Equitable Core Imputations for Max-Flow, MST and b-Matching Games

Rohith Reddy Gangam
University of California, Irvine
Irvine, USA
rgangam@uci.edu

Naveen Garg*
Indian Institute of Technology Delhi
Delhi, India
naveen@cse.iitd.ac.in

Parnian Shahkar
University of California, Irvine
Irvine, USA
shahkarp@uci.edu

Vijay V. Vazirani
University of California, Irvine
Irvine, USA
vazirani@ics.uci.edu

ABSTRACT

We study fair allocation of profit (or cost) for three central problems from combinatorial optimization: Max-Flow, MST and b-matching. The essentially unequivocal choice of solution concept for this purpose would be the *core*, because of its highly desirable properties. However, recent work observed that for the assignment game, an arbitrary core imputation makes no fairness guarantee at the level of individual agents. To rectify this deficiency, special core imputations, called *equitable core imputations*, were defined — there are two such imputations, *leximin and leximax* — and efficient algorithms were given for finding them.

For all three games, we start by giving examples to show that an arbitrary core imputation can be excessively unfair to certain agents. This led us to seek equitable core imputations for our three games as well. However, the ubiquitous tractable vs intractable schism separates the assignment game from our three games. As is usual in the presence of intractability, we resorted to defining the Owen set for each game and algorithmically relating it to the set of optimal dual solutions of the underlying combinatorial problem. We then give polynomial time algorithms for finding equitable imputations in the Owen set.

The motivation for this work is two-fold: the emergence of automated decision-making, with a special emphasis on fairness, and the plethora of industrial applications of our three games.

KEYWORDS

Cooperative games; Core; Fairness

ACM Reference Format:
Rohith Reddy Gangam, Naveen Garg, Parnian Shahkar, and Vijay V. Vazirani. 2026. Equitable Core Imputations for Max-Flow, MST and b-Matching Games. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/VJNO3737

*Work done while the author was visiting the University of Warwick on a Royal Society Visiting Fellowship.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VJNO3737

1 INTRODUCTION

With automated decision-making emerging as the norm, the design of algorithms which ensure fair allocations to agents has become increasingly important; these are not only ethical, but also lead to customer satisfaction, and hence loyalty, to businesses paying attention to this issue. In this vein, our paper studies the fair allocation of profit or cost for three problems: Max-Flow, MST and b-matching. All three are central to combinatorial optimization, with a plethora of industrial applications, e.g., see [16], [1] and [10]. The corresponding games, which distribute profit (or cost), have also received considerable attention in economics and computer science, e.g., see Moulin [7].

As an example, assume that several agents have built pipes for carrying flow from a source to a sink and the resulting profit needs to be distributed among the agents. What is a disciplined way of doing this? The gold standard solution concept in this regard is that of the *core* which gives each sub-coalition at least as much profit as its inherent worth, hence ensuring that no sub-coalition has incentive to secede, see details in Section 1.1. However as pointed out recently by Vazirani [15], for the assignment game, an arbitrary core imputation[1] makes *no fairness guarantee at the level of individual agents*. Similarly, for the three games stated above, we give examples in Section 2.1 to show that an arbitrary core imputation can be excessively unfair to certain agents. This led us to studying *equitable core imputations*, proposed in [15], which comprise of two special imputations in the core: *leximin and leximax*.

The first work to seek a middle ground in the tension between sub-coalitions demanding profit which is commensurate with their inherent worth and the societal norm of "equality", was the well-known *egalitarian solution* of Dutta and Ray [2]. For a convex game[2], this is the unique core imputation which Lorenz dominates all other core imputations. At a high level, our goal, and that of [15], is analogous to that of Dutta and Ray. Section 1.1 states three shortcomings of the approach of [2] as well as the way they are rectified via the approach of [15] and the current paper.

The ubiquitous tractable vs intractable schism separates the assignment game from our three games, making our task different from that of [15]. The core of the assignment game consists of all optimal solutions to the dual of the LP-relaxation of the maximum

[1] An imputation is a way of distributing profit/cost. See Definition 2.2
[2] The characteristic function of such a game is supermodular.

weight matching problem in the underlying bipartite graph [11]. Consequently, the problem of determining if a given imputation is in the core is in P. However, the latter problem was shown to be co-NP-hard for the max-flow [4] and MST games [3]. Building on these results, we show that finding a leximin or leximax core imputation is also NP-hard for these games. For the b-matching game, we establish a weaker evidence of intractability[3], see Section 5.

On the other hand, for all three games, every optimal solution to the dual LP of the corresponding combinatorial problem leads to a core imputation. Hence, the core of these games is non-empty and can be partitioned into two sets: the well-behaved, tractable part derived from optimal solutions to the dual LP and the rest; the latter is where the phenomenon of NP-hardness shows up. This situation had been encountered in the past and researchers had resorted to studying only the tractable partition[4], naming it the *Owen Set*[5] [8, 13]; we will carry over this name to our paper as well. For all three games, Owen set is convex, i.e., every convex combination of two of its elements is in the set, and therefore, by Lemma 3 in [15], the leximin and leximax imputations in the Owen set are unique.

Results and key ideas:

(1) For the max-flow game, we show that every optimal solution to the dual LP efficiently leads to an imputation in the Owen set. Furthermore, we give a polynomial time algorithm for determining if a given imputation belongs to the Owen set and if so, finding a corresponding optimal dual solution—the analogous result for the b-matching game was given in Soriano et al. [12].

(2) For the MST game, we show that every optimal solution to the dual LP efficiently yields a set of core imputations, hence characterizing its Owen set. Among the various primal-dual formulations for the MST game, we establish that the minimum branching LP serves as the right formulation for defining the Owen set. Despite its exponentially many dual variables on sets, we provide an elegant interpretation that maps them to costs on agents. We also provide a separation oracle that facilitates the use of the ellipsoid method for efficiently verifying core membership. We leave the open problem of finding efficient combinatorial algorithms.

(3) We give a combinatorial, strongly polynomial algorithm for finding the leximin and leximax core imputations in the Owen set of the max-flow game. Our approach leverages the Picard-Queyranne structure to efficiently compute dual optimal solutions corresponding to these imputations. For the b-matching game, we do the same by building on the algorithms of Vazirani [15].

