Allocating Indivisible Goods

IVONA BEZÁKOVÁ and VARSHA DANI
University of Chicago

The problem of allocating divisible goods has enjoyed a lot of attention in both mathematics (e.g. the cake-cutting problem) and economics (e.g. market equilibria). On the other hand, the natural requirement of indivisible goods has been somewhat neglected, perhaps because of its more complicated nature. In this work we study a fairness criterion, called the Max-Min Fairness problem, for k players who want to allocate among themselves m indivisible goods. Each player has a specified valuation function on the subsets of the goods and the goal is to split the goods between the players so as to maximize the minimum valuation. Viewing the problem from a game theoretic perspective, we show that for two players and additive valuations the expected minimum of the (randomized) cut-and-choose mechanism is a 1/2-approximation of the optimum. To complement this result we show that no truthful mechanism can compute the exact optimum.

We also consider the algorithmic perspective when the (true) additive valuation functions are part of the input. We present a simple $1/(m-k+1)$ approximation algorithm which allocates to every player at least $1/k$ fraction of the value of all but the $k-1$ heaviest items. We also give an algorithm with additive error against the fractional optimum bounded by the value of the largest item. The two approximation algorithms are incomparable in the sense that there exist instances when one outperforms the other.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.1 [**Discrete Mathematics**]: Combinatorics – *Combinatorial Algorithms*; G.2.2 [**Discrete Mathematics**]: Graph Theory – *Graph Algorithms*

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Approximation Algorithms, Cake-cutting, Fairness, Game Theory, Truthfulness

1. INTRODUCTION

The need for fair division of a set of objects among several parties emerges naturally in many real-life scenarios, ranging from inheritance and divorce settlements to border disputes. Naturally, the political science and economic literature addresses the issues of fair division under various assumptions and objectives (see, e.g., Brams and Taylor [Brams and Taylor 1996]). But few of the methods view the problem from a computational perspective and most assume the existence of several divisible objects or the possibility of a monetary compensation in order to achieve a balanced division. In certain situations such as some inheritance disputes the items are often indivisible, of different value to each of the parties, and the parties might not be able to compensate the other players financially.

General Max-Min Fairness Problem. There are k players and a set A of m indivisible objects. Player i has a non-negative valuation function $v_i : 2^A \to \mathbf{R}_0^+$ which is normalized to $v_i(A) = 1$. The goal is to find a partition $A_1, \ldots, A_k$ of the goods which maximizes $\min_i v_i(A_i)$.

The inspiration for the Max-Min Fairness problem comes from its divisible counterpart: the cake-cutting problem. Initiated by the Polish school of mathematicians in the 1950's, in the cake-cutting problem k parties attempt to divide a cake between them in a fair manner. Generally, one of two objectives comes under consideration: envy-freeness or fair-fractions. A division is *envy-free* if every player thinks she got at least as much as any other player. A division is divided into *fair fractions*, if each person gets at least $1/k$ of the whole cake (from her perspective). The cake-cutting represents a problem from the class of divisible fair allocation problems and several solutions for both objective functions are well-known (see, e.g., Robertson and Webb [Robertson and Webb 1998]). Perhaps the most interesting aspect of the cake-cutting solutions is their truthfulness; in other words, players are given guarantees if they do not lie about their valuation functions.

Lipton, Markakis, Mossel and Saberi [Lipton et al. 2004] studied the indivisible variant of the envy-free cake-cutting problem. Under the assumption that players reveal their true valuation functions, Lipton *et al.* presented a simple polynomial-time algorithm producing an allocating with an absolute bound on the envy. Their envy is guaranteed to be at most the maximal marginal utility defined as the largest value by which a value of a set increases after enlarging it by exactly one element. Their algorithm works for (essentially) unrestricted classes of valuation functions, with the caveat that the true valuation functions are known to the algorithm. It is not known whether there is a reasonable mechanism when the players can lie about their valuations in order to obtain a larger portion.

