On Approximately Fair Allocations of Indivisible Goods[*]

Richard Lipton[†] Evangelos Markakis[‡] Elchanan Mossel[§] Amin Saberi[¶]

Abstract

We study the problem of fairly allocating a set of indivisible goods to a set of people from an algorithmic perspective. Fair division has been a central topic in the economic literature and several concepts of fairness have been suggested. The criterion that we focus on is the maximum envy between any pair of players. An allocation is called envy-free if every player prefers her own share than the share of any other player. When the goods are divisible or when there is sufficient amount of one divisible good, envy-free allocations always exist. In the presence of indivisibilities however this is not the case. We first show that when all goods are indivisible, there always exist allocations in which the envy is bounded by the maximum marginal utility and we present a simple polynomial time algorithm for computing such allocations. We further show that our algorithm can be applied to the continuous cake-cutting model as well and obtain a procedure that produces ϵ-envy-free allocations with a linear number of cuts. We then look at the optimization problem of finding an allocation with minimum possible envy. In the general case, there is no polynomial time algorithm (or even approximation algorithm) for the problem, unless **P** = **NP**. We consider natural special cases (e.g. additive utilities) which are closely related to a class of job scheduling problems. Polynomial time approximation algorithms as well as inapproximability results are obtained. Finally we investigate the problem of designing truthful mechanisms for producing allocations with bounded envy.

1 Introduction

Fair division has been a central problem in economic theory. The first attempt for a mathematical treatment of the problem was made by the Polish school of mathematicians (Steinhaus, Banach, and Knaster [27]) and was the source of many interesting questions. Over the last decades, a vast literature has developed [7, 26] and several notions of fairness have been suggested.

In this work, our focus will be on the concept of envy-freeness. An allocation is *envy-free* if and only if every player likes his own share at least as much as the share of any other player. The class of envy-free allocations as a fairness concept was introduced by Foley [16] and Varian [31] and has been studied extensively since then in the economic literature [26, 7]. However, in most of the models considered so far, either all goods are infinitely divisible or there is at least

[*] A preliminary version of this work has appeared in [22].

[†] College of Computing, Georgia Tech and Telcordia Research. Email: rjl@cc.gatech.edu. Supported by NSF grant CCR-0002299.

[‡] Corresponding author. Center for Math and Computer Science (CWI), Amsterdam. Email: vangelis@cwi.nl.

[§] Department of Statistics, U.C. Berkeley. Email: mossel@stat.berkeley.edu.

[¶] Department of Management Science and Engineering, Stanford University. Email: saberi@stanford.edu.

one divisible good such as “money” to let the players compensate each other in order to achieve envy-freeness [1, 29]. Under these assumptions, it has been proved that envy-free allocations always exist (see among others [28, 10, 1, 29, 3]).

As we believe that issues of indivisibility should be taken into consideration, we study the problem of allocating indivisible goods in a fair manner. When all goods are indivisible, an envy-free allocation may not exist and we wish to find allocations in which the envy is bounded or more preferably allocations that achieve the minimum possible envy. We will study this problem from a computational perspective and we will be interested in finding polynomial time algorithms for producing such allocations. The methodology of algorithms and complexity has been successfully used in the last few years for studying many other game theoretic solution concepts and analyzing games on discrete structures, see e.g. [23].

We consider the following model in our presentation: for each player p there is a utility function $v_p(\cdot)$ so that for each subset S, $v_p(S)$ is the value of player p for this subset. Given an allocation of the goods to the players, a player p envies player q if her valuation for the bundle given to player q is more than her valuation for her own bundle. In that case, her *envy* is the difference.

Our Results: In Section 2 we show that there always exists an allocation with maximum envy at most α, where α is the maximum marginal utility of the goods. The maximum marginal utility is the maximum value by which the utility of a player is increased when one more good is added to her bundle. Furthermore, we give a polynomial time algorithm for producing an allocation with maximum envy at most α. The problem of finding allocations with bounded envy in the presence of indivisible goods was introduced in [10]. A bound of $O(\alpha n^{3/2})$ was obtained, where n is the number of players. Our bound is a substantial improvement and it is also tight.

