two groups: R and $N \setminus R$.

The selection rule simplifies limit-setting for R and ensures a subsidy bound of 1 (Proposition 7.7). However, understanding the dynamics of the second group, $N \setminus R$, presents challenges, as the selection rule is not applicable for them. For an agent $i \in N \setminus R$, we prove a subsidy bound of $w_i \cdot \frac{1}{w_j}$, for some $j \in R$. In particular, the bound is at most $\frac{w_i}{w_{\min}}$.

observation 7.5. *Let X be any non-redundant allocation. Let $P = (i, \ldots, j)$ be a path in $G_{X,w}$. Then $cost_X(P) \leq \frac{|X_j|}{w_j} - \frac{|X_i|}{w_i}$.*

Lemma 7.6. *Let $j \in N$ be any agent, if $i \in R(t)$, then $\frac{|X_j^t|}{w_j} - \frac{|X_i^t|}{w_i} \leq \frac{1}{w_i}$.*

Proof. If j has never been selected to receive an item, then $|X_j^t| = 0$ and the lemma is trivial.

Otherwise, let $t' \leq t$ be the latest iteration in which j was selected. As agents can not be added to R and by the selection rule, $\frac{v_j(X_j^{t'-1})+1}{w_j} \leq \frac{v_i(X_i^{t'-1})+1}{w_i}$. Then by non-redundancy,

$\frac{|X_j^{t'}|}{w_j} = \frac{v_j(X_j^{t'})}{w_j} = \frac{v_j(X_j^{t'-1})+1}{w_j} \leq \frac{v_i(X_i^{t'-1})+1}{w_i} = \frac{v_i(X_i^{t'})}{w_i} + \frac{1}{w_i} = \frac{|X_i^{t'}|}{w_i} + \frac{1}{w_i}$. As $|X_j^t| = |X_j^{t'}|$ and $|X_i^t| \geq |X_i^{t'}|$, the lemma follows. □

From Lemma 7.6, we can conclude the following:

Proposition 7.7. *If $i \in R(t)$, then $\ell_i(X^t) \leq \frac{1}{w_i}$.*

Proof. Assume $P_i = (i, \ldots, j)$ is the path with the highest total cost starting at i in the $G_{X^t,w}$, i.e., $cost_{X^t}(P_i) = \ell_i(X^t)$. Observation 7.5 implies $\ell_i(X^t) \leq \frac{|X_j^t|}{w_j} - \frac{|X_i^t|}{w_i}$. As $i \in R(t)$, Lemma 7.6 implies $\frac{|X_j^t|}{w_j} - \frac{|X_i^t|}{w_i} \leq \frac{1}{w_i}$. □

To prove this upper bound, we need to establish several claims about agents removed from the game. First, we show that an agent removed from the game does not desire any item held by an agent who remains in the game (Proposition 7.8). As a result, these removed agents will not be included in any transfer path (Proposition 7.9).

Next, we demonstrate that if the cost of a path originating from one of these removed agents at the end of iteration t exceeds the cost at the end of iteration $t' \leq t$ — the iteration when the agent was removed — then there exists an edge in this path, (i_j, i_{j+1}), such that $v_i(X_j^t) = 0$ (Proposition 7.10). Based on these claims, we prove that if at the end of iteration t, the cost of the maximum-cost path starting from agent removed from the game at $t' < t$ exceeds its cost at t', we can upper-bound it by $\frac{1}{w_{\min}}$.

Proposition 7.8. *Let i be an agent removed from the game at the start of iteration t'. Then for all $j \in R(t')$, $v_i(X_j^{t'}) = 0$.*

Moreover, for all $t > t'$ and all $j \in R(t)$, $v_i(X_j^t) = 0$.

Proof. Suppose that $v_i(X_j^{t'}) \neq 0$. This implies there exists some item $o \in X_j^{t'}$ such that $v_i(o) = 1$. We consider two cases:

1. $o \in X_j^{t'-1}$. In this case, at the start of iteration t', there exists a transfer path from i to j. Moreover, there is a transfer path from j to i_0 at the start of iteration t' (otherwise, j would have been removed from the game at t' as well). Concatenating these paths gives a transfer path from i to i_0.

2. $o \notin X_j^{t'-1}$, that is, j received item o during iteration t', from some other agent j' (where $j' = i_0$ is possible). At the start of iteration t', there exists a transfer path from i to j'. Moreover, there is a transfer path from j' to i_0, which is used to transfer the newly allocated item. Concatenating these paths gives a transfer path from i to i_0.

Both cases contradict the assumption that i was removed at t'.

To prove the claim for $t > t'$, we use induction over t.

We assume the claim holds for iteration $t-1 > t'$ and prove it for iteration t. Assume, contrary to the claim, that there exists an agent i who was removed at the start of iteration t', and an agent $j \in R(t)$, such that $v_i(X_j^t) \neq 0$. By the induction hypothesis and the fact that an agent can not be added to R, we have $v_i(X_j^{t-1}) = 0$. Therefore, during iteration t, j must have received a new item o_j that i values at 1.

If o_j was part of i_0's bundle at the start of iteration t', then the transfer path starting at i, (i, i_0), must have already existed at the start of iteration t'.

Alternatively, if o_j was originally in another agent's bundle at the start of iteration t', say agent $k \in N$, then there must have been an iteration between t' and t in which o_j has been transferred from k to another agent, while agent k is compensated by some other item o_k, that agent k wants.

If o_k was part of i_0's bundle at the start of iteration t', then the transfer path starting at i, (i, k, i_0), must have existed at the start of iteration t'. Otherwise, o_k was in another agent's bundle at t', and it, too, would be transferred to a different bundle in a later iteration.

Since the number of items is finite, this process must eventually lead to an item that was in $X_{i_0}^{t'}$, forming a transfer path starting at i at the start of iteration t' — a contradiction. □

Proposition 7.9. *Let i be an agent removed from the game at the start of iteration t'. Then, for all $t \geq t'$, i will not be included in any transfer path.*

Proof. First, note that $v_i(X_{i_0}^t) = 0$; otherwise, i would not have been removed at the start of iteration t'. Next, any agent j, who receives an item from i_0's bundle at iteration t, must be in $R(t)$. By Proposition 7.8, $v_i(X_j^t) = 0$, which means that i can not receive any item from j, who in $R(t)$, as compensation for another item from their own bundle. Thus, any transfer path in iteration t includes only agents in $R(t)$. □

Proposition 7.10. *Consider an iteration t and an agent $i \notin R(t)$, who was removed from the game at the start of iteration $t' < t$. Let $P = (i = i_1, \ldots, i_k)$ be a path in the weighted envy graph starting at i. If $cost_{X^t}(P) > cost_{X^{t'-1}}(P)$, then there must exist $j \in \{1, \ldots, k-1\}$ such that $i_j \notin R$ at t, $i_{j+1} \in R(t)$ at t, and $v_{i_j}(X_{i_{j+1}}^t) = 0$.*

Proof. Let $t > t'$ be the earliest iteration in which $cost_{X^t}(P) > cost_{X^{t'-1}}(P)$. This implies that there is at least one agent, say $1 \leq j' \leq k-1$, such that agent $i_{j'+1}$ has received a new item that $i_{j'}$ desires. In other words, $i_{j'+1}$ was part of a transfer path at the start of iteration t, and by Proposition 7.9, $i_{j'+1} \in R(t)$ (in particular, $j' \geq 2$).

Since $i_1 \notin R(t)$ and $i_{j'+1} \in R(t)$, there must exist some $1 \leq p < j'+1$ such that $i_p \notin R(t)$ at t and $i_{p+1} \in R(t)$. By Proposition 7.8, $v_{i_p}(X_{i_{p+1}}^t) = 0$. □

Proposition 7.11. *Let $i \notin R(t)$, be an agent who was removed from the game at the start of iteration $t' < t$. Then, for X^t the resulting allocation from iteration t, $\ell_i(X^t) \leq \frac{1}{w_{\min}}$.*

Proof. Denote by $P_i^{t'-1}, P_i^t$ the highest-cost paths starting from i at iterations $t'-1$ (before agent i removed) and t, correspondingly. In particular, $cost_{X^{t'-1}}(P_i^t) \leq cost_{X^{t'-1}}(P_i^{t'-1}) = \ell_i(X^{t'-1})$. Moreover, by Observation 7.5 we have $cost_{X^{t'-1}}(P_i^{t'-1}) \leq \frac{|X_{i_k}^{t'-1}|}{w_{i_k}} - \frac{|X_i^{t'-1}|}{w_i}$ when i_k is the last agent in $P_i^{t'-1}$. Combined with Lemma 7.6, this gives $\ell_i(X^{t'-1}) \leq \frac{1}{w_i}$. Therefore, if $cost_{X^t}(P_i^t) = \ell_i(X^t) \leq \ell_i(X^{t'-1})$, then we are done.