(4) For the MST (and more generally, minimum branching) game, we give LP-based polynomial time algorithms, using the ellipsoid method, for finding the leximin and leximax imputations in the Owen set. Again we leave the open problem of finding combinatorial polynomial time algorithms.

[3]A very recent paper [6] has established co-NP-hardness as well.

[4]This is consistent with the parable *the streetlight effect*, see [17], which sometimes characterizes the way science makes progress.

[5]Owen Set uses the solutions to the dual LP of the underlying combinatorial optimization problem to construct core imputations.

1.1 Background information

The *core* distributes the total worth of a game among the agents in such a way that the profit received by a sub-coalition is at least as large as the profit which the sub-coalition can generate all by itself. This not only ensures stability of the grand coalition, since no sub-coalition has an incentive to secede, but ensure some degree of fairness, namely to each of exponentially many sub-coalitions – a stringent requirement indeed. Additionally, the core also provides profound insights into the negotiating power of individuals and sub-coalitions, see [7, 14] and Remark 3.6.

However, as pointed out recently in the context of the assignment game [15], which forms a paradigmatic setting for studying the core — in large part due to the classic work of Shapley and Shubik [11] — an arbitrary core imputation makes no fairness guarantee at the level of individual agents. This is due to the fact that a singleton sub-coalition (or a set of players from the same side of the bipartition), can make zero profit, and therefore its profit under a core imputation can be an arbitrary amount.

It is well known that "fairness" can be defined in many ways, depending on the setting. Among these, the use of max-min and min-max fairness is widespread, e.g., in game theory, networking and resource allocation. A leximin allocation maximizes the smallest component and subject to that, it maximizes the second smallest, and so on. It therefore goes much further than a max-min allocation. Similarly a leximax allocation goes much further than a min-max allocation. [15] defined leximin and leximax core imputations as *equitable core imputations.* These two imputations achieve equality in different ways: whereas leximin tries to make poor agents more rich, leximax tries to make rich agents less rich, thereby indirectly making poor agents more rich, hence they may be better suited for different applications.

The egalitarian solution of Dutta and Ray [2] has three shortcomings: it does not apply to several key natural games, including the assignment, MST and max-flow games, since they are not convex; it is not efficiently computable for any non-trivial, natural game; and the Lorenz order is a partial, and not a total, order. As stated above, Vazirani [15] and our work rectify all three shortcomings. (Due to space constraints, detailed discussion of related work is provided in the full version of the paper [5].)

2 PRELIMINARIES

Definition 2.1. Let N be a set of agents. A cooperative game on N is defined by a *characteristic function* $v : 2^N \to \mathbb{R}_+$, where for each $S \subseteq N$, $v(S)$ is the value (or cost) that the sub-coalition S can produce on its own. N is also called the grand coalition.

In the three games that we consider—the max-flow game, the min-cost branching game and the max-weight b-matching game—the set of agents is either the vertex or the edge set of a graph G. We will use (G, v) to represent these games, where v is the characteristic function of the specific game. Depending on the game $v(S)$ is the profit or cost of the coalition $S \subseteq N$.

Definition 2.2. An *imputation* $p : N \to \mathbb{R}_+$ is a partition of the value (or cost) of the game, $v(N)$, among the agents in N. $p(i)$ is the *share* of agent i and $\sum_{i \in N} p(i) = v(N)$.

Definition 2.3. An imputation p is in the *core* of a profit (or cost)-sharing game (N, v) if and only if for every sub-coalition $S \subseteq N$, the total profit (or cost) shares of the members of S is no less (or no more) than the value of S, i.e., $\sum_{i \in S} p(i)$ is at least (or at most) $v(S)$.

Definition 2.4. Let P be a set of imputations of a game (N, v) and $p_1, p_2 \in P$. Let l_1, l_2 be the lists formed by arranging the shares of agents in p_1, p_2 in ascending order. l_1 is *lexicographically larger* than l_2 if l_1 has the larger value at the first index where the two lists differ. The imputation in P which is lexicographically larger than all other imputations in P is the *lexicographically minimum* or *leximin* imputation in P.

Definition 2.5. Let P be a set of imputations of a game (N, v) and $p_1, p_2 \in P$. Let l_1, l_2 be the lists formed by arranging the shares of agents in p_1, p_2 in descending order. l_1 is *lexicographically smaller* than l_2 if l_1 has the smaller value at the first index where the two lists differ. The imputation in P which is lexicographically smaller than all other imputations in P is the *lexicographically maximum* or *leximax* imputation in P.

If the set P in the definitions above is the set of core imputations in the game, then the imputations are called *leximin* and *leximax core imputation* respectively. Similarly, the respective imputations for the Owen set imputations will be called the *leximin* and the *leximax Owen set imputations*.

2.1 The games and their unfair core imputations

Max-flow game. An instance of the s-t maximum flow problem is given by a directed graph $G = (V, E)$, a source vertex $s \in V$, a sink vertex $t \in V$ and edge capacities $c : E \to \mathbb{R}_+$. A flow $f : E \to \mathbb{R}_+$ is a function on the edges of G that satisfies the following constraints.

Capacity constraint: The flow f_e through any edge e must not exceed its capacity c_e, i.e. $\forall e \in E, f_e \leq c_e$.

Conservation: For every vertex v except s and t, the flow entering v equals the flow leaving v, i.e., $\forall v \in V \setminus \{s, t\}, \sum_{u:(u,v) \in E} f_{uv} = \sum_{w:(v,w) \in E} f_{vw}$.

The value of flow f is the sum of the flow on the edges leaving source s or entering sink t (these sums are equal due to flow conservation). The objective is to find a maximum flow f.

The max-flow game is defined over an instance of the maximum s-t flow problem and has an agent for each edge in G. The value/profit of the grand coalition, $v(E)$ is the maximum s-t flow. The profit of a subset of agents $E' \subseteq E$, is the maximum s-t flow in the subgraph induced by the edges E', with the same capacities of edges. An imputation is a distribution of the total profit to the edges/agents and an imputation is in the core if no subset gets a profit less than its value.