The Max-Min Fairness problem is an indivisible analogue of the fair-fractions cake-cutting problem, and it has been proposed by Lipton *et al.* as an open direction. We note that this problem and the envy-minimization problem are of different nature as there exist instances when the minimum value obtained by the envy-minimizing splits is arbitrarily worse than the minimum in the max-min allocations. Similarly, the envy of the max-min allocations might be much bigger than the envy in the envy-minimizing splits.

We consider two different input settings. In the "*known valuations*" setting the algorithm is given an oracle access to the v_i's. In the opposite "*unknown valuations*" setting, the algorithm asks the players directly for the v_i values. This setting is conceptually more difficult, since the players might opt to lie in the hope of achieving a better split for themselves. We assume that the players do not know anything about the other players' valuations.

A general valuation function assigns a real number to every subset of A, and one may need to specify 2^m numbers to describe it. Therefore we will work with two restricted but natural classes of valuations. We call a valuation *additive* if the value of a set equals the sum of the values of the individual items. A valuation is *maximal* if the value of a set of items equals the value of the most expensive item in the set. This situation corresponds to a set of items of similar functionality such as finding a full-time job. Notice that the above valuations are induced by m real numbers, the values of single items from A.

In the case of known additive valuations, the Max-Min Fairness problem is a special case of a scheduling problem where the completion time of any single machine is maximized. Woeginger [Woeginger 1997] and Epstein and Sgall [Epstein and Sgall 2004] presented polynomial-time approximation schemes for uniformly related machines. These results immediately translate to instances of the Max-Min Fairness problem where all the players have equal valuations. Woeginger [Woeginger 2000] gave a pseudo-polynomial algorithm for constant number of machines (players).

1.1 Our Results

Mechanism design. In the unknown valuations setting, the mechanism needs to elicit relevant information from the players who may play deceitfully. Our main result is a randomized allocation procedure for two players, which achieves at least half the optimum for both players in expectation. This result might not sound surprising. A trivial algorithm gives all the items to a single player, determining the recipient by an unbiased coin flip. Every player expects 1/2 value of the goods, so the minimum of the expected values is 1/2. However, the expected minimum is 0, since the other player does not get anything. We want to avoid this situation.

We say that a randomized algorithm for the Max-Min Fairness problem c-approximates the optimum, if the algorithm guarantees that the expected minimum (as opposed to the minimum of the expected values) is within a c factor of the deterministic optimum. Our algorithm, a randomized cut-and-choose mechanism, is an expected 1/2-approximation, provided that the players play rationally. However, full rationality requires the cutter to solve an NP-complete problem: finding the optimal cut. Fortunately this is not necessary. We show that it suffices to be rational in a limited sense: the cutter needs to find a locally optimal cut for which we provide a polynomial-time algorithm. This is the first truthful mechanism with provable guarantees for a problem of fair allocation of indivisible goods.

Omniscient algorithms. We present a simple polynomial matching-based algorithm for known maximal valuations. Building on this algorithm we obtain a $1/(m-k+1)$ approximation of the optimal allocation for known additive utilities. Iteration of a variant of this algorithm guarantees a $1/k$ fraction of all but the $k-1$ heaviest items to every player. More precisely, every player gets a bundle that is at least as good as getting every k-th item, where the items are sorted decreasingly by their individual values.

We also present an algorithm based on an integer rounding of the corresponding linear programming relaxation, which allocates to every player the fractional optimum value minus the heaviest item. The rounding technique is inspired by the work of Lenstra, Shmoys and Tardos [Lenstra et al. 1990] who considered the makespan-minimization problem when scheduling a set of jobs on unrelated machines. Our contribution consists of finding a special combinatorial structure, described in Lemma 3.3.

We note that the two algorithms we exhibit for the additive valuations are complementary, in the sense that each outperforms the other on some inputs.

Our algorithms are not only polynomial time but are efficiently implementable. All our theorems (including the one related to unknown valuations) are tight in the sense that the analysis of the algorithms which the theorems refer to cannot be improved. An improvement could come only from more cleverly designed algorithms.

