Computing Envy-Free up to Any Good (EFX) Allocations via Local Search

Simina Brânzei*

October 8, 2025

Abstract

We present a simple local search algorithm for computing EFX (envy-free up to any good) allocations of m indivisible goods among n agents with additive valuations. EFX is a compelling fairness notion, and whether such allocations always exist remains a major open question in fair division.

Our algorithm employs simulated annealing with the total number of EFX violations as an objective function together with a single-transfer neighborhood structure to move through the space of allocations. It found an EFX allocation in all the instances tested, which included thousands of randomly generated inputs, and scaled to settings with hundreds of agents and/or thousands of items. The algorithm's simplicity, along with its strong empirical performance makes it a simple benchmark for evaluating future approaches.

On the theoretical side, we provide a potential function for identical additive valuations, which ensures that any strict-descent procedure under the single-transfer neighborhood ends at an EFX allocation. This represents an alternative proof of existence for identical valuations.

Keywords. fair division, indivisible goods, envy-freeness-up-to-any-good (EFX), local search, simulated annealing.

1 Introduction.

We study the fair division of a set of indivisible goods among agents with additive valuations. Formally, let $M = \{1, \ldots, m\}$ denote the set of goods and $N = \{1, \ldots, n\}$ the set of agents. Each agent $i \in N$ has a nonnegative value $v_{i,j}$ for each good $j \in M$, and their utility for a bundle $S \subseteq M$ is additive: $u_i(S) = \sum_{j \in S} v_{i,j}$. An *allocation* is a partition of M into disjoint bundles $A_1, \ldots, A_n$ assigned to the agents.

A fundamental goal is to ensure that the allocation is fair. One of the best known fairness notions is *envy-freeness (EF)*, which requires that no agent prefers someone else's bundle: $u_i(A_i) \geq u_i(A_j)$ for all $i, j \in N$. However, envy-free allocations are not guaranteed to exist for indivisible goods. For instance, if there is a single indivisible item to be allocated (e.g. a necklace), an agent who does not get the item will envy the one who does.

Since envy-freeness is often unattainable, alternative fairness notions have been proposed that adapt more naturally to discrete settings. A widely studied notion is *envy-freeness up to one good (EF1)*,

*Purdue University. Work done in part at Google. This research was supported in part by US National Science Foundation grant CCF-2238372. E-mail: simina.branzei@gmail.com. The implementation of the algorithm and the scripts with data used to generate the images are available at https://github.com/domnica/EFX.

which requires that envy can be eliminated by removing some item from the envied bundle. A stronger notion is *envy-freeness up to any good (EFX)* [CKM+19], which demands that for every pair of agents i and j, and for every good $g \in A_j$, agent i does not envy agent j once g is removed: $u_i(A_i) \geq u_i(A_j \setminus \{g\})$.

More generally, an allocation is said to be $(1-\epsilon)$-*EFX*, for some $\epsilon \in [0, 1)$ if for all $i, j \in N$ and all $g \in A_j$, it holds that $u_i(A_i) \geq (1-\epsilon) \cdot u_i(A_j \setminus \{g\})$. For general sub-additive valuations, it is known that (1/2)-EFX allocations exist [PR20], and better approximations are possible for restricted valuation classes.

The existence of *exact* EFX allocations[1] has been conjectured. Proving or disproving this conjecture represents a major open problem in fair division. Existence of exact EFX allocations for additive valuations has been proved for two agents [PR20], three agents [CGM20], and certain special classes of valuations (see, e.g., [AFRV20, HGNV25]).

1.1 Our Contribution.

Our main contribution is to design a simple local search algorithm for computing EFX allocations on instances with additive valuations. The algorithm employs simulated annealing, using the number of EFX violations as an objective function, and explores the space of allocations through a single-item transfer neighborhood structure.

Our experiments show that the algorithm finds EFX allocations consistently and efficiently across a wide range of instance sizes, with uniform and correlated valuations. In fact, the algorithm converged to an EFX allocation on every instance we tested. We also observe that the runtime is affected by the item-to-agent ratio m/n and by how correlated the agents' valuations are. Implementation details and reproducibility resources are described in the experimental results section.

On the theoretical side, we provide a potential function for identical additive valuations, which is defined for each allocation A as the variance across agents of their total bundle values. We show that if an allocation A is not EFX, then some single-item transfer strictly decreases Φ. Consequently, any strict-descent process on Φ with the single-transfer neighborhood structure ends at an EFX allocation, giving an alternative existence proof for identical additive valuations.

1.2 Related Work.

The study of fair allocation of indivisible goods has seen significant growth over the last decade, particularly concerning relaxations of envy-freeness such as EF1 and EFX. For a comprehensive survey, see [AAB+23].

EFX existence. Envy-freeness up to any good (EFX) has become a central open problem in fair division due to its intuitive appeal. Despite significant attention, the general existence of EFX allocations remains unresolved. Prior work has tackled this challenge by focusing on specific scenarios or relaxed versions of EFX. For example, [PR20] initially established EFX existence for two agents and for any number of agents with identical (not necessarily additive) valuations. Extending beyond two agents, [CGM20] confirmed the existence of EFX allocations for three agents with additive valuations and showed they can be computed in pseudo-polynomial time. [AFRV20] showed that EFX allocations exist for bi-valued instances. [HGNV25] showed that EFX allocations exist for n agents whose valuations fall in one of three types.

[1] That is, $(1-\epsilon)$-EFX with $\epsilon = 0$.

Work by [AAC+25] advanced this line of research by proving the existence of EFX allocations in scenarios where two agents possess general monotone valuations, and at least one agent has an MMS-feasible valuation. Their approach simplifies prior methodologies by eliminating the need for complex constructions such as envy or champion graphs.

