REPORT
Independent Study: MTD-350

Characterising the Existance of EF Allocations for the k-Approval Problem

Facilitator: Prof.Ashutosh Rai

Authors

Rakshitha

2021MT10904

mt1210904@iitd.ac.in

Ayush Gupta

2021MT10697

mt1210697@iitd.ac.in

Contents

1 Structure of the Independent Study

The independent study was divided into two main phases. The first half of the semester concentrated on examining foundational concepts and challenges in Computational Social Choice, laying a theoretical foundation through a detailed literature review. The second half was devoted to refining and working on a specific research problem within the domain.

Core references for the initial exploration include the Handbook of Computational Social Choice and presentations from the Algorithmic Frontiers of Fairness Workshop - FSTTCS '23. For the sake of brevity, this report does not include an in-depth discussion of the literature reviewed during the initial phase.

This document focuses on summarizing the research problem explored during the second phase and details the various approaches attempted to achieve meaningful progress. Despite significant efforts over approximately two months, the progress made has been limited, leaving room for future investigation.

2 Introduction

The Fair Division problem is one among the many problems in Computational Social Choice, with relevance across various fields like Economics, Computer Science, Political Science, Operations Research and many more. The problem, in essence, is to allocate a set of resources among a group of agents in an "envy-free"manner. Envy-freeness typically means that no agent prefers another agent's share over their own. Traditionally termed the "Cake Cutting Problem," it focused on equitably dividing divisible goods, where fractions of resources could be allocated. It's widely known that an envy-free division of divisible resources, like a cake, can be achieved under general conditions and efficiently computed for various utility functions. However, recent attention has shifted towards achieving envy-free allocations in scenarios involving indivisible goods, where each item can only be allocated entirely to one agent. In such cases, achieving envy-freeness may not always be possible. Our project primarily studies the literature around envy free allocation of indivisible goods. We also pose a new problem which we make an attempt towards solving as a part of this project.

2.1 The General Fair Division Setting

An instance (N, M, V) of a Fair Division Problem (indivisible case) is defined by a set:

- a set N of n agents
- a set M of m indivisible goods
- an n-tuple $V = (v_1, v_2, \ldots, v_n)$ of valuation functions. $v_i : 2^M \rightarrow R$ is the valuation function of the ith agent. It is a mapping from subsets of M to R and is a representation of the value that the i-th agent associates with each subset of M

An allocation $A = (A_1, A_2, \ldots A_n)$ is a partition of M into n subsets ,where A_i is the set of goods that the agent i receives. An allocation is said to be complete if it assigns all items in M, and is called partial otherwise.

Our aim is to find an allocation which is "envy-free".

Note: In the most general case all the items in M need not have a positive marginal value, in which case it is known as a good. However, we only consider the case where all items are goods.

2.2 Defining Envy Freeness and its relaxations

Let $A = (A_1, A_2, \ldots, A_n)$ be an allocation.

1. **Envy freeness(EF)**: A is said to be envy free if $v_i(A_i) \geq v_i(A_j) \forall i, j \in N$
2. **Envy Freeness up to one item(EF_1)**: A is said to be envy free up to one good if $\forall i, j \in N$, there exists g in A_j such that $v_i(A_i) \geq v_i(A_j \backslash g)$.
3. **Envy Freeness up to any Item(EF_x)**: A is said to be envy free up to any item if $\forall i, j \in N$, $v_i(A_i) \geq v_i(A_j \backslash g)\ \forall g \in A_j$.
4. **Equitability(EQ)**: A is said to be equitable (EQ) if $\forall i, j \in N$, $v_i(A_i) = v_i(A_j)$.
5. **Equitability up to one item(EQ_1)**: A is said to be equitable up to one good if $\forall i, j \in N$, $\exists g \in A_j$ such that $v_i(A_i) \geq v_i(A_j \backslash g)$.
6. **Equitability up to any item(EQ_x)**: A is said to be equitable up to any item if $\forall i, j \in N$, $v_i(A_i) \geq v_i(A_j \backslash g)\ \forall g \in A_j$.

2.3 The popularly studied classes of valuation functions

1. **Additive valuations**: A valuation function is called additive if the function value associated with a subset is the sum of the values associated with the individual elements of the set. Mathematically, for additive valuations, $v(S) = \sum_{i \in S} v(i)$.

2. **Monotone valuations**: A valuation function is called monotone if the value of every subset is at most the value of any of its superset. Mathematically, if $S \subseteq T$, then $v(S) <= v(T)$.

3. **Submodular valuations**: A valuation function is called sub-modular if the marginal value of any good is greater with respect to a subset, that it is with respect to the corresponding superset. Mathematically, for any $S \subseteq T$, and $g \notin T$, $v(S \bigcup g) - v(S) \geq v(T \bigcup g) - v(T)$.