MST game and min-cost branching game. Let $G = (V, E)$ be a directed graph, $c : E \to \mathbb{R}_+$ a cost function on the edge and $r \in V$ a root vertex. A *branching* is a subset of edges $E' \subseteq E$ such that there is a path from every vertex in $V \setminus \{r\}$ to r using edges in E'. The cost of a branching is the sum of the costs of the edges in the branching. The value of a set $S \subseteq V \setminus \{r\}$, $v(S)$, is the minimum cost branching in $G(S \cup r)$ and defines the characteristic function of the game.

Similarly, if the graph G is undirected and the worth of a set S is given by the minimum cost spanning trees involving the vertices of S and the root, then the game is called min-cost spanning tree (MST) game. This is trivially a special case of the min-cost branching game. We will use (G, v) to define the MST or min-cost branching game—vertices of the graph, minus the root, cover the player set and the min-cost branching or the spanning tree formed by the vertices with the root fix the characteristic function of the game.

Let T be a branching (or a spanning tree) of the minimum cost in G. An assignment $s : V \to \mathbb{R}_+$ is a *cost-share* or a *imputation* if $\sum_{v \in V \setminus \{r\}} s(v) = c(T)$. An imputation is in the core of the min-cost branching (or MST) game if $\forall S \subseteq V \setminus \{r\}, \sum_{e \in T'} c(e) \geq \sum_{v \in S} s(v)$ where T' is the minimum cost branching (or MST) in the subgraph induced over $S \cup \{r\}$.

Max-weight bipartite b-matching game. Consider a bipartite graph $G = (U, V, E)$ with an associated edge-weight function $w : E \to \mathbb{R}_+$. The capacity function $b : U \cup V \to \mathbb{Z}_+$ sets a cap on the number of matches a vertex can participate in. While edges may be matched multiple times, the vertex limitations dictated by b inherently set restrictions on the edges as well. Consequently, edge (i, j) can be matched up to $\min\{b(i), b(j)\}$ times. Any choice of edges, with multiplicity, subject to these constraints, is called a b-matching.

In the context of the *max-weight bipartite b-matching game*, "the b-matching game" for short, the *value* of a coalition $(S_u \cup S_v)$, where $S_u \subseteq U$ and $S_v \subseteq V$, is defined by the maximum weight of a b-matching within the subgraph of G limited to $(S_u \cup S_v)$ alone. This value is represented by $v(S_u \cup S_v)$, which forms the *characteristic function* of the game, with $v : 2^{U \cup V} \to \mathbb{R}_+$. An *imputation* is made up of two mappings $p_U : U \to \mathbb{R}_+$ and $p_V : V \to \mathbb{R}_+$ ensuring that $\sum_{u \in U} p_U(u) + \sum_{v \in V} p_V(v) = v(U \cup V)$. The definition of the core, as stated in Definition 2.3, remains the same. The particular instance of the bipartite b-matching game where b is a constant function is called the *uniform bipartite b-matching game*, with the constant represented by $b_c \in \mathbb{Z}_+$. If $b(v) = 1, \forall v \in U \cup V$, then it is the special case of *assignment game*, which is the focus of [11] and [15].

Unfair imputations for these games. In each of these games, there are core imputations that may be considered very unfair. For example, in the max-flow game, consider a path of n unit-capacity edges from the source to the sink. An imputation that allocates all the profit to a single edge while giving nothing to the others is in the core, but a fairer imputation would distribute the profit equally among all edges, giving each a share of $1/n$. Similarly, in the MST game, imagine n vertices connected to the root by a path of n unit-cost edges. An imputation that charges all the cost to the farthest vertex is in the core, but an equitable choice would be to charge each agent one unit of cost. Shapley and Shubik ([11]) demonstrated that in the assignment game, a special case of b-matching game, the set of core imputations corresponds precisely to the optimal solutions of the dual LP-relaxation of the maximum weight matching problem. Moreover, they showed that this set of core imputations forms a lattice, where the extreme points tend to disproportionately favor one side.

Proofs of all lemmas and theorems marked † are provided in the full version of the paper [5]. Although linear programming (LP) yields efficient procedures for computing fair Owen-set imputations in our games—indeed, [5] introduces a new LP formulation that extracts the leximin vector of the associated polytope—the central contribution of this paper is a collection of combinatorial algorithms, described in detail below.

3 MAX-FLOW GAME

We would like to obtain the leximin (and leximax) fair imputations in the core of the Max-flow game. But, a corollary of the following theorem is that finding these imputations is NP-hard. And so, we focus on the Owen set of the max-flow game and consider the linear programming (LP) formulation of the max-flow problem.

Theorem 3.1. † *Finding a core imputation that maximizes the minimum profit-share of any edge in a max-flow game is NP-hard. Similarly, it is NP-hard to find a core imputation that minimizes the maximum profit-share of any edge.*

3.1 LP formulation of the max-flow problem

Consider a max-flow instance on a directed graph $G = (V, E)$, a source vertex $s \in V$, a sink vertex $t \in V$ and edge capacities $c : E \longrightarrow \mathbb{R}_+$. The maximum s-t flow problem can be formulated as a circulation by introducing an edge of infinite capacity from t to s and ensuring flow conservation at all vertices including s, t. The objective of finding a flow of maximum value can be restated as finding a circulation which maximizes the flow in the edge (t, s). A linear program (LP) for the maximum flow problem is formulated by associating a variable f_{ij} with edge $(i, j) \in E$ whose value equals the flow through this edge.

$$\begin{aligned} \text{maximize} \quad & f_{ts} \\ \text{subject to} \quad & \sum_{j:(j,i)\in E} f_{ji} - \sum_{j:(i,j)\in E} f_{ij} \leq 0 \quad \forall i \in V, \\ & 0 \leq f_{ij} \leq c_{ij} \quad \forall (i,j) \in E \end{aligned} \tag{1}$$

The first set of inequalities, taken together for all vertices ensures the conservation of flow while the second set of inequalities imposes the capacity constraints. The dual of the maxflow LP is obtained by associating a potential π_i with vertex $i \in V$ and a length δ_{ij} with edge $(i, j) \in E$.