Assume now that $\ell_i(X^t) > \ell_i(X^{t'-1})$. Denote the path P_i^t by $(i = i_1, \ldots, i_k)$.

From Proposition 7.10, there exists $j \in \{1, \ldots, k-1\}$ such that $i_j \notin R(t)$, $i_{j+1} \in R(t)$ and $v_{i_j}(X_{i_{j+1}}^t) = 0$. Then, by Observation 7.5:

$$\begin{aligned} &\ell_i(X^t) = cost_{X^t}(i, ..., i_j) + cost_{X^t}(i_j, i_{j+1}) + cost_{X^t}(i_{j+1}, ..., i_k) \leq \\ &\leq \left(\frac{|X_{i_j}^t|}{w_{i_j}} - \frac{|X_i^t|}{w_i}\right) + \left(0 - \frac{|X_{i_j}^t|}{w_{i_j}}\right) + \ell_{i_{j+1}}(X^t) \leq \\ &\ell_{i_{j+1}}(X^t). \end{aligned} \tag{3}$$

Since $i_{j+1} \in R(t)$, it follows from (3) and Proposition 7.7 that

$$\ell_i(X^t) \leq \ell_{i_{j+1}}(X^t) \leq \frac{1}{w_{i_{j+1}}} \leq \frac{1}{w_{\min}}.$$

□

Theorem 7.12. *For additive binary valuations, Algorithm 3 computes a WEF-able allocation where the subsidy to each agent $i \in N$ is at most $\frac{w_i}{w_{\min}}$ in polynomial-time. Moreover, the total subsidy is bounded by $\frac{W}{w_{\min}} - 1$.*

Proof. Together Proposition 7.7 and Proposition 7.11 establish that for every $i \in N$ and $t \in [T]$, $\ell_i(X^t) \leq \frac{1}{w_{\min}}$. Along with Lemma 7.4, Algorithm 3 computes a WEF-able allocation X^T where the required subsidy per agent $i \in N$ is at most $\frac{w_i}{w_{\min}}$. As there is at least one agent who requires no subsidy (see Theorem 3.5), the total required subsidy is at most $\frac{W - w_{\min}}{w_{\min}} = \frac{W}{w_{\min}} - 1$.

We complete the proof of Theorem 7.12 by demonstrating that Algorithm 3 runs in polynomial-time.

We represent the valuations using a binary matrix A where $v_i(o_j) = 1 \iff A(i,j) = 1$. Hence, the allocation of items to the bundle of i_0 at line 1 can be accomplished in $O(mn)$ time.

At each iteration t of the while loop, either X_{i_0} or R reduced by 1, ensuring that the loop runs at most $m + n$ times.

Let T_v represent the complexity of computing the value of a bundle of items, and T_ϕ denote the complexity of computing the gain function. Both are polynomial in m.

According to Viswanathan and Zick (2023a), finding a transfer path starting from agent $i \in N$ (or determining that no such path exists) takes $O(T_v \log m)$. Removing agents at the start of each iteration incurs a complexity of $O(nT_v \log m)$. Furthermore, as stated in Viswanathan and Zick (2023a), identifying u requires $O(nT_v)$. Updating the allocation based on the transfer path, according to the same source, takes $O(m)$.

Thus, each iteration has a total complexity of $O(nT_v \log m + nT_v + T_v \log m + m) = O\left(nT_v \log m + m\right)$.

In conclusion, Algorithm 3 runs in $O\left((m+n)\left(nT_v \log m + m\right)\right)$, which is polynomial in both m and n. □

In Appendix A.3, we present a tighter bound that is closer to the lower bound, along with a detailed discussion on its tightness.

Notice that since the output allocation from Algorithm 3 is non-redundant, X maximizes the social welfare. Moreover, as shown in Example 7.2, X might not be *WEF(1,0)* (No matter which item is removed from i_2's bundle, i_1 still envies). However, it is *WEF(0,1)*.

Proposition 7.13. *For additive binary valuations, Algorithm 3 computes a WEF$(0,1)$ allocation.*

Proof. We prove by induction that at the end of each iteration $t \in [T]$ X^t satisfies WEF$(0,1)$. This means that for every $i, j \in N$, there exists a set of items $B \subseteq X_j^t$ of size at most 1 such that $\frac{v_i(X_i^t)+v_i(B)}{w_i} \geq \frac{v_i(X_j^t)}{w_j}$.

The claim is straightforward for the first iteration. We assume the claim holds for the $(t-1)$-th iteration and prove it for the t-th iteration. Note that $v_i(X_i^{t-1}) \leq v_i(X_i^t)$.

1. $X_j^t = X_j^{t-1}$ and $X_i^t = X_i^{t-1}$: the claim holds due to the induction step.

2. $v_i(X_j^t) = v_i(X_j^{t-1})$: This is the case where j was not included in a transfer path, or was included but was not the first agent in the path, and exchanged an item for a new one, both having the same value for i. By the induction assumption, there exists some singleton $B^{t-1} \subseteq X_j^{t-1}$ such that $\frac{v_i(X_i^{t-1})+v_i(B^{t-1})}{w_i} \geq \frac{v_i(X_j^{t-1})}{w_j} = \frac{v_i(X_j^t)}{w_j}$. There exists some singleton $B^t \subseteq X_j^t$, with $v_i(B^t) = v_i(B^{t-1})$. Hence, $\frac{v_i(X_i^t)+v_i(B^t)}{w_i} \geq \frac{v_i(X_i^{t-1})+v_i(B^{t-1})}{w_i} \geq \frac{v_i(X_j^{t-1})}{w_j} = \frac{v_i(X_j^t)}{w_j}$.

3. $v_i(X_j^t) < v_i(X_j^{t-1})$: This is the case where j was included in a transfer path but exchanged an item i values for an item i does not value. Then $\frac{v_i(X_i^t)+v_i(B)}{w_i} \geq \frac{v_i(X_i^{t-1})+v_i(B)}{w_i} \geq \frac{v_i(X_j^{t-1})}{w_j} > \frac{v_i(X_j^t)}{w_j}$ for a set $B \subseteq X_j^t$ of size at most 1.

4. $v_i(X_j^t) > v_i(X_j^{t-1})$: there are two subcases:

(a) If j is the first agent in the transfer path and received a new item o such that $v_i(o) = 1$, then $\frac{v_i(X_i^{t-1})+1}{w_i} \geq \frac{v_j(X_j^{t-1})+1}{w_j}$ due to the selection rule, and $\frac{v_j(X_j^{t-1})+1}{w_j} \geq \frac{v_i(X_j^{t-1})+1}{w_j}$ due to non-redundancy.

We can conclude that $\frac{v_i(X_i^t)+1}{w_i} \geq \frac{v_i(X_i^{t-1})+1}{w_i} \geq \frac{v_i(X_j^{t-1})+1}{w_j} = \frac{v_i(X_j^t)}{w_j}$. The claim holds for $B = \{o\} \subseteq X_j^t$.

(b) If j was not the first agent in the path, but exchanged an item that i does not value for an item o that i values, $v_i(o) = 1$. Let $t' < t$ represent the most recent iteration in which agent j was selected and received a new item. Note that $\frac{v_i(X_i^t)+1}{w_i} \geq \frac{v_i(X_i^{t'-1})+1}{w_i} \geq \frac{v_j(X_j^{t'-1})+1}{w_j}$ due to the selection rule.

Assume to the contrary that $\frac{v_i(X_i^t)+1}{w_i} < \frac{v_i(X_j^t)}{w_j}$. Then,

$$\frac{v_i(X_i^t)+1}{w_i} < \frac{v_i(X_j^t)}{w_j} \leq \frac{v_j(X_j^t)}{w_j} = \frac{v_j(X_j^{t'})}{w_j} = \frac{v_j(X_j^{t'-1})+1}{w_j} \leq \frac{v_i(X_i^{t'-1})+1}{w_i} \leq \frac{v_i(X_i^t)+1}{w_i},$$

a contradiction. Hence, $\frac{v_i(X_i^t)+1}{w_i} \geq \frac{v_i(X_j^t)}{w_j}$ and the claim holds for $B = \{o\} \subseteq X_j^t$.