We then exhibit an application of our algorithm in the continuous model where the cake is the interval $[0, 1]$ and the preferences are given by probability measures. We show that our algorithm can be used to obtain a procedure for obtaining an ϵ-envy-free allocation with $O(n)$ cuts for any constant $\epsilon > 0$.

In Section 3 we look at the optimization problem of computing allocations with minimum possible envy. We show that in most cases the problem is hard. First, using a similar argument as in [24], we show that any algorithm needs exponential time to achieve any approximation to the minimum envy allocation. We then look at the special case of additive utilities, i.e., $v_p(S) = \sum_{i \in S} v_p(\{i\})$. Even in this case we prove that for any constant c, there can be no 2^{m^c}-approximation algorithm for the minimum envy problem unless $\mathbf{P} = \mathbf{NP}$, where m is the number of goods.

We believe that a more suitable objective function is the maximum envy-ratio. The envy-ratio of player p for player q is the utility of p for q’s bundle over her utility for her own bundle. If all players have the same utility function, the problem is closely related to a class of scheduling problems on identical processors. If we think of the players as identical machines and the set of goods as a set of jobs, then our problem is equivalent to scheduling the jobs so as to minimize the ratio of the maximum completion time over the minimum completion time. In [9] it is shown (in the context of job scheduling) that Graham’s greedy algorithm [17] achieves an approximation factor of 1.4 for the envy-ratio problem. We improve this result and derive a polynomial time approximation scheme, i.e., a $(1 + \epsilon)$-approximation for any constant $\epsilon > 0$ (see Section 3 for the relevant definitions).

Finally in Section 4 the issue of incentive compatibility is addressed and we conclude in the last section with many interesting open problems.

2 Existence of Allocations with Bounded Maximum Envy

Let $N = \{1, 2, ..., n\}$ be a set of players and $M = \{1, 2, ..., m\}$ be a set of indivisible goods. A utility function v_p is associated with each player p. For $S \subseteq M$, $v_p(S)$ is the happiness player p derives if she obtains the subset S. We assume that v_p is non-negative and monotone i.e. $v_p(S) \leq v_p(T)$ for every $S \subseteq T$ and every p.

An allocation A is a partition of the goods $A = (A_1, A_2, ..., A_n)$ where $\cup_{p=1}^{n} A_p = M$ and $A_p \cap A_q = \emptyset$ for all $p \neq q$. A_p is the subset allocated to player p. Note that some of the sets A_p may be empty. A partial allocation will be a partition of some subset of M.

Given an allocation $A = (A_1, A_2, \ldots, A_n)$, we say that player p envies player q if she prefers the bundle allocated to q to her own i.e. $v_p(A_p) < v_p(A_q)$. We will denote by e_{pq} the envy of p for q:

$$e_{pq}(A) = \max\{0, v_p(A_q) - v_p(A_p)\}.$$

We define $e(A)$ to be the maximum envy between any pair of players.

$$e(A) = \max\{e_{pq}(A),\ p, q \in N\}.$$

We will often omit the parameter A in the notation.

A natural question is whether there exist allocations with bounded envy. We obtain a bound on the envy in terms of the maximum marginal utility of the goods, α. The marginal utility of a good i with respect to a player p and a subset of goods S, is the amount by which it increases the utility of p, when added to S, i.e., $v_p(S \cup \{i\}) - v_p(S)$. The maximum marginal utility is:

$$\alpha = \max_{S,p,i}\ v_p(S \cup \{i\}) - v_p(S)$$

In addition to proving a bound on the envy, we present an efficient algorithm that computes a desired allocation. For that, we assume that the algorithm can have oracle access to the players' utilities, i.e., it can ask, whenever needed, for the utility of a player p for any subset S.