Approximate EFX. On another front, approximate EFX allocations with limited unallocated goods (charity) have been extensively explored. Initial work by [DGK+14] introduced EFX with charity and showed that meaningful approximations are possible. Later, [CGM21] leveraged combinatorial arguments based on the "rainbow cycle number" (RCN) to achieve $(1-\varepsilon)$-EFX allocations with bounded charity. [BCF+22] show that every 4-agent instance with additive valuations has an EFX allocation that leaves at most a single item unallocated, and every setting with n additive valuations has an EFX allocation with at most $n-2$ unallocated items. [BCF+22] extend these results to nice cancelable valuations (which include additive, unit-demand, budget-additive and multiplicative valuations). [GHL+23] analyze submodular valuations and design a polynomial time algorithm based on local search to find an allocation that is simultaneously 1/2-EFX and approximates the optimal Nash social welfare within a factor of $(8+\epsilon)$ for every $\epsilon > 0$.

EF1 and proportionality. We briefly mention several other prominent fairness notions for indivisible goods. Envy-freeness up to one good (EF1) is guaranteed to exist for arbitrary monotone valuations [LMMS04] and has been extensively studied due to its practical applicability and computational tractability. Reachability of EF1 allocations via sequential exchanges was studied in [IKSY24], where the question is whether one EF1 allocation can be reached from another EF1 allocation via a sequence of exchanges such that every intermediate allocation is EF1. EF1 allocations for mixed (divisible and indivisible) goods has been studied in [BLL+21].

Proportionality requires that each agent has utility at least $1/n$ of the utility they would have if they received the entire set of items. While proportionality cannot always be ensured for indivisible goods, relaxations of it can. There exist various proportionality relaxations such as Maximin share (MMS), Proportionality up to one good (PROP1), and Proportionality up to any good (PROPx). Each of these has different guarantees and computational complexities. [CSHS24] study weighted versions of these fairness notions.

Competitive equilibrium, share-based notions, and stochastic settings. The competitive equilibrium from equal incomes has also been considered as a method for allocating indivisible goods (see, e.g., [Bud11, BL14, OPR14, BHM15]), with extensions to competitive equilibria from unequal budgets [BNTC21] or random incomes [NTV25]. Share based notions of fairness for indivisible goods have also been proposed (see, e.g., [BNTC21, BF22, BFHN24]).

The existence of fair allocations in stochastic settings (e.g. with random valuations) has also been studied (see, e.g., [DGK+14, MS20, BHPV24]).

Local search in other domains. Stochastic local search methods have also been extensively studied in domains outside fair division, such as Boolean satisfiability. For example, the WalkSAT algorithm for satisfiability [Pap91, Sch99] selects an unsatisfied clause uniformly at random and flips the value of a random variable from that clause. In [SKC+93] the algorithm additionally performs a random walk over variables that appear in unsatisfied clauses to escape local minima.

2 Definitions.

In order to introduce the algorithm, we first define the notion of EFX violation.

Definition 1 (EFX violation). *Let $A = (A_1, \ldots, A_n)$ be an allocation. A tuple (i, j, g), where $i, j \in N$ and $g \in M$, represents an EFX violation if $u_i(A_j \setminus \{g\}) > u_i(A_i)$.*

Our measure of progress is the total number of EFX violations.

Definition 2 (EFX violation count). *Given an allocation $A = (A_1, \ldots, A_n)$, the EFX violation count at A is:*

$$f(A) = \sum_{i \in N} \sum_{j \in N \setminus \{i\}} \sum_{g \in A_j} c(i, j, g),$$

where $c(i, j, g) = \mathbb{1}\{u_i(A_j \setminus \{g\}) > u_i(A_i)\}$.

Clearly, an allocation A is EFX if and only if $f(A) = 0$.

We also define a neighborhood structure in the space of allocations.

Definition 3 (Single-transfer neighborhood). *Let $A = (A_1, \ldots, A_n)$ be an allocation. An allocation $A' = (A'_1, \ldots, A'_n)$ is a* neighbor *of A if it differs from A by the transfer of a single item from its current owner in A to another agent.*

3 Algorithm.

Our algorithm explores the space of allocations via local moves, starting from an initial allocation A drawn uniformly at random.

At each step, the algorithm generates a random neighbor A' of the current allocation A by selecting an item j uniformly at random and reassigning it from its owner i in A to another agent chosen uniformly at random from $N \setminus \{i\}$. It then compares the number of EFX violations at A and A':

- If $f(A') < f(A)$, the algorithm moves to A'.
- Otherwise, it moves to A' with a small probability $p = \exp(-\Delta/T)$, where $\Delta = f(A') - f(A)$ and T is a temperature parameter that decreases over time.

This process repeats until an allocation with zero violations is found.

The non-zero transition probability to allocations with higher f values is necessary: there exist instances where the function f has strictly positive local minima.

Pseudocode. Below, we present the pseudocode for our algorithm. Parameter values used in the implementation (such as the initial temperature T) are shown in brackets when introducing each parameter.

```
Input:
    n - number of agents
    m - number of items
    v - valuation matrix, where v[i][j] is agent i's value for item j
Output:
    An EFX allocation (if one is found)
```

```
1. Repeat until an EFX allocation is found:

    a. Initialize a random allocation A, by assigning each item to a uniformly
    random agent.

    b. Set the temperature T to an initial value (e.g., 5.0) and define a minimum
    threshold (e.g., Tmin = 0.0001).

    c. While T > Tmin and the allocation is not yet EFX:

        i. For a number of steps at the current temperature (e.g., 100 × n × m):

            - Count the EFX violations at the current allocation (i.e., f(A)).

            - Propose a new allocation A' by selecting an item uniformly at random,
            removing it from its current owner in A, and reassigning it to a
            different agent (chosen uniformly at random).

            - Let D be the change in the number of EFX violations if the item is
            reassigned (i.e., D = f(A') - f(A))

            - If D < 0 (i.e., fewer violations) accept the move  (i.e., update the
            allocation to A').

            - Else, accept the move with probability exp(-D / T).

        ii. Decrease the temperature  (e.g., T ← 0.99 × T).

2. Return the EFX allocation once one is found (i.e., the number of violations
is zero).
```

4 Experimental Results.

We begin with uniform random instances, where each entry $v_{i,j}$ of the valuation matrix is drawn independently and uniformly from the interval $[0, 1]$.