□

7.1 Matroidal Valuation

Another valuation class, slightly more general than binary additive, is the class of *matroidal valuations*, also called *matroid rank valuations* (Barman and Verma (2020); Benabbou et al. (2021)). In this section we prove that the subsidy bound provided by Algorithm 3 for agents with binary additive valuations, which is $\frac{W}{w_{\min}} - 1$ (Theorem 7.12), is not applicable for agents with matroidal valuations. Specifically, we show a lower bound of $\frac{m}{n}\left(\frac{W}{w_{\min}} - n\right)$, which is linearly increasing with m.

A matroidal valuation is based on a matroid $\mathcal{F}$ over M.[5] Then, the value of each subset $A \subseteq M$ equals $\max_{F \in \mathcal{F}} |A \cap F|$.

Note that a matroidal valuation is submodular, but not necessarily additive. A binary additive valuation is a special case of a matroidal valuation, in which each agent i has $\mathcal{F}_i = \{F \mid F \subseteq B_i\}$, where $B_i := \{o \mid o \in M, v_i(o) = 1\}$.

[5]Formally, $\mathcal{F}$ is a family of subsets of M such that $\emptyset \in \mathcal{F}$, $F' \subseteq F \in \mathcal{F}$ implies $F' \in \mathcal{F}$, and for any $F, F' \in \mathcal{F}$ where $|F| > |F'|$, there exists an item $o \in F \setminus F'$ s.t. $F' \cup \{o\} \in \mathcal{F}$.

We demonstrate that the total subsidy bound derived by Algorithm 3 for agents with binary additive valuations might not hold for agents with matroidal valuations.

Theorem 7.14. *There exists an instance with matroidal valuations for which, in any WEF non-wasteful allocation, the subsidy for some agent $i \in N$ is at least $\frac{m}{n}(\frac{w_i}{w_{\min}} - 1)$, and the total subsidy is at least $\frac{m}{n}\left(\frac{W}{w_{\min}} - n\right)$.*

Proof. Assume there are n agents with weights $w_1 \leq \ldots \leq w_n$. There are $m = nk$ items, and the valuation of each agent i for bundle X_i is given by $\min(k, |X_i|)$, i,e., each agent values at most k items (We can assume that the feasible bundles are defined as $\mathcal{F} = \{F \mid F \subseteq M, |F| \leq k\}$).

In this setting, the only non-wasteful allocation is one in which each agent is allocated exactly k items. An agent $i \in N$ envies another agent $j \neq i \in N$ by amount given by $\frac{k}{w_j} - \frac{k}{w_i}$. Equivalently, the cost of any path $P = (i, \ldots, j)$ for some $j \neq i \in N$ is $k\left(\frac{1}{w_j} - \frac{1}{w_i}\right)$. The maximum-cost path starting from agent i ends at agent 1, and its cost is $k\left(\frac{1}{w_{\min}} - \frac{1}{w_i}\right)$. To eliminate envy, each agent $i \in N$ must receive a subsidy of $k\left(\frac{w_i}{w_{\min}} - 1\right) = \frac{m}{n}\left(\frac{w_i}{w_{\min}} - 1\right)$. The total subsidy required is

$$\sum_{1 \leq i \leq n} \frac{m}{n}\left(\frac{w_i}{w_{\min}} - 1\right) = \sum_{1 < i \leq n} \frac{m}{n}\left(\frac{w_i}{w_{\min}} - 1\right) = \frac{m}{n}\left(\frac{W - w_{\min}}{w_{\min}} - (n-1)\right) = \frac{m}{n}\left(\frac{W}{w_{\min}} - n\right).$$

□

Importantly, whereas the upper bound for binary additive valuations $\frac{W}{w_{\min}} - 1$ is independent of m, the lower bound for matroidal valuations is increasing with m.

8 Additive Valuations and Identical Items

With a slight abuse of notation, we denote by v_i the value of agent i to a single item. Thus, for an allocation X, if $|X_i| = m_i$, then $v_i(X_i) = v_i m_i$. We denote an allocation by a tuple of integers $(m_1, m_2, \ldots, m_n)$, representing the numbers of items allocated to the agents. Note that $m = m_1 + \cdots + m_n$.

In this section, for simplicity, we order the agents in descending order of their *value* rather than their weight, that is, we sort the agents so that $(V \geq)\ v_1 \geq v_2 \geq \ldots \geq v_n$.

We show matching upper and lower bounds on the subsidy per agent and on the total subsidy.

The following lemma states that weighted envy-freeability can be characterized by the weighted reassignment-stability condition for swapping only a pair of two agents.

Lemma 8.1. *For additive valuations with identical items, an allocation* $X = (m_1, m_2 \ldots, m_n)$ *is WEF-able if and only if for each* $1 \leq i, j \leq n$ *with* $v_i < v_j$ *we have* $\frac{m_i}{w_i} \leq \frac{m_j}{w_j}$.

Proof. We first show the only-if part. Assume that $(m_1, \ldots, m_n)$ is WEF-able. Then, by Theorem 3.3, it is weighted reassignment-stable. For the permutation that only swaps i and j, inequality (1) in the definition of weighted reassignment-stability implies that $\frac{v_i \cdot m_i}{w_i} + \frac{v_j \cdot m_j}{w_j} \geq \frac{v_i \cdot m_j}{w_j} + \frac{v_j \cdot m_i}{w_i}$. When $v_i < v_j$, this implies that $\frac{m_i}{w_i} \leq \frac{m_j}{w_j}$.

We now show the if part. WLOG, we can assume when $i > j$ and $v_i = v_j$, $m_i/w_i \geq m_j/w_j$ holds (otherwise we can rename agents' identifiers). We will show that for any $i > j$ we have

$$cost_X(i, j) \leq cost_X(i, i-1) + cost_X(i-1, i-2) + \cdots + cost_X(j+1, j) \qquad (4)$$

Once this is done, we can show that any cycle in the weighted envy-graph has a non-positive cost as follows. Let C be a cycle in the weighted envy-graph. We partition the set of edges of C into the two sets of "ascending" edges and "descending" edges:

$$E(C) = E_+(C) \cup E_-(C),$$

where

$$E_+(C) = \{(i, j) \in E(C) \mid i > j\} \text{ and } E_-(C) = \{(j, i) \in E(C) \mid i > j\}.$$

To show C has non-positive cost is to show

$$\sum_{(i,j) \in E_+(C)} cost_X(i, j) + \sum_{(j,i) \in E_-(C)} cost_X(j, i) \leq 0$$

The inequality (4) implies

$$\sum_{(i,j) \in E_+(C)} cost_X(i, j) \leq \sum_{(i,j) \in E_+(C)} (cost_X(i, i-1) + \cdots + cost_X(j+1, j)).$$

Let $E'_+(C)$ be a *multiset* of edges of C defined as

$$E'_+(C) = \cup_{(i,j) \in E_+(C)} \{(i, i-1), (i-1, i-2), \ldots, (j+1, j)\}.$$

Then it suffices to show that

$$\sum_{(i,j) \in E'_+(C)} cost_X(i, j) + \sum_{(j,i) \in E_-(C)} cost_X(j, i) \leq 0.$$

Now, to each edge $(j_0, i_0) \in E_-(C)$, we assign the subset of edges $\{(i_0, i_0 - 1), \ldots, (j_0 + 1, j_0)\}$ from $E'_+(C)$; this assignment partitions $E'_+(C)$ into disjoint subsets, since C is a cycle.

Then it suffices to show that the sum of costs of "ascending" edges assigned to $(j_0, i_0) \in E_-(C)$ plus the cost of (j_0, i_0) itself (i.e., $cost_X(i_0, i_0 - 1) + cost_X(i_0 - 1, i_0 - 2) + \cdots + cost_X(j_0 + 1, j_0) + cost_X(j_0, i_0)$) is non-positive, since $\sum_{(i,j)\in E'_+(C)} cost_X(i,j) + \sum_{(j,i)\in E_-(C)} cost_X(j,i)$ is the sum of these values.