To complement our results, we note that no ρ approximation algorithm exists for $\rho > 1/2$ unless $P = NP$. The proof of this statement follows from a reduction from 3-Matchings, using an idea similar to that of Lenstra *et al.* [Lenstra et al. 1990]. A linear factor approximation algorithm and a $1/2 + \varepsilon$ hardness result were obtained independently by Markakis and Saberi [Markakis and Saberi 2004].

Due to space constraints we defer all proofs to the full version of the paper.

1.2 Related Economic Work

In the economic and political sciences literature (see, e.g., Brams and Taylor [Brams and Taylor 1996]) the goal differs somewhat from ours: the quality of an allocation is measured in terms of equitability (did the players get the same utility?), envy-freeness, and efficiency (no other allocation gives a larger utility to all players simultaneously). Probably the most well-known method is the so-called adjusted winner procedure. It uses a monetary rebalancing in order to satisfy the equitability, envy-freeness and efficiency conditions simultaneously.

Moulin [Moulin 2002] considered a related problem where m *identical* indivisible objects need to be allocated to k parties, each of which specifies a request for a certain number of objects. He gave a proportional allocation method and proved that it is the only fair method (in the sense of "Equal Treatment of Equals", i. e., two agents with identical claims should receive the same random allocation) satisfying other desirable structural invariance properties.

2. KNOWN MAXIMAL VALUATIONS

Valuation functions induced by values $v_i(j)$ for $i \in [k]$, $j \in [m]$ can be expressed conveniently as a complete weighted bipartite graph which we refer to as the *valuations graph*. One partition represents the set of players and the other the set of goods where the weight of the edge (i, j) is $v_i(j)$. If the valuations are of maximal type, then the Max-Min Fairness problem is equivalent to the following matching problem applied to the valuations graph with the set of goods A and the set of players B. The problem can be solved in polynomial time via a reduction to the Maximum Matching problem.

Max-Min Matching Problem. Input: A bipartite graph $G = (A \cup B, E)$ and edge-weights $w : E \to \mathbf{R}_{\geq 0}$. Goal: A matching $M \subseteq E$ such that (1) M covers B, i.e. for every $b \in B$ there exists a s.t. $(a, b) \in M$, and (2) the minimum weight edge in M is as large as possible, i. e., for every M' covering B, $\min_{e \in M} w(e) \geq \min_{e \in M'} w(e)$.

3. KNOWN ADDITIVE VALUATIONS

In the case of known additive valuations, finding a fairness maximizing allocation is NP-complete since the SUBSET SUM problem (see, e.g., Garey and Johnson [Garey and Johnson 1979]) is a special case for 2 players and identical valuations, i. e., $v_1 \equiv v_2$. We present an approximation guarantee of $1/(m - k + 1)$ where m is the number of items and k the number of players. Moreover every player gets at least the value of every k-th item if items are sorted decreasingly by this player's valuations. We analyze an LP based algorithm with an additive guarantee against the *fractional* optimum. Our algorithm gives a utility of at least the fractional optimum minus the player's largest item to every player. These two algorithms are incomparable, in the sense that neither performs consistently better than the other.

3.1 An approach via Matching

First we state a generalization of the Max-Min Matching problem which can also be solved in polynomial time.

Generalized Max-Min Matching Problem. Input: A bipartite graph $G = (A \cup B, E)$, edge-weights $w : E \to \mathbf{R}_{\geq 0}$, and vertex values $v(b) \in \mathbf{R}^+$ for every $b \in B$. Goal: A matching $M \subseteq E$ such that (1) M covers B, i.e. for every $b \in B$ there exists a s.t. $(a, b) \in M$, and (2) minimum weight edge in M, together with the value of the adjacent vertex, is as large as possible, i. e., for every M' covering A, $\min_{(a,b)\in M} w(a, b) + v(b) \geq \min_{(a,b)\in M'} w(a, b) + v(b)$.