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} c_{ij}\delta_{ij} \\ \text{subject to} \quad & \delta_{ij} - \pi_i + \pi_j \geq 0 \quad \forall (i,j) \in E, \\ & \pi_s - \pi_t \geq 1, \\ & \delta_{ij} \geq 0 \quad \forall (i,j) \in E, \\ & \pi_i \geq 0 \quad \forall i \in V \end{aligned} \tag{2}$$

The first set of constraints requires that the potential drop across any edge be at most the length of the edge. The second constraint requires that the potential difference between s, t is at least 1 and these constraints together imply that the length of any path from s to t under the length function δ is at least 1. Thus the dual LP can be viewed as an assignment of lengths to edges, $\delta : E \longrightarrow \mathbb{R}_+$, that minimizes $\sum_{(i,j)\in E} \delta_{ij}c_{ij}$ subject to the condition that the length of any path from s to t is at least 1. If δ is 0/1, then every path from s to t has an edge of length 1 and hence edges of length 1 form a cut separating s and t; this set of edges is an s-t cut. For this reason, the assignment δ is also called a fractional s-t cut and its capacity is $\sum_{(i,j)\in E} \delta_{ij}c_{ij}$. By the strong duality theorem, the capacity of the minimum fractional s-t cut equals the maximum s-t flow.

3.2 Owen set for the max-flow game

The central idea is to identify the subset of the core that corresponds to optimal solutions of the dual linear program (LP), referred to as the **Owen set**. Let (δ, π) represent an optimal dual solution. Define the *profit* of an edge (i, j) as: $p_{ij} := c_{ij} \cdot \delta_{ij}$ This defines an imputation because, by the LP duality theorem, the worth of the game which is the maximum flow, equals the objective function value of an optimal dual solution. Specifically, worth$(E) = \sum_{(i,j)\in E} c_{ij} \cdot \delta_{ij}$. An imputation is classified as an **Owen set imputation** if and only if there exists an optimal dual solution (δ, π) such that: $p_{ij} := c_{ij} \cdot \delta_{ij}$.

Theorem 3.2. † *For the max-flow game, the Owen set is a subset of the core.*

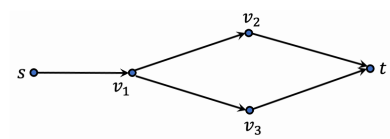

Figure 1: The graph for Example 3.3.

Example 3.3. Consider the graph given in Figure 1. Assume that the capacities of edges (v_1, v_2) and (v_1, v_3) are 1 each. An easy way of showing that there is a core imputation which does not correspond to an optimal dual solution is the following: Assume that the rest of the edges have very high capacities, e.g., 10 each. Assign a profit of 2 to edge (s, v_1) and zero to the rest; observe that the worth of the game is 2. Clearly, the dual will not assign edge (s, v_1) a positive distance label, since a max-flow does not saturate it.

Theorem 3.4. † *Deciding if an imputation of the max-flow game is in the Owen set can be done efficiently.*

Definition 3.5. An edge $e \in E$ is *essential* if it is saturated in every s-t max-flow and it is called *inessential* if there is an s-t max-flow that does not saturate e.

Remark 3.6. *If the capacity of an essential edge is dropped by (a small) ϵ, the worth of the game also drops by ϵ. However, dropping the capacity of a non-essential edge by ϵ will not lead to any change in the worth of the game. Therefore, only essential edges have negotiating power and, by complementarity slackness conditions, profits assigned by an Owen set imputations are consistent with this.*

3.3 Computing the leximin Owen set imputation

3.3.1 High level ideas. Let $G = (V, E), s, t \in V, c : E \longrightarrow \mathbb{R}_+$ be an instance of the max-flow game. Computing the leximin Owen set imputation involves three key ideas:

(1) **Dual LP and Potentials:** Using dual LP 2, we work with vertex potentials instead of edge distance labels. Optimal potentials corresponding to the leximin imputation are computed iteratively, building distance labels and profits from these potentials.
(2) **Iterative Optimization:** In each iteration, vertices and their optimal potentials that maximize the next minimum profit share are obtained. The key idea is that the vertices chosen in each iteration will correspond to certain longest paths in a related graph.
(3) **Efficient Longest Path Computation:** Computing longest paths, like the Hamiltonian path, might take exponential time. To avoid this, we leverage the "Picard-Queyranne structure" of G. This is a Directed Acyclic Graph (DAG) which retains all essential edges and the max-flow of G. This enables us to efficiently compute the required paths.

3.3.2 Picard-Queyranne Structure. Our algorithm first starts by creating a Picard Queyranne structure of the graph, defined as follows.

Definition 3.7. Let $G = (V, E), s, t \in V$ be a max-flow instance. Let f be a maximum s-t flow in G and $G|f$ be the residual graph of G under f.[6] Construct $G' = (V', E')$ from $G|f$ by shrinking each maximal strongly connected component in $G|f$ into a vertex and removing any self-loops. Resulting graph $G' = (V', E')$ is the *Picard-Queyranne structure* of G.

The resulting graph G' will be an acyclic multigraph. Each vertex $i \in V'$ is a strongly connected component in $G|f$ and let $\phi(i) \subseteq V$ be the vertices in this strongly connected component. There is no path from s to t in $G|f$; hence, s, t are in distinct strongly connected components of $G|f$. Therefore $s' \neq t'$ where $s' = \phi^{-1}(s)$ and $t' = \phi^{-1}(t)$.

Remark 3.8. *G' is exactly the graph described by Picard and Queyranne ([9]), and so, we call it the Picard-Queyranne structure. They prove that G' is independent of the chosen max-flow f and that every $s' - t'$ cut of the graph G' corresponds to a min s-t cut of the graph G.*

Define F' and Z' as the sets of edges corresponding to the edges that carry the full flow ($f_{ij} = c_{ij}$) and the zero flow ($f_{ij} = 0$) respectively in G under f. Formally, $F' = \{(j, i) \in E', f_{ij} = c_{ij}\}$ and $Z' = \{(i, j) \in E', f_{ij} = 0\}$. The construction of the graph ensures only these edges are preserved in G'—all other edges will removed when the strongly connected components are shrinked, i.e.,

Lemma 3.9. † $E' = F' \cup Z'$.

Note that F' is defined as the set of edges that carry full flow in G, i.e., they are the set of essential edges in G, although pointed in reverse direction.

Corollary 3.10. *All essential edges of G are preserved in G'.*

3.3.3 Algorithm. As mentioned above, the algorithm works in iterations and in each iteration, it fixes the potentials of vertices along some path. Let us first define a few terms in relation to this.