Indeed, we have

$$\begin{aligned}
&cost_X(i, i-1) + \cdots + cost_X(j+1, j) + cost_X(j, i) \\
&= v_i\left(\frac{m_{i-1}}{w_{i-1}} - \frac{m_i}{w_i}\right) + \cdots + v_{j+1}\left(\frac{m_j}{w_j} - \frac{m_{j+1}}{w_{j+1}}\right) + v_j\left(\frac{m_i}{w_i} - \frac{m_j}{w_j}\right) \\
&\le v_j\left(\frac{m_{i-1}}{w_{i-1}} - \frac{m_i}{w_i}\right) + \cdots + v_j\left(\frac{m_j}{w_j} - \frac{m_{j+1}}{w_{j+1}}\right) + v_j\left(\frac{m_i}{w_i} - \frac{m_j}{w_j}\right) \\
&= v_j\left(\frac{m_j}{w_j} - \frac{m_i}{w_i}\right) + v_j\left(\frac{m_i}{w_i} - \frac{m_j}{w_j}\right) = 0,
\end{aligned}$$

where we use the fact that $v_i \ldots v_{j+1} \le v_j$ and $\frac{m_k}{w_k} \ge \frac{m_{k+1}}{w_{k+1}}$ for $i \le k \le j-1$ in the inequality. Therefore, any cycle in the weighted envy-graph has a non-positive cost, and the allocation is WEF-able by Theorem 3.3.

It remains to prove the inequality (4). We show this by induction on $j - i$. If $i - j = 1$, then $cost_X(i,j) = cost_X(i, i-1)$ holds trivially.

Assume $i - j > 1$. By the inductive hypothesis, we have

$$cost_X(i, j+1) \le cost_X(i, i-1) + cost_X(i-1, i-2) + \cdots + cost_X(j+2, j+1).$$

Hence, it suffices to show that $cost_X(i,j) \le cost_X(i, j+1) + cost_X(j+1, j)$. Indeed,

$$\begin{aligned}
cost_X(i,j) &= v_i\left(\frac{m_j}{w_j} - \frac{m_i}{w_i}\right) = v_i\left(\frac{m_j}{w_j} - \frac{m_{j+1}}{w_{j+1}}\right) + v_i\left(\frac{m_{j+1}}{w_{j+1}} - \frac{m_i}{w_i}\right) \\
&\le v_{j+1}\left(\frac{m_j}{w_j} - \frac{m_{j+1}}{w_{j+1}}\right) + v_i\left(\frac{m_{j+1}}{w_{j+1}} - \frac{m_i}{w_i}\right) = cost_X(j+1, j) + cost_X(i, j+1),
\end{aligned}$$

where we use the fact that $v_i \le v_{j+1}$ and $\frac{m_{j+1}}{w_{j+1}} \le \frac{m_j}{w_j}$ in the inequality. □

Using this Lemma 8.1, we first prove a lower bound.

Theorem 8.2. *Even with additive valuations and identical items, it is impossible to guarantee that the subsidy per agent is smaller than* $V w_i \sum_{1\le j<i} \frac{1}{w_j}$*, or that the total subsidy is smaller than* $V \sum_{2\le i\le n} \left(w_i \sum_{1\le j<i} \frac{1}{w_j}\right)$.

Remark 8.3. Theorem 5.2 shows a lower bound for general additive valuations, which is worse than the lower bound of Theorem 8.2 for identical additive valuations. This is because the bound of Theorem 5.2 holds for any weight vector, whereas the bound of Theorem 8.2 holds only for weight vectors where weights are not integer multiples of adjacent weights (see the proof).

Proof. We construct an instance in which $v_i = V + (n-i) \cdot \epsilon$ for some small $\epsilon > 0$. Note that $v_1 > \cdots > v_n$. The weights are in the same order, that is, $w_{\max} = w_n \geq \cdots \geq w_1 = w_{\min} \geq 2$.

The smallest weight w_n is an integer. For any $i \in \{2, \dots, n\}$, $w_i = (k_i - \epsilon)\, w_{i-1}$, for some integer $k_i \geq 2$ and some small $\epsilon > 0$.

We aim to determine the number of items $m = \sum_{i \in N} m_i$ such that agent n receives exactly w_n items, and each agent $i \in \{2, \dots, n\}$ envies agent $i-1$.

By the WEF property, each agent i must achieve a total utility of approximately $w_i v_i \sim w_i V$. Therefore, the total subsidy required is given by:

$$(W - m)\, V = \sum_{i \in N} (w_i - m_i)\, V,$$

The term mV accounts for the utility generated by allocating m items to the agents.

Our goal is to find values $m_1, \dots, m_n$ such that agent i receives a subsidy of $(w_i - m_i)\, V$.

First, note that for each $i \in \{2, \dots, n\}$, we have $k_i = \frac{w_i}{w_{i-1}} + \epsilon \sim \frac{w_i}{w_{i-1}}$, where ϵ is small.

We proceed by allocating items to agents as follows:

1. Agent 1 receives $m_1 = w_1$ items, as assumed, and receives no subsidy.
2. Agent 2 receives $m_2 = k_2\,(m_1 - 1) \sim \frac{w_2}{w_1}(m_1 - 1) = w_2 - \frac{w_2}{w_1}$ items. The subsidy for agent 2 is $w_2 v_2 \left(\frac{m_1}{w_1} - \frac{m_2}{w_2}\right) \sim V w_2 \frac{1}{w_1}$.
3. Agent 3 receives $m_3 = k_3\,(m_2 - 1) \sim \frac{w_3}{w_2}(m_2 - 1) = w_3 - \frac{w_3}{w_n} - \frac{w_3}{w_2}$. The subsidy for agent 3 is $w_3 \left(v_2 \left(\frac{m_1}{w_1} - \frac{m_2}{w_2}\right) + v_3 \left(\frac{m_2}{w_2} - \frac{m_3}{w_3}\right)\right) \sim V w_3 \left(\frac{1}{w_2} + \frac{1}{w_1}\right)$.
4. In general, for any agent $i \in \{2, \dots, n\}$, the number of items they receive is: $m_i = k_i\,(m_{i-1} - 1) \sim \frac{w_i}{w_{i-1}}(m_{i-1} - 1) = w_i - \left(w_i \sum_{1 \leq j < i} \frac{1}{w_j}\right)$. The subsidy for agent i is $\sim V w_i \sum_{1 \leq j < i} \frac{1}{w_j}$.

By Lemma 8.1, we can not remove any item from any agent $i \in \{1, \dots, n-1\}$ because:

$$\frac{m_{i+1}}{w_{i+1}} = \frac{k_{i+1}\,(m_i - 1)}{w_{i+1}} > \frac{w_{i+1}}{w_i}\frac{m_i - 1}{w_{i-1}} = \frac{m_i - 1}{w_i}.$$

ALGORITHM 4: Weighted Sequence Protocol For Additive Valuations and identical items

Input: Instance $(N, M, v, \mathbf{w})$ with additive valuations and identical items.
Output: WEF-able allocation $(m_1, \ldots, m_n)$ with required subsidy of at most $V w_i \sum_{1 \le j \le i} \frac{1}{w_j}$ per agent, and total subsidy at most $V \sum_{2 \le i \le n} \left(w_i \sum_{1 \le j \le i} \frac{1}{w_j} \right)$.

```
m_i ← 0 for each i ∈ N;
Sort the agents such that v_1 ≥ ··· ≥ v_n.;
for o : 1 to m do
    Let N' ← {i ∈ {2,...,n} | (1+m_i)/w_i ≤ m_{i-1}/w_{i-1}} ∪ {1} /* We always have 1 ∈ N'  */
    Let u ← max_{i∈N'} i;
    m_u ← m_u + 1;
end
return (m_1,...,m_n)
```

This inequality ensures that reducing the number of items allocated to any agent would violate the WEF condition.

Thus, this allocation satisfies WEF with the smallest possible subsidy, as required, and the total required subsidy is

$$V \sum_{2 \le i \le n} \left(w_i \sum_{1 \le j < i} \frac{1}{w_j} \right)$$

□

We now establish a matching upper bound on the subsidy. Our algorithm is defined in Algorithm 4. First, the agents are sorted by their valuations such that $v_1 \ge v_2 \ge \ldots \ge v_n$. The number of items of each agent i is initialized to $m_i = 0$. Then, while there are unallocated items, the algorithm finds the agent $i \in \{2, \ldots, n\}$ with the maximum index such that $\frac{1+m_i}{w_i} \le \frac{m_{i-1}}{w_{i-1}}$. If no such agent exists, the algorithm, selects agent 1. The chosen agent $i \in N$ receives a new item, i.e., m_i increases by 1.