Applying the Generalized Max-Min Matching problem iteratively we get:

Theorem 3.1. *Let $x_{i,1}, \ldots, x_{i,m}$ be a decreasingly sorted sequence of goods according to the i-th player's valuation, i. e., $v_i(x_{i,j}) \geq v_i(x_{i,j+1})$ for every j. There exists a polynomial-time algorithm which produces a partition of the goods such that each player i is guaranteed a value of at least $\sum_\ell v_i(x_{i,k\ell}) \geq (1 - \sum_{j=1}^{k-1} v_i(x_{i,j}))/k$. Moreover, the algorithm has a $1/(m - k + 1)$ approximation guarantee, where m is the number of goods and k is the number of players.*

3.2 Linear Programming Relaxation

Another approach to solving the Max-Min Fairness problem is to formulate it as an integer program. Let $x_{i,j}$ denote the indicator variable for "player i gets item j". Then $x_{i,j} \in \{0, 1\}$. The constraints on the program are that each item must be allocated exactly once and that each player's utility must be at least as much as the objective function. Thus the optimal allocation for the Max-Min Fairness problem instance is the solution to the integer program:

$$\text{Maximize } \omega \text{ subject to: } x_{i,j} \in \{0, 1\},\ \forall j : \sum_i x_{i,j} = 1 \text{ and } \forall i : \omega \leq \sum_j v_i(j)\ x_{i,j}$$

We consider the linear programming relaxation that allows fractional allocations, or in other words, arbitrarily divisible goods.

$$\text{Maximize } \omega \text{ subject to: } 0 \leq x_{i,j} \leq 1,\ \forall j : \sum_i x_{i,j} = 1 \text{ and } \forall i : \omega \leq \sum_j v_i(j)\ x_{i,j}$$

Note that the optimal value of ω for the linear program, henceforth denoted $FOPT$, is at least $1/k$ where k is the number of players. Moreover the optimal (fractional) allocation satisfies the property that all players in a connected component of the valuations graph receive the same total value.

Given a feasible solution $\mathbf{x} = (x_{i,j})_{1\leq i\leq k, 1\leq j\leq m}$ to the above linear program, let $\mathbf{x_i} = (x_{i,j})_{1\leq j\leq m}$ denote the items allocated to player i. By abuse of notation, we'll use $v_i(\mathbf{x_i})$ to denote $\sum_j v_i(j)\ x_{i,j}$, the utility to player i from this allocation.

Theorem 3.2. *Let $\hat{\mathbf{x}}$ be an optimal (fractional) solution of the above program, with $FOPT_i := v_i(\hat{\mathbf{x}}_i)$. Then there exists an integer solution $\mathbf{y}$ such that $v_i(\mathbf{y}_i) \geq \max(0, FOPT_i - \max_j v_i(j))$. Thus, $\min_i v_i(\mathbf{y}_i) \geq \max(0, FOPT - \max_{i,j} v_i(j))$. Moreover, $\mathbf{y}$ can be found in polynomial time.*

We credit part of the proof of the above theorem to Lenstra, Shmoys and Tardos [Lenstra et al. 1990] who used a similar theorem in a different scheduling context. Their rounding method is based on the insight that the graph underlying the fractional solution is a bipartite pseudoforest, which may then be rounded to a matching. We use the same framework for the first part of our rounding scheme. However, since our problem is of an opposite nature (a maximization rather than a minimization), we need to round the pseudoforest edges to a structure which has a matching-like behavior for one partition of the underlying bipartite graph while in the other partition it behaves like an "anti-matching", as summarized in the following lemma.