The key goals are to evaluate whether the algorithm can find EFX allocations and measure the runtime required for convergence as a function of the number of agents and items. To the best of our knowledge, no efficient, publicly available algorithm exists for computing EFX allocations in these settings. For context, the search space for $n = 4$ and $m = 1000$ is already computationally prohibitive for exhaustive search methods.

Experimental environment. Experiments were conducted within a KVM virtual machine running Debian GNU/Linux (kernel 6.12.27). The virtual machine was provisioned with 24 threads from an AMD EPYC 7B12 host processor and 94 GB of RAM. The code was written in Python 3.13.5 and is available at [Bra25].

4.1 Scaling with the Number of Items (Fixed Agents)

In this section we check the performance of the algorithm for various combinations of agents (n) and items (m).

Table 1 summarizes these initial results, presenting the average runtime (left) and the corresponding number of algorithmic steps (right) for several configurations. The "number of steps" refers to the total count of neighboring allocations generated before an EFX allocation is found.

An observation from Table 1 is that the runtime is not monotonic with respect to the number of items. For example, when $n = 15$, the algorithm is substantially slower for $m = 150$ items than for $m = 750$ items. This finding suggests that the performance of the algorithm depends more critically on the item-to-agent ratio (m/n) than on the absolute number of items. To investigate this phenomenon more closely, the next section provides a detailed analysis focused on the m/n ratio for a fixed number of agents.

Agents (n)	Items			
	1000	$10n$	$50n$	$5n^2$
4	0.41 ± 0.24	0.004 ± 0.003	0.02 ± 0.01	0.007 ± 0.004
10	1.46 ± 0.53	0.34 ± 0.26	0.42 ± 0.13	0.42 ± 0.13
15	2.22 ± 0.63	15.24 ± 18.72	1.34 ± 0.43	2.75 ± 0.94
20	2.73 ± 0.67	599.70 ± 405.04	2.73 ± 0.67	8.63 ± 2.37

(a) Average runtime (in seconds).

Agents (n)	Items			
	1000	$10n$	$50n$	$5n^2$
4	797 ± 472	135 ± 79	232 ± 136	135 ± 71
10	1902 ± 697	3485 ± 2737	1169 ± 375	1169 ± 375
15	2544 ± 714	99186 ± 121372	2177 ± 694	2802 ± 948
20	3181 ± 780	2941904 ± 1990182	3181 ± 780	4783 ± 1324

(b) Average number of steps.

Table 1: Performance metrics for finding EFX allocations across various problem sizes. Each entry represents the average over 100 instances with agent valuations drawn uniformly at random. The value below each mean indicates one standard deviation.

4.2 Effect of the Item-to-Agent Ratio

To investigate the effect of item availability, we conducted an experiment with a fixed number of agents ($n = 15$) and varied the number of items from $m = 3$ to $m = 1050$. This design allowed us to observe the algorithm's performance as it transitioned from a state of item scarcity ($m < n$) to one of item abundance ($m \gg n$). The evolution of the average runtime as the number of items grows is plotted in Figure 1. The corresponding plot for the step count is in the appendix (Figure 5).

The primary finding is that the problem is hardest for the algorithm not when the number of items is largest, but within a volatile region at low item-to-agent ratios. As shown in Figure 1, the runtime does not exhibit a single sharp peak; instead, it oscillates with the maximum trending upward, reaching its global maximum near $m/n \approx 3.5$.

Beyond this hard region, as the number of items increases further, the problem becomes easier, and the runtime dramatically decreases to a minimum around $m/n \approx 25$. Subsequently, the runtime begins to increase again, but very slowly. This non-monotonic behavior strongly suggests that the hardness is driven by the problem's combinatorial structure rather than merely the size of the input.

Prior research from [DGK+14] has shown that for random valuations, an exact envy-free allocation exists with high probability when $m \in \Omega(n \log n)$ and is unlikely to exist when $m \in n + o(n)$.