Theorem 8.4. *Algorithm 4 outputs a WEF allocation with subsidy at most* $w_i V \sum_{1 \le j \le i} \frac{1}{w_j}$ *for each agent, and total subsidy at most* $V \sum_{2 \le i \le n} \left(w_i \sum_{1 \le j \le i} \frac{1}{w_j} \right)$.

Proof. We first prove that the allocation output by Algorithm 4 is WEF-able by Lemma 8.1. The definition of N' ensures that, if agent $u \ge 2$ receives an item, then after the update, $\frac{m_u}{w_u} \le \frac{m_{u-1}}{w_{u-1}}$. Therefore, this condition holds throughout the algorithm for all adjacent pairs of agents. By transitivity, $\frac{m_u}{w_u} \le \frac{m_j}{w_j}$ for all $u > j \ge 1$.

We prove that while running the algorithm, the cost of the highest-cost path is always bounded by $V \sum_{1 \le j \le i} \frac{1}{w_j}$.

From the proof of Lemma 8.1, WLOG, we can assume the highest-cost path from each agent i is $P_i = (i, i-1, \ldots, 1)$, since $cost_X(j, j-1)$ is non-negative, $cost_X(j,k)$ where $j < k$ is non-positive, and there exists no positive cycle. Also, $cost_X(i,j)$ where $j \geq i+2$ is smaller than or equal to $\sum_{i<k\leq j} cost_X(k, k-1)$. In other words, $\ell_i(X) = \sum_{1<j\leq i} cost_X(j, j-1)$.

Now, we prove for each agent $i \in N$, $\ell_i(X) \leq V \sum_{1\leq j\leq i} \frac{1}{w_j}$. The proof is by induction over the iteration.

When there is no allocated item, this must be true. Now, suppose that the algorithm allocates an item to agent u. We consider three cases.

<u>Case 1:</u> $u > i$. The cost of the highest-cost path starting at i is not affected by allocating the item to u in this iteration, as $\ell_i(X) = \sum_{1<j\leq i} cost_X(j, j-1)$. By the induction assumption, $\ell_i(X) \leq V \sum_{1<j\leq i} \frac{1}{w_j}$.

<u>Case 2:</u> $1 > i \geq u$. Then $cost_X(u+1, u)$ increases by at most $\frac{v_{u+1}}{w_u}$ (or does not increase at all if $u = i$), while $cost_X(u, u-1)$ decreases by $v\frac{v_u}{w_u}$. The costs of all other edges remain unchanged. Thus, the total path cost weakly decreases. By the induction assumption, $\ell_i(X) \leq V \sum_{1<j\leq i} \frac{1}{w_j}$.

<u>Case 3:</u> $u = 1$. Then, by the fact that the algorithm chose 1, before the allocation we must have $\frac{1+m_i}{w_i} > \frac{m_{i-1}}{w_{i-1}}$ for each $i > 1$. Otherwise, the algorithm would have chosen some agent $i > 1$ to allocate the item.

Thus, for each $i > 1$, before the allocation,

$$cost_X(i, i-1) = \frac{v_i m_{i-1}}{w_{i-1}} - \frac{v_i m_i}{w_i} < v_i\left(\frac{1+m_i}{w_i} - \frac{m_i}{w_i}\right) = \frac{v_i}{w_i} \leq \frac{V}{w_i}.$$

This implies that before allocating the item to agent 1: $\ell_i(X) = \sum_{1<j\leq i} cost_X(j, j-1) < \sum_{1<j\leq i} \frac{1}{w_j}$. After allocating the item to agent 1, the path cost increases by at most $\frac{V}{w_1}$ Therefore, $\ell_i(X) < V \sum_{1\leq j\leq i} \frac{1}{w_j}$.

Now we can conclude that the cost of a highest-cost path is bounded by $V \sum_{1\leq j\leq i} \frac{1}{w_j}$. Thus, the subsidy for agent i is at most $w_i \left(\sum_{1\leq j\leq i} \frac{1}{w_j}\right)$.

Note that agent 2 receives a subsidy of $p_2 = v_2 w_2 \left(\frac{m_1}{w_1} - \frac{m_2}{w_2}\right)$, which ensures that

$$\frac{v_2 m_2 + p_2}{w_2} = \frac{v_2 m_2}{w_2} + \frac{v_2 m_1}{w_1} - \frac{v_2 m_2}{w_2} = \frac{v_2 m_1}{w_1}.$$

Thus, agent 1 must receive no subsidy, because otherwise, the subsidy p_2 would not eliminate the envy of agent 2 towards agent 1.

Therefore, the total subsidy is at most $V \sum_{2\leq i\leq n} \left(w_i \sum_{1\leq j\leq i} \frac{1}{w_j}\right)$. □

Brustle et al. (2020) proved that when the weights are equal, the total subsidy required for agents with additive valuations is bounded by $(n-1)V$. The following proposition demonstrates that with different weights, even when the weights are nearly equal, the total subsidy bound can be linear in n^2.

Proposition 8.5. *Even when the weights are nearly equal, the total subsidy lower bound for agents with additive valuations and identical items is linear in n^2.*

Proof. Assume that $w_1 = 2n$ and $w_i = w_{i+1} - \epsilon$ for each $i \in 2, \ldots, n-1$, with $\epsilon > 0$. Note that all weights are positive, nearly equal, and $w_1 > w_2 > \ldots > w_n$.

Let $m_1 = w_1$ and $m_i = m_{i-1} - 1$ for $i \in \{2, \ldots, n\}$. Let $m := \sum_i m_i = n \cdot (3n+1)/2$.

By Lemma 8.1, agent $i \in 2, \ldots, n$ can receive at most $m_i = m_{i-1} - 1$ items. Thus, the subsidy for agent $i \in \{2, \ldots, n\}$ is at least $w_i V \sum_{1 \le j < i} \frac{1}{w_j} \sim (i-1)V$. Hence, the total subsidy is at least $\sim V \sum_{2 \le i \le n} i - 1 = V \sum_{1 \le i \le n-1} i = \frac{(n-1)(n-2)}{2} V = \Omega(n^2 V)$. □

8.1 Optimal Algorithm for a Special Case of Additive Valuations and Identical Items

In most settings studied in this paper, computing the optimal subsidy is NP-hard. However, in the special case of identical items and additive valuations, we have a polynomial-time algorithm.

Here, for convenience, we assume that $v_1 < v_2 < \cdots < v_n$, contrary to the assumption in the previous subsection. We propose an algorithm based on dynamic programming that computes an allocation with the minimum total subsidy.

We say that an allocation $(m_1, \ldots, m_n)$ is *feasible* if $m_1 + \cdots + m_n = m$.

For $1 \le i \le n$, $0 \le j \le m$, and $0 \le m_i \le m$, let $T(i, j, m_i)$ be defined as the minimum total subsidy when

- the agents are restricted to $1, 2, \ldots, i$,
- the number of items is j, and
- the number of items allocated to agent i is m_i.

Then the minimum total subsidy we want to compute equals $\min_{1 \le m_i \le m} T(n, m, m_i)$.

Lemma 8.6. *The following recursive formula holds.*

$$T(i, j, m_i) = \begin{cases} 0 & i = 1 \text{ and } j = m_i \\ \min\limits_{0 \le m_{i-1} \le \min\left(j - m_i, \frac{w_{i-1}}{w_i} m_i\right)} \left\{ T(i-1, j-m_i, m_{i-1}) + \left(\sum_{i'=1}^{i-1} w_{i'} \right) \cdot \left(\frac{m_i}{w_i} - \frac{m_{i-1}}{w_{i-1}} \right) v_{i-1} \right\} & i \ge 2 \text{ and } \frac{w_i}{\sum_{i'=1}^{i} w_{i'}} j \le m_i \le j \\ \infty & \text{otherwise}. \end{cases} \tag{5}$$

Proof. When $i = 1$, i.e., the number of agents in the market is one, an allocation is feasible iff $j = k$, and the total subsidy is zero for the feasible allocation.