[6]Recall that the *residual graph* $G|f$ is obtained from G by reducing the capacity of every edge $(i, j) \in E$ from c_{ij} to $c_{ij} - f_{ij}$ and introducing edge (j, i) with capacity f_{ij} for every edge $(i, j) \in E$ with $f_{ij} > 0$. Edges with zero capacity are then deleted.

Definition 3.11. For a given iteration, Fixed is the set of all the vertices with assigned potentials and Free is the set containing all remaining vertices in V'.

Initially, the source and sink vertices of G', s' and t' are given a potential of 1 and 0 and are in Fixed. Fixed and Free change with each iteration.

Definition 3.12. A *Free Path* is sequence of vertices $a = v_0, v_1, v_2, \ldots, v_{n-1}, v_n = b$ with $(v_i, v_{i+1}) \in E', \forall i \in \{0, 1, 2, \ldots, n-1\}$ such that $a, b \in$ Fixed and $v_1, v_2, \ldots, v_{n-1} \in$ Free.

We first give new length function l to edges in G' such that $l_e = \frac{1}{c_e}$ if $e \in F'$, and $l_e = 0$ otherwise. Our algorithm, see Figure 2, will run a sequence of iterations and, in each iteration, it finds the longest free path (under lengths l_e) between every pair of fixed vertices. It then assigns potentials to vertices on this path such that every essential edge gets equal profit share, under the rule $\delta_{ij} = \max\{\pi_i - \pi_j, 0\}$, and the potential difference across every non-essential edge is zero—giving them zero profit.

Intuition: To understand why this algorithm gives us the leximin imputation, consider a free path P_{uv} in $\mathcal{P}_{uv}$—the set of all free paths between u and v—after i iterations of the algorithm. If the free vertices on this path were the only free vertices left, we would maximize the next minimum profit share by choosing potentials in a way that gives equal profits, say α_{uv} to all the essential edges and zero to the rest. But, the potential difference across this path is fixed at $\pi_v - \pi_u$, which when split across edges gives - $\pi_v - \pi_u = \sum_{(i,j)\in P_{uv}\cap F'}(\pi_j - \pi_i) + \sum_{(i,j)\in P_{uv}\cap Z'}(\pi_j - \pi_i) \geq \sum_{(i,j)\in P_{uv}\cap F'} \frac{\alpha_{uv}}{c_{ij}} = \sum_{(i,j)\in P_{uv}}\{\alpha_{uv} \cdot l_{i,j}\}$.

For the inequality, we use $\delta_{ij} \geq \pi_i - \pi_j$ for the essential edges and $\delta_{ij} \geq 0$ for the non-essential edges. So, to maximize the profit, potential difference across the non-essential edges should be zero and potential difference across the essential edges should be split proportional to $1/c_e$. Our length function, l_e, was chosen to combine these constraints, and so, we get $\alpha_{uv} = \frac{\pi_v - \pi_u}{\sum_{e\in P_{uv}} l_e}$, to be the maximum profit to the edges this path can provide. And so, the longest free path in G' is the free path with worst potential constraints that ensures minimum profit, α_{uv}, to the edges.

Since G' is acyclic, such a path can be computed efficiently using topological sort. Let $a, b \in$ Fixed after $(i-1)$ iterations and $P \in \mathcal{P}_{ab}$ a path for which the minimum profit is achieved. For vertices $j, k \in P$ let $P[j, k]$ be the subpath of P from j to k. For all vertices j on P we assign $\pi_j = \pi_a + \alpha_{ij} \sum_{e\in P[a,j]} l_e$. Iteration i ends with adding vertices of P to Fixed and the algorithm ends when all vertices are in Fixed.

Claim 3.13. † $\forall (i, j) \in E', \pi_i \leq \pi_j$.

The above claim shows that the potentials conform to the dual by proving a useful property of π. The potential assigned to vertices in V' is extended to vertices in V by assigning all vertices in $\phi(i)$ the potential π_i. The potentials on vertices in V are in turn used to assign lengths to edges in E; for edge $(i, j) \in E$, let $\delta_{ij} = \max\{0, \pi_i - \pi_j\}$. Claim 3.14 proves that the profit share is an Owen set imputation and then Lemma 3.15 shows that it is also the leximin imputation. Theorem 3.16 shows that algorithm 2 is efficient.

Claim 3.14. † *(π, δ) is an optimum solution to LP 2.*

Finding leximin Owen set imputation in max-flow game:
Input: $G = (V, E)$, $s, t \in V$, $c : E \to \mathbb{R}_+$.
(1) **Initialization:**
  (a) $G' = (V', E') \leftarrow$ Picard-Queyranne structure of G.
  (b) $\pi(s') \leftarrow 1; \pi(t') \leftarrow 0$.
  (c) Fixed $\leftarrow \{s', t'\}$; Free $\leftarrow V' \setminus$ Fixed.
  (d) $\forall e \in E', l_e = \frac{1}{c_e}$ if e is essential and 0 otherwise.
(2) **While** Fixed $\neq V'$ **do:**
  (a) **For** every pair $u, v \in$ Fixed **do:**
    (i) Compute longest free path, P_{uv}, between them.
    (ii) Compute profits of essential edges, $\alpha_{uv} = \frac{\pi_v - \pi_u}{\sum_{e \in P_{uv}} l_e}$, on P_{uv}.
  (b) Set potentials of vertices on free path P_{ab} with the minimum profit α_{ab}.
  (c) Update Fixed and Free.
(3) **Output:** The imputation p, where $\forall (i, j) \in E, p_{ij} = c_{ij} \cdot \max\{\pi_i - \pi_j, 0\}$.

Figure 2: Algorithm to find the leximin Owen set imputation in a max-flow game.

Lemma 3.15. † *The imputation defined by the solution (π, δ) is the leximin Owen set imputation for the s-t max-flow game.*

Theorem 3.16. † *There exist $O(mn^2)$ run time algorithms to find the leximin and leximax Owen set imputations of the max-flow game on a graph with n nodes and m edges.*

4 MST GAME AND MIN-COST BRANCHING GAME

In this section, we will solve the general case of the min-cost branching game. The undirected version, the min-cost spanning tree game is a special case of this and can be solved easily by replacing each undirected edge with two directed edges. Note that, unlike the max-flow game, we deal with costs instead of profits in these games. Firstly, we would like to obtain the leximin (and leximax) fair imputations in the core of these games. But, a corollary of the following theorem is that finding these imputations is NP-hard.