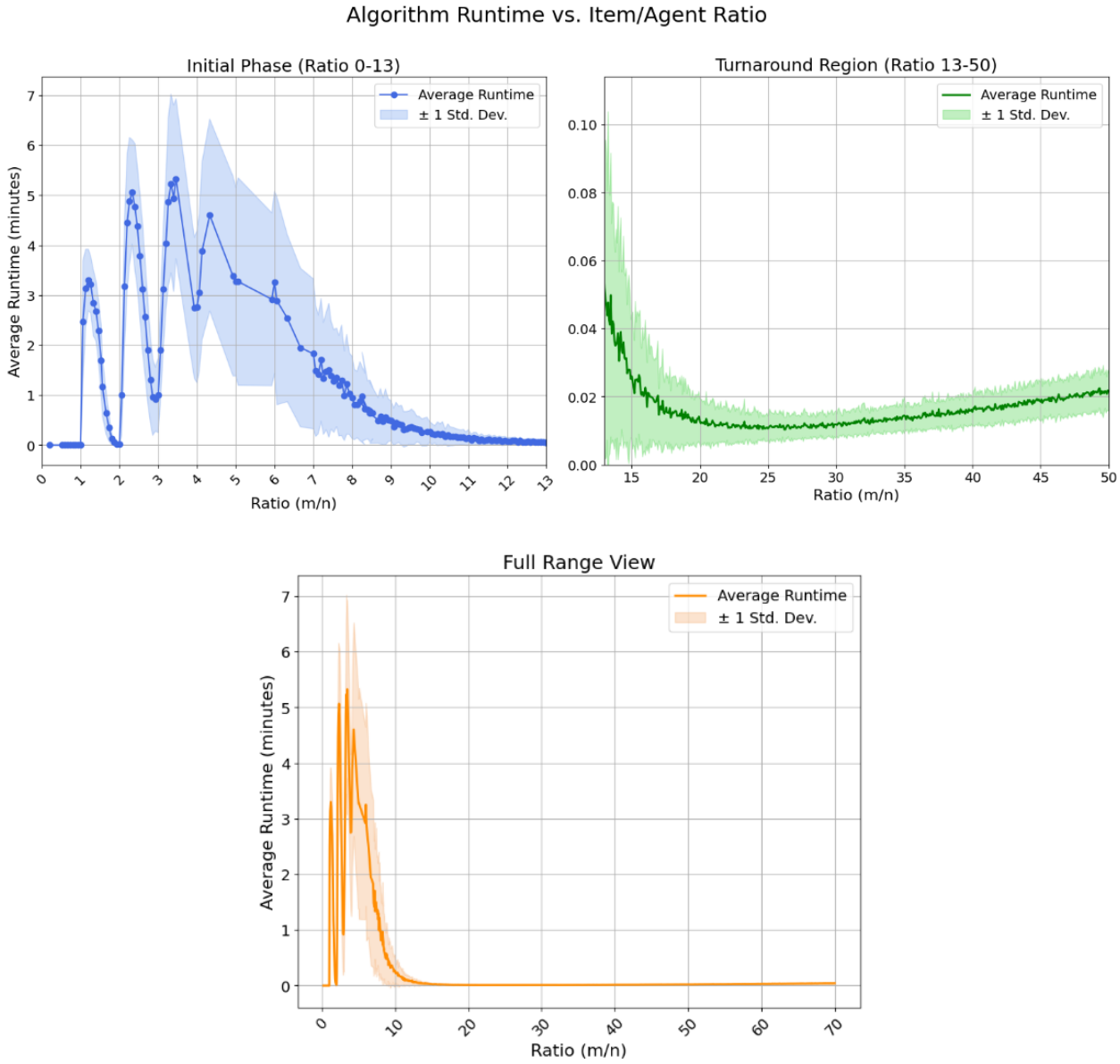

Figure 1: Average runtime (in minutes) vs. the item-to-agent ratio (m/n) for finding an EFX allocation with $n = 15$ agents. Each data point shows the average runtime over 100 independent instances where agent valuations were drawn uniformly at random. The shaded region indicates one standard deviation. The figure presents three panels: a close-up of the region $m/n \in [0, 13]$, a second close-up of the region $m/n \in [13, 50]$, and a full-range view.

Follow-up work by [MS20] improved the existence bound from [DGK+14], showing that envy-free allocations exist with high probability when $m \geq 2n$ and n divides m.

Understanding whether this computational barrier stems from the structure of the problem or the nature of the algorithm could shed light on the broader complexity of computing EFX allocations. For example, when $m = n + 1$, the problem can in fact be solved efficiently using a round-robin approach: let the agents, in the order $1, 2, \ldots, n-1, n, n$, choose their favorite items among the remaining ones. The resulting allocation is EFX.

4.3 Scaling with the Number of Agents (Fixed Items)

We next evaluate the algorithm's scalability with respect to the number of agents, n. For this experiment, we fixed the number of items to a large value ($m = 10^4$) and increased the number of agents from $n = 4$ to $n = 100$. The results are shown in Figure 2.

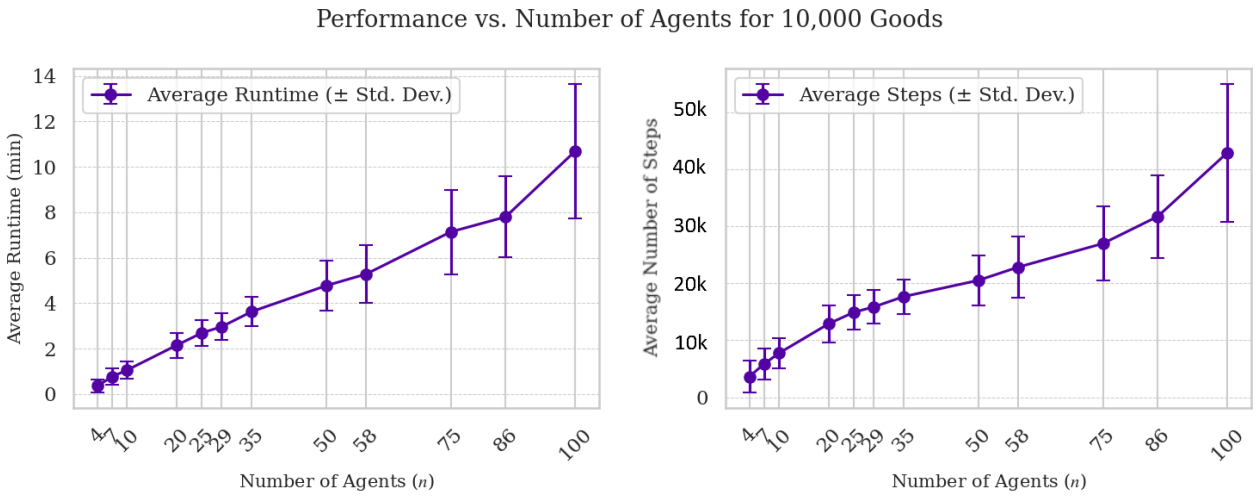

Figure 2: Performance of the algorithm with $m = 10,000$ items as the number of agents grows from $n = 4$ to $n = 100$. For each value of n displayed, the data point is obtained by taking the average of 100 runs with uniformly random valuations. The left panel shows the average runtime and standard deviation (in minutes), while the right panel shows the average and standard deviation in steps.