2.4 Additional Terminology that has been used

1. **Envy:** Agent i is said to envy agent j if $vi(Ai) < vi(Aj)$.

2. **Envy Graph:** For any partial allocation A of M, we can define an envy graph G(V, E) where $|V| = n$, each vertex represents an agent. $(i, j) \in E$ iff agent i envies agent j.

3. **Preference:** An agent i is said to prefer item r over item s (in an additive setting) if $vi(r) > vi(s)$.

4. **Approval:** In a binary additive valuation setting, any item with value one for some agent is said to be approved by the agent.

2.5 A survey of the studied settings and Results that have been established

1. Envy free allocations may not exist for the indivisible problem setting.

   Proof. Consider the case where $n = 2$, $m = 1$. In any allocation, the one good in M may go to either agent1 or to agent2, the agent that does not receive the good envies the agent who does. Hence, EF can never exist. Hardness Results. □

2. One can very quickly show that it is NP-hard to determine if there exists an EF allocation, even with two agents with identical, additive, monotone valuations.

Proof. Given an instance of partition consisting of integers $a_1, ..., a_n$, we create a fair division instance consisting of 2 agents and n items, where item i has value a_i for both agents. Then it is easy to see that there is an EF allocation (or an EQ allocation) if and only there exist a subset of the integers that have sum $(a_1 + ... + a_n)/2$, as required by the partition instance. □

3. Determining whether a given instance admits an envy-free allocation is NP-complete even for binary utilities.

   Proof. Membership in NP follows from the fact that given an allocation, checking whether it is envy-free can be done in polynomial time.
   To show NP-hardness, we can show a reduction from Set Splitting which is known to be NP-complete and asks the following question: Given a universe U and a family F of subsets of U, does there exist a partition of U into two sets U_1,U_2 such that each member of F is split by this partition, i.e., no member of F is completely contained in either U_1 or U_2? Intricate details of the proof can be found in the references stated underneath. □

4. Round Robin Algorithm for finding EF1 allocations given additive valuation functions:

   Theorem 2.1. *For an instance with additive valuations of agents over a set of goods, an $EF1$ allocation always exists and can be given by the Round Robin Algorithm. Algorithm:*

```
(a) Arrange the n agents around a round table.
(b) $M' \leftarrow M$
(c) Pick an arbitrary starting agent. Assign the agent her most preferred good from M', say g.
    $M' \leftarrow M' \backslash \{g\}$
(d) While M' != empty:
    i. Move the agent pointer to the next agent on the table in counter clockwise direction.
    ii. Assign the agent her most preferred good from M', say g. $M' \leftarrow M' \backslash \{g\}$
(e) Return the allocation
```

   Proof. We first notice that the 1st agent is always envy free since he gets his most preferred good at the beginning of each iteration. If we look at the i-th agent, we see that there could only possibly be envy between that agent with an agent that comes before in the agent order, not with anyone coming later. If we remove the first good allocated to each agent before i, we see that it is just equivalent to starting the cycle with agent i with a smaller set of items. Here the ith agent will be envy free. Hence, by definition, ith agent will be $EF1$ in the original setting.

□

5. Envy Cycle Elimination Algorithm for finding EF1 allocations given monotone valuation functions:

Theorem 2.2. *For any set of goods and any set of players,there exists an allocation A such that the maximum envy of A is bounded by the maximum marginal utility of the goods,α.Furthermore,given oracle access for the utility functions of the players,there is an $O(mn^3)$ time algrithm for finding such an allocation*

Proof. To prove the above theorem we will use the below lemma:

Lemma 2.3. *For any partial allocation A with envy graph G we can find another partial allocation B with envy graph H such that:*

- $e(B) \leq e(A)$
- *H is acyclic*

Here $e(A) = max\{e_{pq}(A), p, q \in N\}$ *where* $e_{pq}(A) = max\{0, v_q(A_q) - v_p(A_p)\}$

We then proceed with the following algorithm:

(a) Allocate good 1 to some player.

(b) At round $(i+1)$ we construct the envy graph corresponding to current allocation.

(c) We use the Lemma 1.3 to obtain an allocation A in which the the maximum envy is atmost α and the new envy graph G is acyclic.

(d) As graph G is acyclic there is a player $p \in N$ with in degree 0 which implies nobody envies p.We allocate good $(i+1)$ to p.Repeat the above steps until all the goods are allocated.

The complete proof can be found in the reference stated at the end of the document. □

3 Problem Statement

We study the special case of the general fair allocation problem and try to find tractable results for the special case, the “k-approvals problem”.