Assume that $i \geq 2$. when agent i is added to the market with $i-1$ agents, if m_i (resp., m_{i-1}) items are allocated to i (resp., $i-1$), then the cost of edge $(i-1, i)$ in the weighted-envy graph is $\left(\frac{m_i}{w_i} - \frac{m_{i-1}}{w_{i-1}}\right) v_{i-1}$. Since a highest-cost path from agent $i'(< i)$ is $(i', i'+1, \ldots, n)$ from the proof of Theorem 8.4, $\sum_{i'=1}^{i-1} w_{i'}$ times the cost of edge $(i-1, i)$ is added to the total subsidy. Moreover, since the cost of the edges before i does not affect any highest-cost path path starting from i, allocations for agents $i' \leq i-2$ achieving $T(i-1, j-m_i, m_{i-1})$ do not affect the subsidies for agents $i'' \geq i-1$. Hence, one can optimize the subsidies for agents $i' \leq i-2$ in $T(i-1, j-k, k')$ and Equation 5 is correct.

We note that by Lemma 8.1, $\frac{m_i}{w_i} \geq \frac{\sum_{i'=1}^{i} m_{i'}}{\sum_{i'=1}^{i} w_{i'}}$. Hence, if $m_i < \frac{w_i}{\sum_{i'=1}^{i} w_{i'}} j$, then for an allocation to be WEF-able, $\sum_{i'=1}^{i} m_{i'}$ should be less than j and thus there exists no WEF-able and feasible allocation. Therefore, we require $m_i \geq \frac{w_i}{\sum_{i'=1}^{i} w_{i'}} j$ in the second case of Equation 5. Moreover, for an allocation to be WEF-able, m_{i-1} should be less than or equal to $\frac{w_{i-1}}{w_i} m_i$ and thus we require $m_{i-1} \leq \frac{w_{i-1}}{w_i} m_i$ in the second case of Equation 5. □

Theorem 8.7. *There exists a polynomial time algorithm to compute an allocation with the minimum total subsidy for additive valuations and identical items if the valuations of each agent are all different.*

Proof. The minimum subsidy equals $\min_{1 \leq k \leq m} T(n, m, k)$ from Lemma 8.6. We can also compute an allocation achieving the minimum by keeping the value m_{i-1} attaining the minimum in Equation 5. The size of the table T is $\mathrm{O}(nm^2)$, and it takes $\mathrm{O}(m)$ time to fill each cell of the table. Therefore, the total running time is $\mathrm{O}(nm^3)$. □

9 Monetary Weighted Envy-Freeness

In practice, the available subsidy may be smaller than what's required for WEF. A natural question is what relaxation of fairness can be achieved. One solution is to allocate the subsidy only to agents whom nobody envies, preventing additional envy. This motivates the following concept:

Definition 9.1. An outcome $(X, \mathbf{p})$ is called *monetarily weighted envy-free (MWEF)* if $p_j = 0$ for all $j \in N$ such that some agent $i \in N$ has weighted envy towards j.

Note that the definition avoids the issue of whether the indivisible item allocation is fair. MWEF formalizes the idea that a limited subsidy is used effectively to improve fairness, regardless of whether the allocation X is "good" or "bad". Denote the total amount of money available for subsidy by d. If $d = 0$, so there is no subsidy at all ($p_i = 0$ for all $i \in N$), then the allocation is vacuously MWEF. Also, every WEF allocation (with or without subsidy) is MWEF.

Our main result in this section is that MWEF can be achieved using any WEF-able allocation and any total subsidy amount d. As far as we know, this algorithm is new even for the unweighted setting.

Theorem 9.2. *There is a polynomial-time algorithm that, for any instance with monotone valuations, given any WEF-able allocation X and any amount of money d, finds a subsidy vector $\mathbf{p}$ with $\sum_i p_i = d$, such that $(X, \mathbf{p})$ is MWEF.*

Proof. For any given outcome $(X, \mathbf{p})$, the corresponding *weighted envy-graph respecting the subsidy*, denoted $G_{X,w,\mathbf{p}}$, is a complete directed graph with vertex set N. For any pair of agents $i, j \in N$, $cost_X(i, j) = \frac{v_i(X_j)+p_j}{w_j} - \frac{v_i(X_i)+p_i}{w_i}$, represents the envy agent i has for agent j under $(X, \mathbf{p})$.

Let $\ell_i(X)$ be the maximum cost of any path in the weighted envy-graph $G_{X,w}$ that starts from i.

If the total money d is at exactly $\sum_{i \in N} \ell_i(X) \cdot w_i$, we can let each agent i's subsidy be $p_i = \ell_i(X) \cdot w_i$. The outcome is WEF by Theorem 3.5, hence also MWEF.

If $d > \sum_{i \in N} \ell_i(X) \cdot w_i$, we first allocate $p_i = \ell_i(X) \cdot w_i$, and then allocate the surplus amount in proportion to the weights of the agents, which is WEF as well.

The challenging case is if $d < \sum_{i \in N} \ell_i(X) \cdot w_i$. In this case, we initialize $p_i = 0$ for all $i \in N$, and then gradually increase the subsidies of some agents as follows.

Let $\ell_i(X, \mathbf{p})$ be the maximum cost of any path in the weighted envy-graph $G_{X,w,\mathbf{p}}$ that starts at i. We identify the set of agents N^* who have the highest $\ell_i(X, \mathbf{p})$.

Note that there is no agent j outside N^* who has zero or positive edge to any agent in N^* because if this is the case, j would be in N^*. Therefore, any agent $i \in N^*$ can be given a tiny amount of money without leading some agent outside N^* to become envious. Moreover, no agent $i \in N^*$ has a strictly positive edge to $k \in N^*$ or else k would not be a part of N^*.

When we allocate the money in proportion to the weights, all the $\ell_i(X, \mathbf{p})$ for $i \in N^*$ decrease at the same rate (so they remain maximum), and $\ell_j(X, \mathbf{p})$ for $j \notin N^*$ might increase.

As we do this, the set N^* may increase. Eventually, all the money is allocated. □

10 Experiments

In this section, we compare the minimum subsidy required for weighted envy-free allocation and the subsidy obtained by our proposed algorithms, along with their theoretical guarantees. We generate synthetic data for our experiments. We consider $n \in \{5, 8, 10\}$

agents and choose the number of items $m \in \{n, 2n, 3n, 4n, 5n\}$ and fix the weight vector $\mathbf{w} = (1, 2, \ldots, n)$. For each agent-item pair $(i, o) \in N \times M$, the valuation $v_i(o) \in \{\text{Discrete Uniform}(5,6), \text{Bernoulli}(0.5)\}$ is randomly generated.[6] We assume the additive valuations.

For each realization of the random instance, we solved the following integer linear programming (ILP) problem using the Gurobi Optimizer solver (version 11.0.3) to compute the minimum subsidy and also we computed the total subsidy obtained by our Algorithms 1-4. We repeated the experiment 50 times and reported the average total subsidy in Table 2 - 5.

$$\begin{aligned} &\min \sum_{i \in N} p_i \\ \text{s.t.} \quad &\sum_{o \in M} \frac{v_i(o) x_{i,o} + p_i}{w_i} \geq \sum_{o \in M} \frac{v_i(o) x_{j,o} + p_j}{w_j} \; \forall i, j \in N \\ &\sum_{i \in N} x_{i,o} = 1 \; \forall o \in M \\ &x_{i,o} \in \{0, 1\} \; \forall o \in M \\ &p_i \geq 0 \; \forall i \in N. \end{aligned}$$

[6]Discrete Uniform(5, 6) meaning we uniformly sampled from the set $\{5, 6\}$ and Bernoulli(0.5) means values are sampled from the set {0,1} each with probability 0.5.

Table 2: The table shows the minimum subsidies obtained by solving the ILP problem, subsidies obtained by Algorithm 1, and subsidies theoretically guaranteed by Algorithm 1 for $n \in \{5, 8, 10\}$ respectively. The valuation function is $v_i(o) \sim$ Discrete Uniform$(5, 6)$.

Number of items	**Total subsidy**		
	Algorithm 1	**Minimum**	**Theoretical bound**
5	62.5	10.3	84
10	35.02	7.615	84
15	7.84	2.085	84
20	55.06	3.42	84
25	29.2	5.555	84

Number of items	**Total subsidy**		
	Algorithm 1	**Minimum**	**Theoretical bound**
8	171.7800	15.78	210
16	128.24	7.9529	210
24	84.06	9.6214	210
32	40.08	4.5557	210
40	176.1	3.7569	210

Number of items	**Total subsidy**		
	Algorithm 1	**Minimum**	**Theoretical bound**
10	275	14.7583	324
20	220.24	14.3654	324
30	165.06	11.476	324
40	109.9	10.2745	324
50	55.06	3.5088	324

Table 3: The table shows the minimum subsidies obtained by solving the ILP problem, subsidies obtained by Algorithm 2, and subsidies theoretically guaranteed by Algorithm 2 for $n \in \{5, 8, 10\}$ respectively. The valuation function is $v(o) \sim$ Discrete Uniform$(1, 2)$.