Theorem 1 *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, there is an $O(mn^3)$ time algorithm for finding such an allocation.*

Before going into the proof, we introduce some terminology that we will use. Given an allocation A, we define the *envy graph* of A as follows: every node of the graph represents a player and there is a directed edge from p to q iff p envies q. The proof of Theorem 1 is based on the following Lemma:

Lemma 2 *For any partial allocation A with envy graph G, we can find another partial allocation B of the same subset of goods with envy graph H such that:*

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

Proof :

If G has no directed cycles, we are done. Suppose that $C = p_1 \to p_2 \to \cdots \to p_r \to p_1$ is a directed cycle in G. If $A = \{A_1, ..., A_n\}$, we can obtain $A' = (A'_1, ...A'_n)$ by re-allocating the goods as follows: $A'_p = A_p$ for all $p \notin \{p_1, \ldots, p_r\}$, and $A'_{p_1} = A_{p_2}, A'_{p_2} = A_{p_3}, \ldots, A'_{p_r} = A_{p_1}$.

Note that all players evaluate what they have in A' at least as much as what they have in A. Note also that A' uses the same bundles of goods as A, it only assigns different owners to them. Therefore it is easy to see that $e(A') \leq e(A)$.

We can also show that the number of edges in the envy graph G' corresponding to A' has decreased. To see this, first note that the set of the edges between pairs of vertices in $N \setminus C$ has not changed. Also every edge of the form $p \to p_j$ for $p \in N \setminus C$ and $p_j \in C$ has now become the edge $p \to p_{j-1}$ (or $p \to p_r$ if $j = 1$) in G' and no more edges of this form have been added. The number of edges of the form $p_j \to p$ has either decreased or remained the same since players in C are strictly happier. Finally for $p_i \in C$ the number of edges from p_i to vertices in C has decreased by at least 1.

Thus by repeatedly finding and removing cycles using the above procedure, we will obtain an allocation B with corresponding envy graph H such that $e(B) \leq e(A)$. Since the number of edges decreases at every step, the process will terminate and at termination the graph H will be acyclic. □

Proof of Theorem 1 : We give an algorithm that produces the desired allocation. The algorithm proceeds in m rounds. At each round one more good is allocated to some player.

In the first round, we allocate good 1 to some player arbitrarily. Clearly the maximum envy is at most α. Suppose at the end of round i, the goods $\{1, ..., i\}$ have been allocated to the players and the maximum envy is at most α. At round $i+1$, we construct the envy graph corresponding to the current allocation. We use the procedure of Lemma 2 to obtain an allocation A in which the maximum envy is at most α and the new envy graph G is acyclic. Since G is acyclic, there is a player $p \in N$ with in-degree 0, which implies that nobody envies p. We then allocate good $i + 1$ to p. Let $B = (B_1, ..., B_n)$ be the new allocation. For any 2 players q, r with $q, r \neq p$, $e_{qr}(B) = e_{qr}(A) \leq \alpha$. For $q \in N \setminus \{p\}$, since $e_{qp}(A) = 0$ we have:

$$\begin{aligned} e_{qp}(B) &= \max\{0, v_q(A_p \cup \{i\}) - v_q(A_q)\} \\ &\leq \max\{0, \alpha + v_q(A_p) - v_q(A_q)\} \leq \alpha \end{aligned}$$

We use a simple amortized analysis for the running time of the algorithm. In Lemma 2, we keep removing cycles until the envy-graph is acyclic. Finding a cycle and removing it takes at most $O(n^2)$ time and it decreases the number of edges by at least one. Initially the envy graph has no edges. Allocating a good at any round adds at most n edges to the new envy graph. Since every cycle removal decreases the number of edges, the number of times we have to remove a cycle is at most $O(nm)$ and the total running time is $O(mn^3)$. □

In the case of additive utilities, [10] consider a more general model, where the object to be partitioned is a measurable space $(\Omega, \mathcal{F})$ (think as an example that $(\Omega, \mathcal{F}) = ([0, 1], \text{Borel sets})$).

Formally each player has a utility function which is a probability measure v_p (hence additive) on $(\Omega, \mathcal{F})$ such that for each v_p the maximum value of an *atom* is α. A subset $S \subseteq \Omega$ is an atom for v_p if $v_p(S) > 0$ and $\forall E \subset S$, either $v_p(E) = 0$ or $v_p(E) = v_p(S)$. The case of indivisible goods corresponds to the case where there are m atoms and the measures are such that they assign positive value only to the atoms. In that case, the marginal utility of any good is clearly the maximum value of an atom, which is α. It was shown in [10] that there exist allocations with envy at most $O(\alpha n^{3/2})$.