LEMMA 3.3. *Let G be a bipartite pseudotree, i. e., a connected graph with at most as many edges as vertices. Let A and B be the vertex sets of G and let E be the edge set. Then there is a set $S \subset E$ such that for all $j \in A$, S covers j by exactly one edge, and for every $i \in B$, S covers i by at least $\deg(i) - 1$ edges. This set can be found in polynomial time.*

4. UNKNOWN ADDITIVE VALUATIONS

We now consider the case where the algorithm does not know the players' true valuations as part of its input but must instead elicit information about them from the players. Thus the task of finding an optimal allocation becomes a mechanism design problem, as the players may try to cheat to their own advantage. Ideally, one would like a mechanism which forces all the players to be truthful (by arranging the payoffs so that revealing their true valuation is the players' best strategy). Unfortunately, as the next example shows, no mechanism (deterministic or randomized) for the Max-Min fair allocation problem can be truthful.

EXAMPLE 4.1. *Suppose $\mathcal{M}$ is a mechanism to compute a Max-Min fair allocation. Consider two players and three items: Alice has valuation $(\frac{2}{3}, \frac{1}{3}, 0)$ while Bob has valuation $(0, \frac{1}{2}, \frac{1}{2})$. $\mathcal{M}$ will allocate item 1 to Alice and the rest to Bob, giving Alice a utility of $\frac{2}{3}$. However by declaring her valuation as $(\frac{1}{3}, \frac{2}{3}, 0)$ Alice can trick the mechanism into giving Bob only item 3, and increasing her utility to 1.*

In light of this, we would like to examine what can be accomplished by mechanisms which elicit only partial information about the valuations from the players. In the spirit of the classical cake cutting problem of Banach, Knaster, and Steinhaus [Steinhaus 1948] we would like to have a mechanism that gives each player a guarantee based on the information received from the player, such that deceitful play always results in a worse guarantee to the player, even if not a worse payoff in every instance. Also, we would like deceitful play by one player to affect the guarantees of other players as little as possible. The guarantees provided by the mechanism should be as nice as possible, in terms of the following definition.

Definition 4.2. Let OPT denote the optimum value. We say that a randomized algorithm is an *expected c-approximation* of the optimum if $E(\min_i v_i(A_i)) \geq cOPT$.

As a first attempt, we consider mechanisms that get no information from the players, giving them no opportunity to cheat. Consider the mechanism that simply assigns all the goods to a single player chosen uniformly at random. Then each player's expected utility is $1/k$ and hence the minimum expected utility is $1/k$. However the mechanism always gives nothing to all but one player. Thus the actual minimum utility (and hence also the expected minimum utility) is always 0, which is rather unsatisfactory.

The mechanism that allocates the items by a separate k-way lottery for each item also gives each player an expected utility of $1/k$ but it too fails to give a good guarantee on the expected minimum utility, as the next example shows.

EXAMPLE 4.3. *Consider k players and k items. Suppose for each i, $1 \leq i \leq k$, player i's valuation is given by $v_i(i) = 1$, $v_i(j) = 0$ for all $j \neq i$. The optimal allocation is to assign each item to the unique player desiring it, so that everyone has utility 1. The randomized mechanism described above achieves this allocation $1/k^k$ of the time. The rest of the time it gives at least one item to a "wrong" player, so that the minimum utility is 0. Thus, the expected minimum utility is $1/k^k$.*

Thus, unless the players provide some information about their valuations, no reasonable guarantee can be made for the expected minimum utility.

In Section 4.1 we present a randomized mechanism for two players which with some assumptions on the players' strategies, gives an expected 1/2-approximation to the optimal allocation, while simultaneously guaranteeing each player at least 1/2 of what they would have got in the optimal allocation. How to extend this to more than two players is as yet unclear.

4.1 Two Players: Cut-and-Choose

In this section we investigate the cut-and-choose mechanism for two players. In the basic cut-and-choose mechanism player 1 divides the goods into two parts, and player 2 chooses one of the parts. Player 1 then receives the other part. Even in the case of divisible goods, player 2 may have an advantage in this game. We'll consider the more symmetric version in which the first player ("cutter") is chosen by a fair coin flip. (We do not assume that the players have the same valuations.)