We define an instance (N, M, V) of the k-approvals problem by:

- $N = [n]$, a set of n agents,
- M, a set of m indivisible goods
- $V = (v_1, v_2, \ldots, v_n)$, a set of n valuation functions

Additional constraints:

1. Every v_i is an additive valuation function.
2. $v_i(\{j\}) \in \{0, 1\}$ $\forall i \in N$ and $j \in M$
3. $1 \leq v_i(M) \leq k$ $\forall i \in N$.

All of the additional constraints imposed together can be seen as the “k-approvals problem”, in the sense that each agent gives her approval for utmost k items in M and denies the rest, where $v_i(j) = 1$ is treated as an approval.

Aim: To find if an envy free allocation exists for a given instance of the problem tractably. We are interested in looking at FPT algorithms in the introduced parameter, k.

4 Methodology

4.1 Brute Force Analysis

4.1.1 Exploration of Search Space

The most naive approach to characterising the existence of an EF allocation would be to check all possible allocations (partitions).

4.1.2 Complexity Analysis

For a setting with n agents and m items, each of the m items could go into any of the n agents' bundles. Hence, there are n^m possible partitions. Thus, we will require O(n^m) queries, each query needing O(n^2) time.

4.2 Algorithm for the case where each agent values exactly k=2 items

Observation-01: For the k=2 case, if any agent has a private good (that agent is the only agent who prefers the good), then the good can be allocated to that agent, and both the agent and the good can be removed from the picture. We can assure that the removed agent is envy-free.
Hence, let us consider a setting with no agents with private goods.
Every agent has atleast two preferred goods and every item has atleast two agents preferring it.
$\implies 2n \geq 2m \implies n \geq m$

1. **Case-01:** $n = m$

   Every item has exactly two agents who prefer the item. We get a 2-bipartite graph and hence EF allocation exists.

2. **Case-02:** $n > m$

   This will be the case where in every allocation that we look at, there will be at least one agent who has an empty allocation. Hence, no allocation can be envy free. (see lemma 3.1 given below)

4.3 A cleverer Manifestation of the Search Space

Lemma 4.1. *In any envy free allocation, each agent has at least one of her preferred goods in her bundle.*

Proof. We see that every agent has at least one preferred good in the item set M. If the agent i has none of her preferred goods in her bundle, then her self valuation is zero. But due to non-wastefulness of allocation, some agent must have a positive valuation with respect agent i (due to having one of agent i 's preferred goods). Hence agent i is no longer envy free. □

We will use the lemma proved above to get a reduction in our search space.

4.3.1 Exploration of Search Space

We do not have to explore the partitions in which there exists an agent who does not have any of her preferred goods. We look at partitions using the method of branching.

1. **Case 1**: $m < n$.

   Corollary 4.1. *If $m \leq n$ then there exists no EF allocation.*

   Proof. This follows from the Lemma 3.2 that every agent values at least one item. □

2. **Case 2**: $m = n$

   This is equivalent to finding a bipartite matching in a graph. We can hence use Ford Fulkerson which will help us check if EF allocation exists.

3. **Case 3**: $m > n$

   Suppose every agents prefers atmost k items. We present an algorithm which generates partitions via recursion:

   (a) Check if agent 1 values exactly k items. If not, move to the next agent.

   (b) Let agent i be the first agent that value exactly k items. Allot agent i one of any of her preferred items.

   (c) Remove this preferred item from the list of preferred items of all other agents.

   (d) Find the next agent who values exactly k items.

   (e) Repeat the above process until all agents value at most $k-1$ items. And run the partition generator on (k-1)

We get the recursion: $T(k) = k^n * T(k-1)$ which gives a time complexity of $O(k^{k^n})$

Algorithm 1 *Generate_Partition_*(M, N, V, k)

```
$i = 0$
$M' = M$
**while** $i \quad != N$ **do**
    **if** agent $i$ has exactly $k$ items **then**
        $M = M \backslash \{j$ : such that agent $i$ values item $j\}$
    **end if**
    $i = i + 1$
**end while**
Call $Generate\_Partition(M', N, V, k - 1)$
```

4.4 Special case where each agent values exactly k items

Lemma 4.2. *In the case where each agent values exactly k items,if there exists an EF allocation then every agent i must have value* $v_i(A_i) \geq \lfloor \frac{k_i}{n} \rfloor$ *where* $k_i = v_i(M)$.