We can prove that our result also holds in this model and hence it improves the bound of $O(\alpha n^{3/2})$ to α. The proof is based on the observation that we can reduce this model to the indivisible goods setting by partitioning Ω into indivisible goods of value at most α and then applying Theorem 1.

Theorem 3 *When the utilities of the players are probability measures on $(\Omega, \mathcal{F})$ with atoms of value at most α, there exists a measurable partition $A = (A_1, ..., A_n)$ of Ω such that $e(A) \leq \alpha$.*

Proof : The case $\alpha = 0$ corresponds to an infinitely divisible cake and an envy-free allocation always exists [10]. For $\alpha > 0$ we can reduce the problem to allocating indivisible goods of value at most α through the following Lemma:

Lemma 4 *The space Ω can be partitioned in m disjoint sets $S_1, ..., S_m$ such that $m \leq 2n/\alpha$ and $v_p(S_j) \leq \alpha$ for every $p = 1, ..., n, j = 1, ..., m$*

Lemma 4 is a direct consequence of the arguments used in the proof of Lemma 2.2 of [18]. In particular, it follows from [18] that for a single probability measure v_p with atoms of value at most α, we can partition Ω into k disjoint sets $S_1, ..., S_k$ such that $k \leq 2/\alpha$ and $v_p(S_i)) \leq \alpha$. Hence for n players, we will have at most $2n/\alpha$ sets each of which will be of value at most α for every player.

We can now treat the intervals $S_1, ..., S_m$ produced by the Lemma as indivisible goods and Theorem 1 completes the proof. $\square$

2.1 An Application to the Continuous Model

In the most common formulation of cake cutting problems in the literature the cake Ω is the interval $[0, 1]$ and the preferences are given by atomless probability measures. The measures $v_p, p = 1, ..., n$ according to the standard assumptions are defined on Lebesgue measurable sets and they are divisible, i.e., for any $X \subseteq \Omega$ and any $\lambda \in [0, 1]$, there is $X' \subseteq X$ such that $v_p(X') = \lambda v_p(X)$. In this continuous model, we know that envy-free partitions always exist [28] and an interesting question is to come up with algorithms that achieve fair solutions with a minimal number of cuts. For more information on the variants of this problem we refer the reader to [7, 26].

For the weaker concept of *proportional fairness*, there has been a series of works [14, 15, 25, 34] where upper and lower bounds have been obtained on the number of required cuts both for proportional and approximately proportional fairness. To obtain an upper bound, one needs to design an algorithm and to do so, we first need to define formally what are the allowable operations that the algorithm can perform since we are now dealing with a continuous object. In the literature, there are usually two types of operations that are considered, namely cut and evaluation operations. Below we define these operations following the notation of Edmonds and Pruhs [14]:

> $Cut(p; x_1, \lambda)$: With this operation the algorithm asks player p to make a cut at a point $x_2 \geq x_1$ such that $v_p([x_1, x_2]) = \lambda$.
>
> $Eval(p; x_1, x_2)$: This is an evaluation query, asking player p to report the value $v_p([x_1, x_2])$.

We may also consider approximate versions of the operations as defined in [14] (e.g. the player can return a cut point such that the value is within a $(1 + \epsilon)$ factor of λ). The complexity of a protocol is then given by the number of cut and evaluation queries that are performed. In some works it is only the cut operations that are counted towards the complexity, while evaluations are considered free (see [25]).

Although proportional fairness as well as approximate proportional fairness (i.e., every agent receives at least $(1-\epsilon)/n$ of the cake) have been to some extent understood, for envy-freeness, we still do not have satisfactory answers. For approximate envy-freeness the question is not resolved either. Here we will say that a partition is ϵ-envy-free with $\epsilon > 0$, if the maximum envy is at most ϵ. We exclude from this discussion the moving knife procedures as these cannot in general be simulated by cut and evaluation operations. Hence we are interested in the following problem:

Problem 1 *Design an algorithm that for any constant $\epsilon > 0$, it produces an ϵ-envy-free partition and uses a minimal number of cuts and as few evaluation queries as possible. What if $\epsilon = 0$?*

When n is constant, e.g., $n = 2, 3$ or 4, there exist procedures with bounded number of cuts, described in [7]. For arbitrary n, a procedure is described in [7] based on [30] which is ϵ-envy-free but needs an exponential number of cuts, $O((\frac{1}{\epsilon})^n)$. We are not aware of any other procedure that works for any number of agents and uses a number of cuts bounded by a function of n only. In [8] a trimming procedure is suggested, however the number of cuts depends on the preferences of the players and hence though finite, it cannot be bounded as a function of n.