We take the standpoint that each player wishes to optimize her worst case outcome against adversarial play by the opponent. Thus the cutter's goal is to divide the goods as equally as possible. However, the cutter's optimization problem is NP-hard by reduction from SUBSET SUM. With this in mind we introduce the following notion of local optimality.

Definition 4.4. *We will call a partition (S, T) of the goods* locally optimal *for valuation v if moving a single item from either side to the other does not decrease the disparity of the partition, $|v(S) - v(T)|$. Moreover we require that whenever the disparity is positive, all items of zero value are on the smaller side.*

We give a greedy algorithm which, given any valuation v, computes a locally optimal partition in time $O(m \log m)$. The algorithm iteratively assigns the most valuable remaining good to the less valuable of S and T.

Our main result here is that, assuming both players produce locally optimal partitions when cutting and select the larger piece when choosing the randomized cut-and-choose mechanism produces an allocation whose expected minimum utility is at least half the maximum minimum utility. Moreover, it guarantees each player at least half what they would have received in a true optimal solution. These guarantees are preserved if the players are truly rational (computationally unconstrained) as the optimal partition *does* satisfy the local optimality conditions.

THEOREM 4.5. *Let players P_1 and P_2 have additive valuation functions v_1 and v_2. Let A_1, A_2 be the optimal allocation for these valuations, i. e., the allocation maximizing $\min_i v_i(A_i)$ and let $OPT = \min_i v_i(A_i)$. Then, assuming both players produce locally optimal partitions, the randomized cut and choose mechanism produces an allocation S_1, S_2 such that*

$$\mathbf{E}\left(\min_i v_i(S_i)\right) \geq \frac{1}{2}OPT \quad \text{and for each } i \quad \mathbf{E}\left(v_i(S_i)\right) \geq \frac{1}{2}v_i(A_i)$$

5. OPEN PROBLEMS

There are many intriguing open questions related to the Max-Min Fairness problem. In the case of known additive valuations, can we use randomized rounding to give a better-than-linear approximation guarantee? Can we get a better non-approximability factor using Probabilistically Checkable Proofs [Arora et al. 1998]?

In the case of unknown additive utilities, can we obtain an improved approximation factor for two players? How can one generalize the two player mechanism for three or more players?

ACKNOWLEDGMENTS

We are indebted to Lance Fortnow for guiding and encouraging us throughout this work. Many thanks go to Tom Hayes and Daniel Štefankovič for helpful discussions. We are also grateful to Pedro Felzenszwalb, Elchanan Mossel, Amin Saberi, Gerhard Woeginger and anonymous referees for useful comments and pointers to the literature.

REFERENCES

ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. 1998. Proof verification and the hardness of approximation problems. *J. ACM 45,* 3, 501–555.

BRAMS, S. J. AND TAYLOR, A. D. 1996. *Fair Division : From Cake-Cutting to Dispute Resolution.* Cambridge University Press.

EPSTEIN, L. AND SGALL, J. 2004. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica 39,* 1, 43–57.

GAREY, M. AND JOHNSON, D. 1979. *Computers and Intractability.* Freeman and Co., New York.

LENSTRA, J. K., SHMOYS, D. B., AND TARDOS, É. 1990. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program. 46*, 259–271.

LIPTON, R. J., MARKAKIS, E., MOSSEL, E., AND SABERI, A. 2004. On approximately fair allocations of indivisible goods. In *ACM Conference on Electronic Commerce*. 125–131.

MARKAKIS, E. AND SABERI, A. 2004. Personal communication.

MOULIN, H. 2002. The proportional random allocation of indivisible units. *Social Choice and Welfare 19,* 2, 381–413.

ROBERTSON, J. AND WEBB, W. 1998. *Cake Cutting Algorithms: Be Fair If You Can.* A. K. Peters, Ltd.

STEINHAUS, H. 1948. The problem of fair division. *Econometrica 16,* 1 (January), 101–104.

WOEGINGER, G. J. 1997. A polynomial time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters 20*, 149–154.

WOEGINGER, G. J. 2000. When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing 12*, 57–75.