Number of items	**Total subsidy**		
	Algorithm 2	**Minimum**	**Theoretical bound**
5	3.515	2.17	8
10	4.24	1.455	8
15	3.85	2.005	8
20	4.02	1.68	8
25	4.205	2.14	8

Number of items	**Total subsidy**		
	Algorithm 2	**Minimum**	**Theoretical bound**
8	6.5531	3.9654	14
16	6.9571	3.6863	14
24	7.7911	2.1414	14
32	6.0966	3.9274	14
40	6.6254	3.8506	14

Number of items	**Total subsidy**		
	Algorithm 2	**Minimum**	**Theoretical bound**
10	8.5921	4.4552	18
20	9.5916	4.5768	18
30	8.9475	5.843	18
40	9.1292	3.9327	18
50	8.8797	4.9171	18

Table 4: The table shows the minimum subsidies obtained by solving the ILP problem, subsidies obtained by Algorithm 3, and subsidies theoretically guaranteed by Algorithm 3 for $n \in \{5, 8, 10\}$ respectively. The valuation function is $v_i(o) \sim \text{Bernoulli}(0.5)$.

Number of items	**Total subsidy**		
	Algorithm 3	**Minimum**	**Theoretical bound**
5	1.69033	1.48699	14
10	0.98299	0.23533	14
15	0.370666	0	14
20	0.29333	0	14
25	0.422	0	14

Number of items	**Total subsidy**		
	Algorithm 3	**Minimum**	**Theoretical bound**
8	3.1364	2.4306	35
16	1.8120	0.5452	35
24	1.0444	0.0688	35
32	1.1500	0	35
40	0.2393	0	35

Number of items	**Total subsidy**		
	Algorithm 3	**Minimum**	**Theoretical bound**
10	3.5305	3.0417	44
20	3.9967	0.7505	44
30	1.9807	0.1652	44
40	0.9708	0	44
50	2.2950	0	44

Table 5: The table shows the minimum subsidies obtained by solving the ILP problem, subsidies obtained by Algorithm 4, and subsidies theoretically guaranteed by Algorithm 4 for $n \in \{5, 8, 10\}$ respectively. The valuation function is $v_i(o) \sim$ Discrete Uniform$(5, 6)$. Here, an agent has valuation identical for all items.

Number of items	**Total subsidy**		
	Algorithm 4	**Minimum**	**Theoretical bound**
5	70.8417	26.1153	169.5
10	98.3267	23.2863	169.5
15	85.0533	0	169.5
20	98.8933	25.0247	169.5
25	102.16	22.494	169.5

Number of items	**Total subsidy**		
	Algorithm 4	**Minimum**	**Theoretical bound**
8	228.1196	51.0788	497.0574
16	265.4938	59.0252	497.0574
24	274.1384	50.6484	497.0574
32	324.5231	20	497.0574
40	344.4849	45.7897	497.0574

Number of items	**Total subsidy**		
	Algorithm 4	**Minimum**	**Theoretical bound**
10	374.8001	75.628	825.5598
20	413.9721	104.0536	825.5598
30	489.8345	93.5287	825.5598
40	496.2941	68.3228	825.5598
50	529.3542	25	825.5598

From Tables 2-5, we observe that our algorithm consistently provides a lower average subsidy than theoretically obtained bound, while it is larger than the ILP computed minimum subsidy.

11 Future Work

Although several important techniques from the unweighted setting do not work in the weighted setting, we have managed to develop new techniques and used them to prove that WEF allocations with subsidies can be computed for agents with general monotone valuations. We even proved a worst-case upper bound on the amount of subsidy required to attain WEF. Our bounds are tight for general monotone valuations, superadditive and supermodular valuations, and identical additive valuations; however, for additive and binary valuations our bounds are not tight. Tightening these bounds is one of the main problems left open by the present paper.

Preliminary simulation experiments (Section 10) show that our algorithms require less subsidy than the worst-case bound, but more than the optimal amount. We plan to perform more comprehensive experiments in the future.

Moreover, it remains an open problem to find a WEF-able allocation for additive valuations when the ratios between the agents' weights are non-integer, as well as to find a WEF-able allocation for identical items with a minimum subsidy.

References

Haris Aziz. 2021. Achieving envy-freeness and equitability with monetary transfers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5102–5109.

H. Aziz, H. Chan, and B. Li. 2019. Weighted Maxmin Fair Share Allocation of Indivisible Chores. In *Proceedings of the 28h International Joint Conference on Artificial Intelligence (IJCAI)*.

Haris Aziz, Aditya Ganguly, and Evi Micha. 2023. Best of Both Worlds Fairness under Entitlements. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 941–948.

Haris Aziz, Xin Huang, Kei Kimura, Indrajit Saha, Zhaohong Sun, Mashbat Suzuki, and Makoto Yokoo. 2024. Weighted Envy-free Allocation with Subsidy. *arXiv preprint arXiv:2408.08711* (2024).

Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2021. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5119–5126.

Moshe Babaioff, Tomer Ezra, and Uriel Feige. 2024. Fair-share allocations for agents with arbitrary entitlements. *Mathematics of Operations Research* 49, 4 (2024), 2180–2211.

S Barman, A Krishna, Y Narahari, and S Sadhukan. 2022. Achieving Envy-Freeness with Limited Subsidies under Dichotomous Valuations. In *IJCAI International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 60–66.

Siddharth Barman and Paritosh Verma. 2020. Existence and computation of maximin fair allocations under matroid-rank valuations. *arXiv preprint arXiv:2012.12710* (2020).

Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. 2021. Finding fair and efficient allocations for matroid rank valuations. *ACM Transactions on Economics and Computation* 9, 4 (2021), 1–41.

Steven J Brams and Alan D Taylor. 1996. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press.

Johannes Brustle, Jack Dippel, Vishnu V Narayan, Mashbat Suzuki, and Adrian Vetta. 2020. One dollar each eliminates envy. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 23–39.

Ioannis Caragiannis and Stavros D Ioannidis. 2021. Computing envy-freeable allocations with limited subsidies. In *International Conference on Web and Internet Economics*. Springer, 522–539.

Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. 2021a. Weighted envy-freeness in indivisible item allocation. *ACM Transactions on Economics and Computation (TEAC)* 9, 3 (2021), 1–39.

Mithun Chakraborty, Ulrike Schmidt-Kraepelin, and Warut Suksompong. 2021b. Picking sequences and monotonicity in weighted fair division. *Artificial Intelligence* 301 (2021), 103578.

Mithun Chakraborty, Erel Segal-Halevi, and Warut Suksompong. 2022. Weighted fairness notions for indivisible items revisited. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4949–4956.

Edward H Clarke. 1971. Multipart pricing of public goods. *Public choice* (1971), 17–33.

Peter C Cramton, Yoav Shoham, Richard Steinberg, and Vernon L Smith. 2006. *Combinatorial auctions*. Vol. 1. MIT press Cambridge.

Gabrielle Demange, David Gale, and Marilda Sotomayor. 1986. Multi-item auctions. *Journal of political economy* 94, 4 (1986), 863–872.

Jack Edmonds and Richard M Karp. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 19, 2 (1972), 248–264.

Duncan Karl Foley. 1966. *Resource allocation and the public sector*. Yale University.

Hiromichi Goko, Ayumi Igarashi, Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, Yu Yokoi, and Makoto Yokoo. 2024. A fair and truthful mechanism with limited subsidy. *Games and Economic Behavior* 144 (2024), 49–70.

Andrew V Goldberg, Éva Tardos, and Robert Tarjan. 1989. *Network flow algorithm*. Technical Report. Cornell University Operations Research and Industrial Engineering.

Andrew V Goldberg and Robert E Tarjan. 1989. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM (JACM)* 36, 4 (1989), 873–886.

Theodore Groves. 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society* (1973), 617–631.