We now show that by using our algorithm for the discrete case, we can obtain a new linear upper bound on the number of cuts for ϵ-envy-freeness and any constant $\epsilon > 0$ or more generally we can get $\frac{1}{p(n)}$-envy-freeness with a polynomial number of cuts for any polynomial $p(n)$.

Theorem 5 *For any constant $\epsilon > 0$, there exists a protocol that uses $O(n)$ cuts and $O(n^2)$ evaluation queries and produces an ϵ-envy-free allocation.*

Proof : Fix $\epsilon > 0$. We first ask every player to partition the interval $[0, 1]$ in contiguous subintervals $S_1, ..., S_k$ such that $0 < v_p(S_i) \leq \epsilon$ for all i. This is possible since the measures are divisible and it can be achieved by $\lceil 1/\epsilon \rceil - 1$ cut operations. In total we will get at most

$n(\lceil 1/\epsilon \rceil - 1)$ cut points and it holds that any interval within two consecutive cutpoints has value at most ϵ for any player. We can then treat these intervals as indivisible goods and apply Theorem 1. To run the algorithm of Theorem 1, we need roughly n^2/ϵ evaluation queries to be able to build the envy graph at every stage. □

If we do not want to allow our algorithm to handle irrational numbers, we can use the approximate versions of the cut and evaluation queries. In that case Theorem 5 still holds but with a worse constant in the $O(n)$ term (essentially the number of cuts will increase roughly from n/ϵ to $4n/\epsilon$). Note that the number of cut and evaluation queries that we obtain are in analogy with the result of Woeginger and Sgall [34] for ϵ-proportional fairness. In fact our Theorem 5 implies Theorem 1 of [34] as it can be easily checked that any ϵ-envy-free allocation is also $\frac{1-\epsilon}{n}$-fair (but not vice versa). It remains an open problem whether one can reduce the number of cuts to as close to $n-1$ as possible or whether the number of evaluation queries can be improved.

3 Minimizing Envy as an Optimization Problem

Coming back to the model with indivisible goods, in this section we address a different question. Although we can always have an allocation with an upper bound guarantee on the envy, in many instances the maximum envy can be smaller than α. Therefore we would like to look at the following optimization problem:

Problem 1: Minimum envy
Compute an allocation A that minimizes the envy

$$\max_{p,q}\{0, v_p(A_q) - v_p(A_p)\}$$

We can also think of using the ratio instead of the difference of the utilities as a measure for the envy. This leads us to a different optimization problem:

Problem 2: Minimum envy-ratio
Compute an allocation A that minimizes the envy-ratio

$$\max_{p,q}\{1, \frac{v_p(A_q)}{v_p(A_p)}\}$$

In the definition above we have ensured that the envy-ratio is always at least one and an allocation is envy-free if and only if the envy-ratio is equal to one.

As we will see it is not always possible to have a polynomial time algorithm for computing an optimal solution, hence we will also be interested in obtaining approximate solutions in the usual sense of multiplicative approximation error [32]:

Definition 1 *Given a minimization problem Π, we say that an algorithm has an approximation factor of ρ for Π ($\rho \geq 1$), if for any instance I of Π, the algorithm outputs a solution with value at most ρ $OPT(I)$, where $OPT(I)$ is the optimal solution for I.*

Definition 2 *An algorithm is a Polynomial Time Approximation Scheme (PTAS) if for any instance I and any error parameter $\epsilon > 0$, the algorithm runs in time polynomial in the input size, $|I|$, and outputs a solution with value at most $(1+\epsilon)OPT(I)$. If in addition the running time is polynomial in $1/\epsilon$ then we say that the algorithm is a Fully Polynomial Time Approximation Scheme (FPTAS).*

In the following theorem we show that when the utilities are non-additive, any algorithm needs an exponential number of queries in the worst case to produce an optimal solution or even an approximate one. We assume here that the algorithm is allowed to ask evaluation queries to the players over subsets of goods. Our construction is similar to Nisan and Segal [24].