Theorem 4.1. † *Finding a core imputation that maximizes the minimum cost-share of any vertex in an MST game or a min-cost branching game is NP-hard. Similarly, it is NP-hard to find a core imputation that minimizes the maximum cost-share of any vertex.*

4.1 Owen set for the min-cost branching game

Let $G = (V, E)$ be a directed graph, $c : E \to \mathbb{R}_+$ a cost function on the edge and $r \in V$ a root vertex. The value/worth of a set $S \subseteq V \setminus \{r\}$, $v(S)$, corresponds to the minimum cost branching in the subgraph of G restricted to $(S \cup \{r\})$ and defines the characteristic function of the game.

The problem of finding a minimum-cost branching can be formulated as an integer program. Let $x(e) = 1$ if $e \in E$ is included in the branching and is 0 otherwise. Let $\nabla(S) \subseteq E$ be the edges with tail in S and head in $\overline{S}$. For every $S \subseteq V \setminus \{r\}$, at least one edge from the set $\nabla(S)$ should be contained in any branching and hence $\sum_{e \in \nabla(S)} x(e) \geq 1, \forall S \subseteq V \setminus \{r\}$. The integer program which minimizes $\sum_{e \in E} x(e)c(e)$ subject to the above constraint yields a minimum cost branching in G. The LP-relaxation of the integer program replaces the integrality constraint $x(e) \in \{0, 1\}$ with $x(e) \geq 0$.

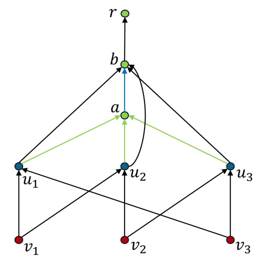

Figure 3: The green edges have cost 0 and the black edges are unit cost. The blue edge has cost 2.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} x(e)c(e) \\
\text{subject to} \quad & \sum_{e \in \nabla(S)} x(e) \geq 1 \quad \forall S \subseteq V \setminus \{r\}, \\
& x_e \geq 0 \quad \forall e \in E
\end{aligned} \tag{3}
$$

The dual of this LP has a non-negative variable $y(S)$ for $S \subseteq V \setminus \{r\}$.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{S \subseteq V \setminus \{r\}} y(S) \\
\text{subject to} \quad & \sum_{S: e \in \nabla(S)} y(S) \leq c(e) \quad \forall e \in E, \\
& y(S) \geq 0 \qquad \forall S \subseteq V \setminus \{r\}
\end{aligned} \tag{4}
$$

Let y be a feasible dual solution. The function $y' : 2^V \times V \to \mathbb{R}_+$ is a *split* of y if $\forall S \subseteq V \setminus \{r\}$, $\sum_{v \in S} y'(S, v) = y(S)$ and $y'(S, v) > 0$ implies $v \in S$. A cost-share s is *consistent with the dual solution y* if there exists a split y' of y such that for all $v \in V \setminus \{r\}$, $s(v) = \sum_{S: v \in S} y'(S, v)$. We say that a cost-share (/imputation), s, is in the Owen set of Min-cost branching game if there exists a feasible dual solution y such that s is consistent with y.

Lemma 4.2. † *For the min-cost branching game, the Owen set is a subset of the core.*

However, not all core imputations are in Owen set and the following example (see Figure 3) establishes this. Consider a graph with vertices $V = \{v_1, v_2, v_3, u_1, u_2, u_3, a, b, r\}$ and edges $E = E_1 \cup E_2 \cup E_3 \cup \{(a, b), (b, r)\}$, where $E_1 = \{(v_1, u_1), (v_1, u_2), (v_2, u_2), (v_2, u_3), (v_3, u_3), (v_3, u_1)\}$,
$E_2 = \{(u_1, a), (u_2, a), (u_3, a)\}$ and $E_3 = \{(u_1, b), (u_2, b), (u_3, b)\}$. All edges in E_2 have cost 0 and edge (a, b) has cost 2. All other edges have unit cost. The root is r.

It is easy to check that a cost-share of 2 to each vertex in the set $X = \{v_1, v_2, v_3\}$ and 0 to all other vertices is a core imputation. However, this cost-share is not in the Owen set. Note that the minimum cost branching in this instance has cost 6 and this is also the total cost-share of vertices in X. Hence, if there exists a feasible dual solution y which can be split to obtain this cost-share, then $y(S) > 0$ implies $S \cap X \neq \phi$.

Note that if $S \cap X \neq \phi$ and $b \notin S$ then $|\nabla(S) \cap (E_1 \cup E_3)| \geq 2$. Since total cost of edges in $E_1 \cup E_3$ is 9, the total y-value of sets S such that $S \cap X \neq \phi, b \notin S$ is at most 4.5. The total y-value of sets S such that $S \cap X \neq \phi, b \in S$ is at most 1 since $(b, r) \in \nabla(S)$. Thus the total y-value of sets S such that $S \cap X \neq \phi$ is at most 5.5 which implies that the total cost-share of vertices in T in any Owen set imputation is at most 5.5. Thus the imputation which assigns a cost-share of 2 to each vertex in T is not in Owen set.

Lemma 4.3. † *Given an imputation, we can efficiently decide if it is in the Owen set by checking if their exists a corresponding dual optimal solution.*

4.2 Finding leximin Owen set imputation using linear programming

We begin by formulating a linear program (LP) for finding an Owen set imputation that maximizes the minimum cost-share of any vertex. Our LP has non-negative variables $y(S, v), v \in S, S \subseteq V \setminus \{r\}$, which is the share of vertex v in the dual variable $y(S)$. Thus $y(S) = \sum_{v \in S} y(S, v)$ and hence the feasibility of the dual solution can be captured by the constraints $\sum_{S: e \in \nabla(S)} \sum_{v \in S} y(S, v) \leq c(e), \forall e \in E$. The cost-share of a vertex v equals $\sum_{S: v \in S} y(S, v)$ and since we wish to maximize the minimum cost we should include the constraint $\sum_{S: v \in S} y(S, v) \geq \lambda, \forall v \in V \setminus \{r\}$, and maximize λ.