4.4 Correlated Valuations

We also test correlated valuations using a simple "shared value + individual term" model parametrized by a correlation strength $\rho \in [0, 1]$.

For each item $j \in [m]$ we draw a common value $w_j \sim \text{Unif}[0, 1]$. Independently, for each agent $i \in [n]$ we also draw an individual term $u_{i,j} \sim \text{Unif}[0, 1]$. The value of agent i for item j is then defined as

$$v_{i,j} \;=\; \rho\, w_j + (1 - \rho)\, u_{i,j}\,.$$

This convex combination keeps $v_{i,j} \in [0, 1]$, gives i.i.d. uniform valuations when $\rho = 0$, and yields identical valuations across agents when $\rho = 1$. Intermediate ρ smoothly interpolates between heterogeneous and identical preferences.

Empirical behavior. In our experiments, the effect of the correlation strength ρ depends on the item-to-agent ratio m/n.

We first test an item-abundant regime with $n = 8$ and $m = 160$, corresponding to a ratio $m/n = 20$. This is a setting where instances with uniform valuations are computationally straightforward. For correlated valuations, the mean runtime increases with the correlation strength ρ, peaking sharply near $\rho = 0.95$ due to a heavy-tailed runtime distribution, before dropping again for identical valuations ($\rho = 1$). See Figure 3 for runtimes, with full numeric values in the appendix (Table 2 and 3).

For our second case, we select parameters $n = 15$ and $m = 52$. This creates an item-scarce regime ($m/n \approx 3.46$) specifically chosen because this exact setting represents one of the most computationally challenging peaks for uniform valuations (as shown in Figure 1). Here the trend is reversed for small ρ, with the hardest regime occurring at $\rho = 0$. See Figure 4 for runtimes, with full numeric values in the appendix (Table 4 and 5).

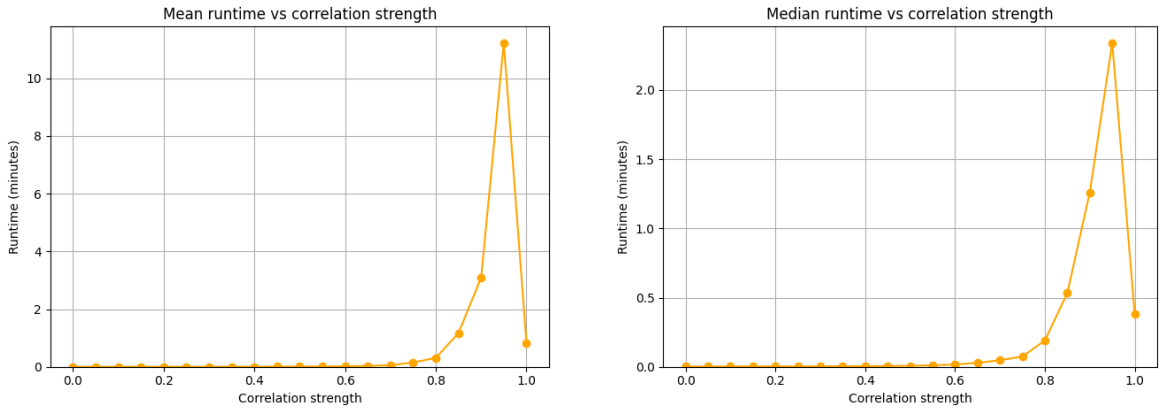

Figure 3: Runtime (in minutes) for finding an EFX allocation with $n = 8$ agents and $m = 160$ items as the correlation strength grows from 0 to 1. For each correlation strength, the results are averaged over 100 trials. The left figure shows the mean and the right figure shows the median.

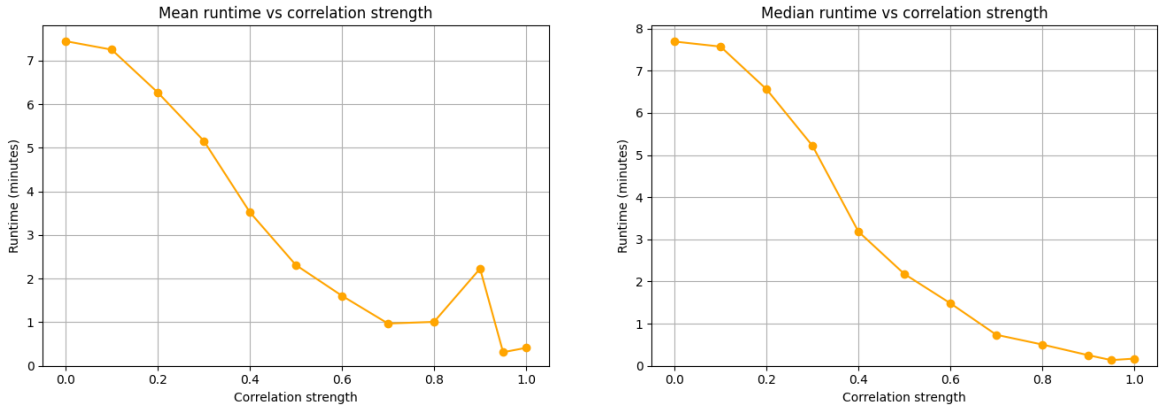

Figure 4: Runtime (in minutes) for finding an EFX allocation with $n = 15$ agents and $m = 52$ items as the correlation strength grows from 0 to 1. For each correlation strength, the results are averaged over 100 trials. The left figure shows the mean and the right figure shows the median.