Daniel Halpern and Nisarg Shah. 2019. Fair division with subsidy. In *Algorithmic Game Theory: 12th International Symposium, SAGT 2019, Athens, Greece, September 30–October 3, 2019, Proceedings 12*. Springer, 374–389.

Martin Hoefer, Marco Schmalhofer, and Giovanna Varricchio. 2024. Best of both worlds: Agents with entitlements. *Journal of Artificial Intelligence Research* 80 (2024), 559–591.

Yasushi Kawase, Kazuhisa Makino, Hanna Sumita, Akihisa Tamura, and Makoto Yokoo. 2024. Towards optimal subsidy bounds for envy-freeable allocations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9824–9831.

Noga Klein Elmalem, Rica Gonen, and Erel Segal-Halevi. 2024. Weighted Envy Freeness With Bounded Subsidies. *arXiv preprint arXiv:2411.12696* (2024).

Shengxin Liu, Xinhang Lu, Mashbat Suzuki, and Toby Walsh. 2024. Mixed fair division: A survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22641–22649.

Luisa Montanari, Ulrike Schmidt-Kraepelin, Warut Suksompong, and Nicholas Teh. 2024. Weighted envy-freeness for submodular valuations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9865–9873.

Szilvia Pápai. 2003. Groves sealed bid auctions of heterogeneous objects with fair prices. *Social choice and Welfare* 20, 3 (2003), 371–385.

Jack Robertson and William Webb. 1998. *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press.

David Schmeidler and Menahem Yaari. 1971. Fair allocations. *Unpublished Manuscript* (1971).

Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.

Warut Suksompong and Nicholas Teh. 2022. On maximum weighted Nash welfare for binary valuations. *Mathematical Social Sciences* 117 (2022), 101–108.

Warut Suksompong and Nicholas Teh. 2023. Weighted fair division with matroid-rank valuations: Monotonicity and strategyproofness. *Mathematical Social Sciences* 126 (2023), 48–59.

William Vickrey. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* 16, 1 (1961), 8–37.

Vignesh Viswanathan and Yair Zick. 2023a. A general framework for fair allocation under matroid rank valuations. In *Proceedings of the 24th ACM Conference on Economics and Computation*. 1129–1152.

Vignesh Viswanathan and Yair Zick. 2023b. Yankee Swap: A Fast and Simple Fair Allocation Mechanism for Matroid Rank Valuations. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 179–187.

Eric W Weisstein. 2008. Floyd-warshall algorithm. *https://mathworld. wolfram. com/* (2008).

Simon Wimmer and Peter Lammich. 2017. The floyd-warshall algorithm for shortest paths. *Arch. Formal Proofs* 2017 (2017).

Xiaowei Wu, Cong Zhang, and Shengwei Zhou. 2023. One Quarter Each (on Average) Ensures Proportionality. In *International Conference on Web and Internet Economics*. 582–599.

Xiaowei Wu and Shengwei Zhou. 2024. Tree Splitting Based Rounding Scheme for Weighted Proportional Allocations with Subsidy. *arXiv preprint arXiv:2404.07707* (2024).

Dao-Zhi Zeng. 2000. Approximate envy-free procedures. *Game Practice: Contributions from Applied Game Theory* (2000), 259–271.

APPENDIX

A Tightness of the Subsidy Bounds

A.1 Subsidy Bound of Algorithm 1

As Theorem 5.9 implies, Algorithm 1 computes a WEF-able allocation with a total subsidy of at most $(W - w_{\min})V$. However, this bound is not tight. To understand why, consider the case of 2 items, each valued at V by agent $i \in \{1, \dots, n-1\}$, who has an entitlement of $w_i \geq 2$, and $V - \epsilon$ by all other agents. Our algorithm will allocate all the items to agent i, resulting in a subsidy of $\frac{w_j}{w_i}2(V-\epsilon)$ by each other agent $j \neq i \in N$, leading to a total subsidy of $(W - w_i)\frac{2(V-\epsilon)}{w_i}$, for arbitrarily small $\epsilon > 0$.

In general, a WEF-able allocation can achieve a lower subsidy by allocating one item to another agent with higher index $j > i$, i.e, $w_j \geq w_i$. For instance, if one item is allocated to such agent j, agent i envies agent j by $\frac{V}{w_j} - \frac{V}{w_i} \leq 0$, and agent j envies agent i by $\frac{V-\epsilon}{w_i} - \frac{V-\epsilon}{w_j} < \frac{2(V-\epsilon)}{w_i}$. If $\frac{V-\epsilon}{w_i} - \frac{V-\epsilon}{w_j} \leq 0$, then no subsidy is required. Otherwise, the subsidy required by agent j is $\left(\frac{V-\epsilon}{w_i} - \frac{V-\epsilon}{w_j}\right)w_j < \frac{w_j}{w_i} \cdot 2(V-\epsilon)$. The subsidy required by each other agent $k \neq i, j$ is significantly lower than $w_k \cdot \frac{w_j}{w_j} \cdot \frac{2(V-\epsilon)}{w_i} = \frac{w_k}{w_i} \cdot 2(V - \epsilon)$. Therefore, the required total subsidy is significantly lower than $(W - w_i)\frac{2(V-\epsilon)}{w_i}$.

In both cases, the resulting total subsidy bound is better than the bound obtained by allocating all items to agent i.

A.2 Subsidy Bound of Algorithm 2

Example A.1. Consider 2 agents with weights $w_1 = 1$ and $w_2 = 2$, and 2 items o_1 and o_2, where $v(o_1) = \frac{V}{2}$ and $v(o_2) = V$. The allocation X output by Algorithm 2 is $X = (\emptyset, \{o_1, o_2\})$ with total subsidy $\frac{3}{4}V$. On the other hand, total subsidy needed for allocation $(\{o_1\}, \{o_2\})$ is zero. Hence, Algorithm 2 does not necessarily output an allocation with minimum total subsidy and the upper bound on the total subsidy given in Theorem 6.6 is not always optimal.

A.3 Subsidy Bound of Algorithm 3

As Theorem 7.12 implies, Algorithm 3 computes a WEF-able allocation with a total subsidy of at most $\frac{W}{w_{\min}} - 1$. However, with more careful analysis, we can prove a tighter bound.

There are two cases to consider:

1. **Agent 1 with the minimum entitlement receives a positive subsidy.** Together, Proposition 7.7 and Proposition 7.11 imply that $p_i \leq \frac{w_i}{w_{\min}}$ for each agent $i \in N$. Since agent 1 does receive a positive subsidy, and by Theorem 3.5, there exists at least one agent who requires no subsidy, the total required subsidy is bounded by $\frac{W-w_2}{w_{\min}}$.

2. **Agent 1 with the minimum entitlement receives no subsidy.** We can modify Proposition 7.11 in the following way: for each agent $i \notin R(t)$, where $t \in [T]$, $\ell_i(X^t) \leq \frac{1}{w_2}$. By the proof of Proposition 7.11, $\ell_i(X^t) \leq \ell_{i_{j+1}}(X^t)$. If $i_{j+1} = i_1$, then $\ell_i(X^t) \leq \ell_{i_{j+1}}(X^t) \leq 0$ (because agent 1 requires no subsidy). Otherwise, $\ell_i(X^t) \leq \ell_{i_{j+1}}(X^t) \leq \frac{1}{w_{i_{j+1}}} \leq \frac{1}{w_2}$. Overall, the subsidy required by each agent is bounded by $\frac{w_i}{w_2}$, and by Theorem 3.5, there exists at least one agent who requires no subsidy. Therefore, the total required subsidy is bounded by $\frac{W-w_{\min}}{w_2}$.

To sum up, the total required subsidy is at most $\max\left\{\frac{W-w_{\min}}{w_2}, \frac{W-w_2}{w_{\min}}\right\}$.

A.4 Subsidy Bound of Algorithm 4

The following example shows that Algorithm 4 does not necessarily output an allocation with minimum total subsidy:

Example A.2. Consider 3 agents with weights $w_1 = w_2 = w_3 = 1$ and 4 identical items, where $v_1 > v_2 > v_3$. The allocation X output by Algorithm 4 is $X = (2, 1, 1)$ with total subsidy $2v_2$. On the other hand, total subsidy needed for allocation $(2, 2, 0)$ is $2v_1$, which is smaller than $2v_2$.

This raises an important question that remains open: with identical items and additive valuations, is it possible to find a WEF allocation with minimum subsidy in polynomial time?