Proof. Let $A = (A_1, A_2, ...A_n)$ be an EF allocation for given instance (N, M, V). To the contrary,suppose agent i is such that $v_i(A_i) \leq \lfloor \frac{k_i}{n} \rfloor$.This implies:

$$\sum_{j!=i} v_i(A_j) = v_i(M) - v_i(A_i) > k_i - \lfloor \frac{k_i}{n} \rfloor$$

By the rule of averages,$\exists \quad j! = i$ such that $v_i(A_j) \geq \frac{k_i - \lfloor \frac{k_i}{n} \rfloor}{n-1}$.Therefore we have:

$$v_i(A_j) \geq \frac{n * \frac{k_i}{n} - \frac{k_i}{n}}{n-1} = \frac{k_i}{n} \geq \lfloor \frac{k_i}{n} \rfloor \Rightarrow v_i(A_j) \geq v_i(A_i)$$

We get a contradiction. □

We hope to use the above lemma to reduce the search space.

5 Other Paths Explored to Solve the Problem

5.1 An approach using Integer Linear Programs

- Let:

$$x_{ij} = \begin{cases} 1 & \text{if} \quad \text{agent } i \text{ gets item } j \\ 0 & \quad \text{otherwise} \end{cases} \qquad \mu_{ij} = \begin{cases} 1 & \text{if} \quad \text{agent } i \text{ approves item } j \\ 0 & \quad \text{otherwise} \end{cases}$$

- Our objective function would be to maximize the minimum of negative of ”enviness” between agents.If this turns out to be 0 then we would claim that there is an envy free allocation.

$$max(min_{i,k}(\sum_j x_{kj} * \mu_{ij} - \sum_j x_{ij} * \mu_{ij}))$$

  subject to:

  1. $\sum_j x_{kj} * \mu_{ij} - \sum_j x_{ij} * \mu_{ij} \leq 0 \quad \forall \quad i, k \in [n]$
  2. $\sum x_{ij} = 1 \quad \forall \quad j \in [m]$
  3. $x_{ij} \in \{0, 1\} \quad \forall \quad i \in [n], j \in [m]$

- The above expression could further be simplified by taking $min_{i,k}(\sum_j x_{kj} * \mu_{ij} - \sum_j x_{ij} * \mu_{ij}) = z$.We would then obtain:

$$max \quad z$$

  subject to:

  1. $\sum_j x_{kj} * \mu_{ij} - \sum_j x_{ij} * \mu_{ij} \leq 0 \quad \forall \quad i, k \in [n]$
  2. $z \leq \sum_j x_{kj} * \mu_{ij} - \sum_j x_{ij} * \mu_{ij} \quad \forall \quad i, k \in [n]$
  3. $\sum x_{ij} = 1 \quad \forall \quad j \in [m]$
  4. $x_{ij} \in \{0, 1\} \quad \forall \quad i \in [n], j \in [m]$

- We see that the matrix associated with the ILP is very structured. Firstly, it is a 0-1 matrix. We can also impose restrictions on the sum of rows and columns of the matrix based on the parameter k. The matrix will be sparse matrix due to the parameter. Hence, we speculate that this property could be used to our advantage.

5.2 An approach using Matching

We see, as we explore the branching approaches, that the problem bears an uncanny resemblance to the matching problem, except here it is going to be a two sided matching instead. We can generate a bipartite graph, with one part being the agents, and the other part being the items. We can connect an agent and an item by an edge if the agent prefers the item. This can also be converted into a flow. We can work on investigating how an EF allocation looks like through the lens of a network flow. This would be an interesting problem.

References

[1] *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[2] Umang Bhaskar, Neeldhara Misra, Aditi Sethia, and Rohit Vaish. The price of equity with binary valuations and few agent types. In Argyrios Deligkas and Aris Filos-Ratsikas, editors, *Algorithmic Game Theory - 16th International Symposium, SAGT 2023, Egham, UK, September 4-7, 2023, Proceedings*, volume 14238 of *Lecture Notes in Computer Science*, pages 271–289. Springer, 2023.

[3] Umang Bhaskar, A. R. Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources, 2022.

[4] Robert Bredereck, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier. High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 505–523, New York, NY, USA, 2019. Association for Computing Machinery.

[5] Ioannis Caragiannis, Jugal Garg, Nidhi Rathi, Eklavya Sharma, and Giovanna Varricchio. Existence and computation of epistemic efx allocations, 2022.

[6] Ganesh Ghalme, Xin Huang, Yuka Machino, and Nidhi Rathi. A discrete and bounded locally envy-free cake cutting protocol on trees, 2022.

[7] Pallavi Jain, Krzysztof Sornat, Nimrod Talmon, and Meirav Zehavi. Participatory budgeting with project groups, 2020.

[8] Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. volume 5, pages 125–131, 05 2004.

[9] Fedor Sandomirskiy and Erel Segal-Halevi. Efficient fair division with minimal sharing. *Operations Research*, 70(3):1762–1782, May 2022.