To ensure that $p(v) = \sum_{S: v \in S} y(S, v)$ is an imputation we require that

$$\sum_{v \in V \setminus \{r\}} p(v) = \sum_{v \in V \setminus \{r\}} \sum_{S: v \in S} y(S, v) = \sum_{S \subseteq V \setminus \{r\}} y(S) = \mathtt{opt}$$

where $\mathtt{opt}$ is the cost of the minimum cost branching in G.

We now formulate a series of LPs designed to return a leximin Owen set imputation. A corresponding set of LPs for finding a leximax Owen set imputation, along with their respective separation oracle, can be found in in the full version of the paper [5]. Consider the following LP that returns an Owen set imputation that maximizes the minimum cost-share.

$$\begin{aligned} \text{maximize} \quad & \lambda \\ \text{subject to} \quad & \sum_{S \subseteq V \setminus \{r\}: e \in \nabla(S)} \sum_{v \in S} y(S, v) \leq c(e) \quad \forall e \in E, \\ & \sum_{S \subseteq V \setminus \{r\}: v \in S} y(S, v) \geq \lambda \quad \forall v \in V \setminus \{r\}, \\ & \sum_{S \subseteq V \setminus \{r\}} \sum_{v \in S} y(S, v) = \mathtt{opt}, \\ & y(S, v) \geq 0 \quad \forall S \subseteq V \setminus \{r\}, \forall v \in S \end{aligned} \tag{5}$$

Since the number of variables in this LP is exponentially large we consider the dual of this LP that has variables $z(e), e \in E$, $z(v), v \in V \setminus \{r\}$, and β.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} z(e)c(e) - \beta \mathtt{opt} \\ \text{subject to} \quad & \sum_{e \in \nabla(S)} z(e) \geq z(v) + \beta \quad \forall S \subseteq V \setminus \{r\}, \forall v \in S, \\ & \sum_{v \in V \setminus \{r\}} z(v) = 1, \\ & z(e) \geq 0 \quad \forall e \in E, \\ & z(v) \geq 0 \quad \forall v \in V \setminus \{r\} \end{aligned} \tag{6}$$

Although LP 6 has exponentially many constraints, it can be solved in polynomial-time using the ellipsoid method since there is a polynomial-time separation oracle to identify a violating constraint. Given an assignment of values to edges $z(e), e \in E$ and vertices $z(v), v \in V, \sum_{v \in V \setminus \{r\}} z(v) = 1$ and a value β, we should evaluate whether the first constraint from LP 6 corresponding to some $S \subseteq V \setminus \{r\}$ and some $v \in S$ fails.

Seperation Oracle: Let $\alpha(v)$ be the maximum flow from v to r in the graph G with edge capacities given by $z(e), e \in E$. If $\alpha(v)$ is less than $z(v) + \beta$ then by the max-flow min-cut theorem there exists a set $S \subseteq V \setminus \{r\}, v \in S$ such that $\sum_{e \in \nabla(S)} z(e) < z(v) + \beta$. The minimum cut separating v and r then identifies the violated inequality. If for all $v \in V \setminus \{r\}$, $\alpha(v) \geq z(v) + \beta$ then no constraint is being violated.

Using this separation oracle, we can find the maximin λ value using the ellipsoid method. Using the technique from [5], we can also find a set of vertices that attain this value in the leximin solution—all these vertices must have a positive $z(v)$ and the constraint $\sum_{v \in V \setminus \{r\}} z(v) = 1$ implies there is at least one such vertex. We can now iteratively update the primal and dual LPs to maximize the second minimum, third minimum and so on till we obtain the leximin imputation. A complete overview of this procedure and the extension leximax core impuation are discussed in [5].

Theorem 4.4. *The leximin and leximax Owen set imputations for the min-cost branching game can be found efficiently.*

While a combinatorial algorithm to find the leximin Owen set imputation might exist, we prove that such an algorithm for a slightly generalized problem—given a set T of vertices, find an Owen set imputation which maximizes the minimum cost-share of a vertex in T—will yield a combinatorial algorithm for finding the optimum fractional set cover. The latter problem is well-researched and while there are combinatorial algorithms known to compute a $(1 + \epsilon)$ approximation to the optimum value, the only algorithm known for determining the exact value is to solve the LP.

Theorem 4.5. † *A combinatorial algorithm that maximises the minimum cost-share of a vertex in T in an Owen set imputation would yield a combinatorial algorithm for finding the optimum fractional set cover.*

5 b-MATCHING GAME

We would like to obtain the leximin (and leximax) fair imputations in the core of these games. But the following lemma states that finding a sub-coalition failing core constraint for an "approximate" profit share is NP-hard. We expect this NP-hardness result to extend to normal profit shares and to computing leximin (and leximax) core imputations in the same way as with the max-flow and MST games.[7]

Lemma 5.1. † *Given a b-matching game on a bipartite graph $G = (U, V, E), w : E \to \mathbb{R}_+, b : U \cup V \to \mathbb{Z}_+$ and an "approximate" profit-share $p : U \cup V \to \mathbb{R}_+$ such that $p(U \cup V) = \alpha \cdot v(U \cup V)$, deciding if there exists a coalition $S \subseteq U \cup V$ such that $p(S) < v(S)$ is NP-complete for any $\alpha > 1$.*

5.1 Owen set imputations for the b-matching game

The linear program, LP 7, depicts the LP relaxation for finding a max-weight bipartite b-matching. In this formulation, the variable x_{ij} indicates the degree to which edge (i, j) is included in the solution. Note that the variables x_{ij} are not bounded above as an edge may be matched more than once. Taking u_i and v_j to be the dual variables for the first and second constraints of (7), we obtain the dual LP:

$$\begin{aligned} \text{maximize} \quad & \sum_{(i,j)\in E} w_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{(i,j)\in E} x_{ij} \le b_i \quad \forall i \in U, \\ & \sum_{(i,j)\in E} x_{ij} \le b_j \quad \forall j \in V, \\ & x_{ij} \ge 0 \quad \forall (i,j) \in E \end{aligned} \tag{7}$$

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in U} b_i u_i + \sum_{j \in V} b_j v_j \\ \text{subject to} \quad & u_i + v_j \ge w_{ij} \quad \forall (i,j) \in E, \\ & u_i \ge 0 \quad \forall i \in U, \\ & v_j \ge 0 \quad \forall j \in V \end{aligned} \tag{8}$$

Definition 5.2. A bipartite b-matching imputation (p_U, p_V) is in the **Owen set** if there exists an optimal solution (u, v) for the dual LP 8 and profit of each agent $i \in U$ is $p_U(i) = b_i u_i$ and the profit of each agent $j \in V$ is $p_V(j) = b_j v_j$.