Theorem 6 *Any deterministic algorithm that achieves any finite approximation for the minimum envy or minimum envy-ratio requires at least $\Omega(2^{m/2}/m)$ queries in the worst case.*

Proof : We give an outline of the proof. Suppose $m = 2k$. We consider the following class of utility functions $\mathcal{F}$. A function v is in $\mathcal{F}$ if:

$v(S) = 0$ for all S with $|S| < k$.

$v(S) = 1$ for all S with $|S| > k$.

$v(S) = 1 - v(\bar{S})$ for all $|S| = k$

The number of functions that belong to this family is doubly exponential in k and hence in m. Now, consider instances of the form (v, v) in which there are two players with the same utility function v for some $v \in \mathcal{F}$. If the algorithm asks less than the stated number of queries, it can be verified that there exist two different functions $u, v \in \mathcal{F}$ such that the query sequences corresponding to the instances defined by u and v are the same. Here by a query sequence, we mean a specification of the player being asked at each query, the subset over which he is being asked as well as the answer that is given by the player.

Consider now the instances (u, v) and (v, u). The algorithm will produce the same query sequences for both instances which will be the same as the query sequence for the instances (u, u) and (v, v). Hence it will produce the same allocation for (u, v) and (v, u). But it is easy to see that although for either case, there exists an allocation which is envy-free, there is no single allocation that is envy-free for both instances. Moreover, note that by the nature of the functions of $\mathcal{F}$, when an allocation is not envy-free, the envy is 1 and the envy-ratio is ∞. Hence for one of the two instances, the algorithm does not produce any finite approximation of the optimal solution. □

3.1 Additive Utilities

Given the strong inapproximability of the general problem, we consider here a natural special case of the problem in which the utility functions of all players are additive, i.e., for all $p \in N$, $v_p(S) = \sum_{i \in S} v_p(\{i\})$. In that case, an instance of the problem is specified by a $n \times m$ matrix $V = (v_{p,i})$, with $v_{p,i} = v_p(\{i\})$.

3.1.1 The minimum envy problem

Still, the problem of finding a minimum-envy allocation is **NP**-hard, even when the number of players is two and they have the same utility function. This follows from the fact that for two players with the same utility functions, deciding whether an envy-free allocation exists is equivalent to deciding if there exists a partition of a set of positive integers in two subsets of equal sum, which is **NP**-complete [32].

Since the objective function of the minimum envy problem can take the value zero for some instances and since distinguishing whether the minimum envy is zero or not is **NP**-complete, this implies that we cannot have a polynomial time algorithm with a finite approximation factor, unless $\mathbf{P} = \mathbf{NP}$. One potential way to remedy this is to add the constant term 1 to our objective function so that it is always positive. In that case, Theorem 1 guarantees a $(1+\alpha)$-approximation algorithm, where $\alpha = \max\ v_{p,i}$. Still, a strong hardness result holds in this case too. We can show that for any constant c, there is no 2^{m^c}-approximation algorithm unless $\mathbf{P} = \mathbf{NP}$. The proof is along the same lines as the inapproximability result for the problem Subset-Sums Difference in [6] and we omit it.

3.1.2 The minimum envy-ratio problem

We believe that a more suitable objective function for a multiplicative approximation guarantee is the envy-ratio. Using the same reduction as for the minimum envy problem, we get that the envy-ratio problem is also **NP**-complete even with two identical agents. In the rest of this section, we study the envy-ratio problem in the case of identical agents. Since now agents have the same utility function, we will denote by $v(i)$ the utility that a player derives from having good i, which is independent of the player.