5 Potential Function for Identical Valuations

When valuations are identical, each good $g \in M$ has a common value w_g; that is $v_{i,g} = w_g \ \forall i \in N$.

We find a potential function Φ that assigns a value to each allocation A and has the property that, if A is not EFX, there exists a single-item transfer that can be made to strictly decrease Φ. Consequently, any strict-descent procedure on Φ with the single-transfer neighborhood terminates at an EFX allocation. While the existence of EFX allocations is known for identical additive valuations from [PR20], to the best of our knowledge this proof is new.

Proposition 1. *Consider an instance with identical valuations, where each good $g \in M$ has a common value $w_g > 0$. For an allocation $A = (A_1, \ldots, A_n)$, define*

$$\Phi(A) \ := \ \sum_{i=1}^{n} (Y_i - \mu)^2 \,, \quad \textit{where } Y_i := \sum_{g \in A_i} w_g \ \ \textit{and} \ \ \mu := \frac{1}{n} \sum_{g \in M} w_g \,. \tag{1}$$

Then every strict descent procedure on Φ using the single-good transfer neighborhood from Definition 3 is guaranteed to terminate at an EFX allocation.

Proof. An allocation A with identical valuations is EFX if and only if for each pair of agents $i \neq j$ and all $g \in A_j$, we have $u_i(A_i) \geq u_i(A_j \setminus \{g\})$. Since the goods have common value, the EFX condition is equivalent to

$$\sum_{h \in A_i} w_h \geq \big(\sum_{h \in A_j} w_h \big) - w_g \,. \tag{2}$$

Using the definition of Y_k from (1), inequality (2) can be rewritten as $Y_j - Y_i \leq w_g$.

Consider an allocation A that is not EFX. Then there exists a triple (i, j, g) with $i, j \in N$ and $g \in A_j$ such that $Y_j - Y_i > w_g$. Let A' be the new allocation obtained from A by transferring good g from agent j to agent i. The change in the objective Φ is:

$$\Delta\Phi = \Phi(A') - \Phi(A) = 2w_g(w_g - (Y_j - Y_i)).$$

Since $w_g > 0$ and we assumed $Y_j - Y_i > w_g$, it follows that $w_g - (Y_j - Y_i) < 0$. Thus $\Delta\Phi < 0$, and so $\Phi(A') < \Phi(A)$.

This shows that the existence of any EFX violation at allocation A guarantees that a strictly improving move (i.e. single-item transfer) is available from A. A strict descent algorithm on a finite state space is guaranteed to terminate in a local minimum. Since non-EFX allocations cannot be local minima, the algorithm must terminate at an EFX allocation. □

6 Discussion.

EFX existence. Our algorithm found an EFX allocation in all instances tested. This suggests not only that EFX allocations are likely to exist for additive valuations on the distributions tested, but that they are also computationally accessible via simple local search dynamics. It would be interesting to better understand the computational hardness of the EFX problem as a function of the ratio m/n and the degree to which the valuations are correlated.

Accelerating convergence. One could investigate the effect of warm starts on the algorithm's performance. A natural choice for initialization is the welfare-maximizing allocation, where the social welfare of an allocation A is defined as $SW(A) = \sum_{i \in N} u_i(A_i)$. Prior work [DGK+14] shows that when the number of goods satisfies $m \in \Omega(n \log n)$, the welfare-maximizing allocation is envy-free (and thus EFX) with high probability.

We observed that when items are abundant, initializing the algorithm from a welfare-maximizing allocation leads to faster convergence than starting from a random one, even when the initial allocation is not EFX; the algorithm corrects the remaining violations quickly. While we do not report detailed experiments on this aspect, it suggests that high-welfare initializations may serve as effective warm starts when the number of items is large.

Other potential directions include exploring different neighborhood structures (e.g., 2-local moves), experimenting with alternative objective functions and annealing parameters (such as temperature decay schedules), and adding a regularization term to the objective.

Comparison with round robin. We also compared to the round robin algorithm, which is a sequential method where agents take turns making a greedy choice—selecting their most valued item from the available pool—while following a fixed, cyclical order (e.g., $1, \ldots, n$) until all items are allocated. Round robin fails to produce an EFX allocation in many instances, especially for correlated valuations.

7 Acknowledgements

The author would like to thank Ruta Mehta for helpful feedback. The author also acknowledges using large language models (ChatGPT and Gemini) while brainstorming to explore candidate objective/potential functions.

References

[AAB+23] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.*, 322:103965, 2023.

[AAC+25] Hannaneh Akrami, Noga Alon, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. Efx: A simpler approach and an (almost) optimal guarantee via rainbow cycle number. *Operations Research*, 73(2):738–751, 2025.

[AFRV20] Georgios Amanatidis, Aris Filos-Ratsikas, and Alexandros A. Voudouris. On the pareto frontier of envy-freeness and nash social welfare. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 35–41, 2020.

[BCF+22] Niv Berger, Ariel Cohen, Michal Feldman, Amos Fiat, and Gleb Polevoy. Almost envy-freeness with arbitrary valuations. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 548–567. SIAM, 2022.

[BF22] Moshe Babaioff and Uriel Feige. Fair shares: Feasibility, domination and incentives. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 435, New York, NY, USA, 2022. Association for Computing Machinery.

[BFHN24] Yakov Babichenko, Michal Feldman, Ron Holzman, and Vishnu V. Narayan. Fair division via quantile shares. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1235–1246, New York, NY, USA, 2024. Association for Computing Machinery.