For the b-matching games, the Owen set is always non-empty and a subset of the core, but it does not contain all the imputations in the core, see [5].

5.2 Combinatorial algorithm to compute the leximin Owen set imputation

The leximin Owen set imputation for the b-matching game can be computed using the LP method (see [5]). We can also reduce the problem to finding leximin core imputation of an assignment game (see [5]) albeit in exponential time,. Below, we provide a high-level overview of an efficient combinatorial algorithm adapted from Vazirani [15] for the assignment game. A more detailed description of the algorithm is given in [5]. [15] first divides each edge (i, j) into three groups—*essential*, *viable*, and *subpar*— based on whether the edges are matched $min(b_i, b_j)$ times in all, some or no maximum weight b-matching. The graph $G = (V, E)$ is then restricted to just the essential and viable edges to give graph $H = (V, T_0)$; note that complementarity slackness conditions ensure that these edges are always tight in LP 8. Each connected component of H is labeled as a *unique imputation* component if all vertices receive the same profit share in every Owen set imputation, or a *fundamental* component otherwise. A key feature of the algorithm in [15] is that the profit shares of vertices in a fundamental component can be "rotated" without violating core conditions. In the assignment game, this means profits (dual values) on the U-side can be increased (or decreased) while the corresponding profits on the V-side are decreased (or increased) by the same amount. For the b-matching game, this generalizes: the duals can still be adjusted at the same rate on both sides (with opposite signs), but the profit changes are proportional to the b_i values of each vertex. The following lemma proves that such profit adjustments will still yield a valid Owen set imputation.

Lemma 5.3. † *Let C be any fundamental component in H_0. Then the sum of b_i values of the U vertices is same as the sum of b_i values of the V vertices.*

The algorithm to compute the leximin Owen set imputation for the b-matching game will differ from the assignment game in just one key aspect. An arbitrary imputation is lexicographically improved, so that the profits of all minimum-profit vertices change at an equal rate. The duals must then move at a rate proportional to $1/b_v$ relative to the profits. Consequently, each fundamental component rotates at a corresponding rate. This modification gives us an efficient algorithm to compute the leximin Owen set imputation for the b-matching game.

Theorem 5.4. *The leximin and leximax Owen set imputations of the b-matching game can be computed in time $O(m^{2+o(1)})$.*

[7] Recently, Gangam et al. [6] has proved this by establishing the co-NP-hardness of deciding whether a given imputation is in the core, and the NP-hardness of deciding if a given imputation is the leximin/leximax core imputation for b-matching games.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under grant CCF-2230414. We would like to thank all the anonymous reviewers for their valuable comments.

REFERENCES

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice hall.
[2] Bhaskar Dutta and Debraj Ray. 1989. A concept of egalitarianism under participation constraints. *Econometrica: Journal of the Econometric Society* (1989), 615–635.
[3] Ulrich Faigle, Walter Kern, Sándor P Fekete, and Winfried Hochstättler. 1997. On the complexity of testing membership in the core of min-cost spanning tree games. *International Journal of Game Theory* 26 (1997), 361–366.
[4] Qizhi Fang, Shanfeng Zhu, Maocheng Cai, and Xiaotie Deng. 2002. On computational complexity of membership test in flow games and linear production games. *International Journal of Game Theory* 31 (2002), 39–45.
[5] Rohith R. Gangam, Naveen Garg, Parnian Shahkar, and Vijay V. Vazirani. 2025. Equitable Core Imputations for Max-Flow, MST and b-Matching Games. arXiv:2403.06037 [cs.GT] https://arxiv.org/abs/2403.06037
[6] Rohith Reddy Gangam, Shayan Taherijam, and Vijay V. Vazirani. 2025. On the Core of the b-Matching Game. arXiv:2502.01914 [cs.GT] https://arxiv.org/abs/2502.01914
[7] Hervé Moulin. 2014. *Cooperative microeconomics: a game-theoretic introduction.* Vol. 313. Princeton University Press.
[8] Guillermo Owen. 1975. On the core of linear production games. *Mathematical programming* 9, 1 (1975), 358–370.
[9] Jean-Claude Picard and Maurice Queyranne. 1980. *On the structure of all minimum cuts in a network and applications.* Springer Berlin Heidelberg, Berlin, Heidelberg, 8–16. https://doi.org/10.1007/BFb0120902
[10] A. Schrijver. 1986. *Theory of Linear and Integer Programming.* John Wiley & Sons, New York, NY.

[11] Lloyd S Shapley and Martin Shubik. 1971. The assignment game I: The core. *International Journal of Game Theory* 1, 1 (1971), 111–130.
[12] Joaquin Sánchez Soriano, Marco A López, and Ignacio Garcia Jurado. 2001. On the core of transportation games. *Mathematical Social Sciences* 41, 2 (2001), 215–225.
[13] JRG Van Gellekom, Jos AM Potters, JH Reijnierse, MC Engel, and SH Tijs. 2000. Characterization of the Owen set of linear production processes. *Games and Economic Behavior* 32, 1 (2000), 139–156.
[14] Vijay V Vazirani. 2022. The General Graph Matching Game: Approximate Core. *Games and Economic Behavior* 132 (2022).
[15] Vijay V Vazirani. 2024. Equitable Core Imputations via a New Adaptation of The Primal-Dual Framework. *arXiv preprint arXiv:2402.11437* (2024).
[16] Gary R Waissi. 1994. Network flows: Theory, algorithms, and applications.
[17] Wikipedia. [n.d.]. Streetlight effect. https://en.wikipedia.org/wiki/Streetlight_effect.