[BHM15] Simina Brânzei, Hadi Hosseini, and Peter Bro Miltersen. Characterization and computation of equilibria for indivisible goods. In *International Symposium on Algorithmic Game Theory*, pages 244–255. Springer, 2015.

[BHPV24] Gerdus Benade, Daniel Halpern, Alexandros Psomas, and Paritosh Verma. On the existence of envy-free allocations beyond additive valuations. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 1287, New York, NY, USA, 2024. Association for Computing Machinery.

[BL14] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, page 1321–1328, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[BLL+21] Xiaohui Bei, Zihao Li, Jinyan Liu, Shengxin Liu, and Xinhang Lu. Fair division of mixed divisible and indivisible goods. *Artificial Intelligence*, 293:103436, 2021.

[BNTC21] Moshe Babaioff, Noam Nisan, and Inbal Talgam-Cohen. Competitive equilibrium with indivisible goods and generic budgets. *Mathematics of Operations Research*, 46(1):382–403, 2021.

[Bra25] Simina Branzei. Jupyter notebook with implementation of the algorithm, 2025. https://github.com/domnica/EFX.

[Bud11] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[CGM20] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. Efx exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 1–19, 2020.

[CGM21] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta. A little charity guarantees almost envy-freeness. In *Proceedings of the 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1044–1055. IEEE, 2021.

[CKM+19] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, pages 305–322. ACM, 2019.

[CSHS24] Mithun Chakraborty, Erel Segal-Halevi, and Warut Suksompong. Weighted fairness notions for indivisible items revisited. *ACM Trans. Econ. Comput.*, 12(3), September 2024.

[DGK+14] John P. Dickerson, Jonathan Goldman, Jeremy Karp, Ariel D. Procaccia, and Tuomas Sandholm. The computational rise and fall of fairness. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1405–1411. AAAI Press, 2014.

[GHL+23] Jugal Garg, Edin Husić, Wenzheng Li, László A. Végh, and Jan Vondrák. Approximating nash social welfare by matching and local search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1298–1310, New York, NY, USA, 2023. Association for Computing Machinery.

[HGNV25] Vishwa Prakash HV, Pratik Ghosal, Prajakta Nimbhorkar, and Nithin Varma. EFX exists for three types of agents. In Itai Ashlagi and Aaron Roth, editors, *Proceedings of the 26th ACM Conference on Economics and Computation, EC 2025, Stanford University, Stanford, CA, USA, July 7-10, 2025*, pages 101–128. ACM, 2025.

[IKSY24] Ayumi Igarashi, Naoyuki Kamiyama, Warut Suksompong, and Sheung Man Yuen. Reachability of fair allocations via sequential exchanges. *Algorithmica*, 86(12):3653–3683, September 2024.

[LMMS04] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[MS20] Pasin Manurangsi and Warut Suksompong. When do envy-free allocations exist? *SIAM Journal on Discrete Mathematics*, 34(3):1505–1521, 2020.

[NTV25] Thành Nguyen, Alexander Teytelboym, and Shai Vardi. Efficiency, envy and incentives in combinatorial assignment. In *Proceedings of the 26th ACM Conference on Economics and Computation*, EC '25, page 476, New York, NY, USA, 2025. Association for Computing Machinery.

[OPR14] Abraham Othman, Christos Papadimitriou, and Aviad Rubinstein. The complexity of fairness through equilibrium. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, page 209–226, New York, NY, USA, 2014. Association for Computing Machinery.

[Pap91] Christos H Papadimitriou. On selecting a satisfying truth assignment. In *Proceedings of the 32nd Annual Symposium of Foundations of Computer Science*, pages 163–169. IEEE Computer Society, 1991.

[PR20] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2140–2147, 2020.

[Sch99] T Schoning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 410–414. IEEE, 1999.

[SKC+93] Bart Selman, Henry A Kautz, Bram Cohen, et al. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability*, 26:521–532, 1993.

A Appendix

In the next two tables we include the runtimes and number of steps required to find an EFX allocation for $n = 8$ agents and $m = 160$ items with correlated valuations.

	Correlation Strength											
Runtimes (s)	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1.0
Mean ± SD	0.13 ±0.05	0.16 ±0.07	0.19 ±0.07	0.24 ±0.11	0.34 ±0.19	0.54 ±0.36	1.22 ±0.81	3.60 ±2.99	18.69 ±20.83	185.55 ±274.18	673.91 ± 3457.02	49.88 ± 63.43
Median	0.12	0.14	0.18	0.21	0.29	0.46	1.05	2.89	11.44	75.62	140.41	22.75

Table 2: Runtime (in seconds) for finding an EFX allocation with $n = 8$ agents and $m = 160$ items. The first data row shows the mean ± std. dev.; the second shows the median. For each correlation strength, the results are averaged over 100 trials.

	Correlation Strength											
Steps	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
Mean ± SD	540 ±205	647 ±282	783 ±306	978 ±463	1422 ±784	2270 ±1493	5093 ±3399	15002 ±12464	77906 ±86746	774k ±1.15M	2.81M ± 14.44M	205k ± 263k
Median	483	563	742	889	1198	1871	4352	12221	48082	314186	585780	90008

Table 3: Number of algorithmic steps (neighboring allocations generated) required to find an EFX allocation with $n = 8$ agents and $m = 160$ items. The first data row shows the mean ± std. dev.; the second shows the median. For each correlation strength, the results are averaged over 100 trials. We use 'k' for thousands and 'M' for millions.

In the next two tables we include the runtimes and number of steps required to find an EFX allocation for $n = 15$ agents and $m = 52$ items with correlated valuations.

	Correlation Strength											
Runtimes	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1.0
Mean ± SD	446.83 ±151.79	435.16 ±152.13	376.23 ±160.83	309.64 ±159.77	211.43 ±134.80	138.72 ±95.93	96.46 ±73.91	58.25 ±53.93	60.52 ±78.80	133.79 ±1093.61	18.75 ± 26.64	24.79 ± 38.23
Median	461.51	454.35	393.66	313.25	191.05	130.71	89.08	44.21	30.35	15.16	8.20	10.28

Table 4: Runtime (in seconds) for finding an EFX allocation with $n = 15$ agents and $m = 52$ items. The first data row shows the mean ± std. dev.; the second shows the median. For each correlation strength, the results are averaged over 100 trials.

	Correlation Strength											
Steps	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1.0
Mean ± SD	3.41M ± 1.17M	3.29M ± 1.15M	2.94M ± 1.26M	2.41M ± 1.24M	1.64M ±1.05M	1.07M ±743k	745k ±570k	450k ±417k	464k ±603k	1.02M ±8.38M	144k ± 203k	190k ± 294k
Median	3.56M	3.42M	3.09M	2.45M	1.49M	1.01M	692k	340k	233k	117k	63k	79k

Table 5: Number of algorithmic steps (neighboring allocations generated) required to find an EFX allocation with $n = 15$ agents and $m = 52$ items. The first data row shows the mean ± std. dev.; the second shows the median. For each correlation strength, the results are averaged over 100 trials. We use 'k' for thousands and 'M' for millions.

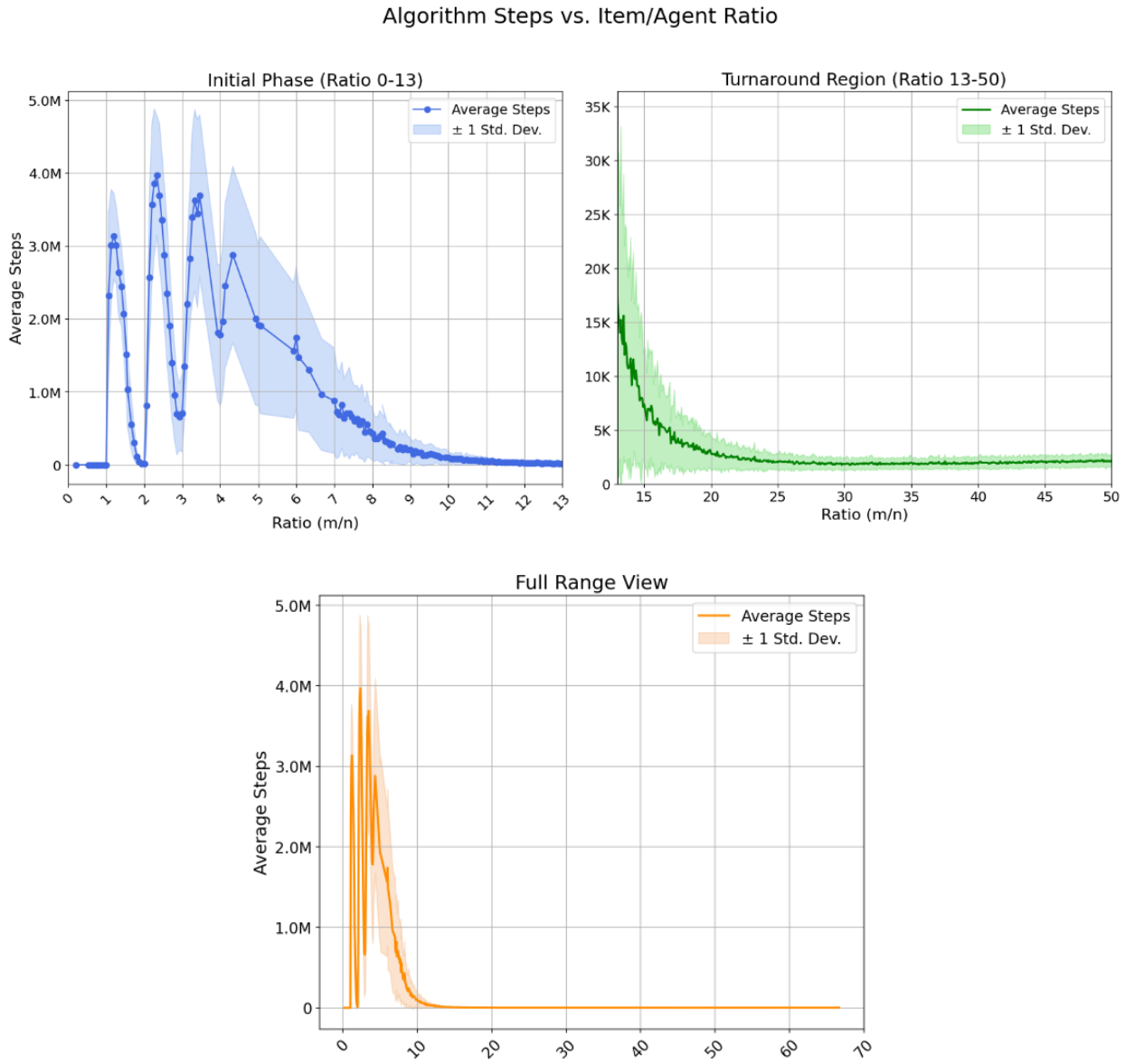

Figure 5: Average number of steps vs. the item-to-agent ratio (m/n) for finding an EFX allocation with $n = 15$ agents. A step counts a neighboring allocation of the current one that was generated by the algorithm. Each data point shows the average number of steps taken by the algorithm over 100 independent instances where agent valuations were drawn uniformly at random. The shaded region indicates one standard deviation. The figure presents three panels: a close-up of the region $m/n \in [0, 13]$, a second close-up of the region $m/n \in [13, 50]$, and a full-range view.