This special case is closely related to a class of job scheduling problems on identical processors. We can think of the set of players as a set of identical machines and the set of goods as a set of m jobs to be scheduled on the machines. Every job has a positive processing time and the load of every processor is the sum of the processing times of the jobs assigned to it. Several objective functions have been considered in the scheduling literature such as minimizing the maximum completion time (makespan) [17, 19] or maximizing the minimum completion time [13, 33, 2]. Our problem is equivalent to minimizing the ratio of the maximum completion time over the minimum completion time.

The following greedy algorithm was proposed by Graham for the minimum makespan problem [17]: Sort the goods in decreasing order of their values and allocate them one by one in that order. At every step, allocate the next good to the player whose current bundle has the least value. In [9] it was proved (in the context of scheduling) that the approximation factor of Graham's algorithm is 1.4 for the ratio problem.

Theorem 7 *[9] Graham's algorithm achieves an approximation factor of* 1.4 *for the envy-ratio problem.*

In the next Theorem, we improve this result and show that we can achieve any constant factor arbitrarily close to 1 for the envy-ratio problem.

Theorem 8 *There is a PTAS for the envy-ratio problem when all players have the same utility for each good. Furthermore, when the number of players is constant, there is an FPTAS.*

Proof : Before going into the details of the proof we give a brief outline of the technique. Our algorithm is similar to [2] and [33]. However our objective function does not fit in their framework. The algorithm is as follows: Given our original instance, we round the utility of each good to obtain a coarsest instance in which there is a constant number of distinct utilities (i.e., a constant number of different types of goods). We then show that in the new instance, we can find an optimal solution by searching for every player among a constant number of distinct assignments of goods. The constant will be exponential in the approximation parameter $1/\epsilon$. This observation enables us to compute the optimal solution in the rounded instance by solving a series of integer programs with a constant number of variables using Lenstra's algorithm [20]. After finding an optimal allocation in the rounded instance, we will convert it into an allocation for the original instance. In the whole process, there are 2 sources of error: computing the rounded instance from the original one and transforming the optimal allocation of the rounded instance to an allocation of the original instance. We are able to bound the error by $1 + \epsilon$.

Let I be an instance of the problem, with n players, m goods and utility $v(i)$ for good i. If $m < n$ then the optimal envy-ratio is ∞ and any allocation is optimal. Hence we can assume without loss of generality that $m \geq n$. We start with some basic facts about the optimal solution.

Let L be the average utility of the players,

$$L = \frac{1}{n} \sum_{i \in M} v(i)$$

If all the goods were divisible, we could allocate a fraction of $1/n$ from each good to a player and everybody would receive a utility of exactly L.

We briefly sketch how to handle goods with utility greater than L. Suppose there exists a good i with $v(i) \geq L$. If i is assigned to a player p in an optimal allocation, then there is an allocation with the same or less envy-ratio in which i is the only good allocated to p. To see this, suppose that player p receives good i and some other good, say j in an optimal solution. Let q be the person with minimum utility and bundle S_{min}. Then $v(S_{min}) < L$ and by giving good j to q, it is easy to see that the ratio does not increase, and hence the new solution is also optimal. Therefore goods with high utility can always be assigned to players who will receive no other good by the algorithm. This does not mean yet that if we have a PTAS for instances where $v(i) < L$ for all i, we can derive a PTAS for the general problem, as is the case in [2]. Instead we will have to round "big" goods appropriately so that in the rounding instance their utility is also higher than the corresponding new average utility, L^R (defined below). We will then output an optimal solution for the rounded instance in which such goods are assigned to players with no other good in their bundle.

We omit the rest of the details for handling goods with $v(i) \geq L$ and from now on we will assume that $v(i) < L$ for every i. We have the following fact:

Claim 9 *If $v(i) < L$ for every good i, then there exists an optimal allocation $A = (A_1, ..., A_n)$ such that $\frac{1}{2}L < v(A_i) < 2L$.*

The proof is by showing that in a given optimal solution, it is possible to reallocate the goods so that the envy-ratio does not increase and the conditions of the claim are satisfied.

We will now describe how to round the values of the goods and obtain an instance in which there is only a constant number of different types of goods (i.e. a constant number of distinct values for the goods). The construction is the same as in [2] and we repeat it here for the sake of completeness.

We will denote the rounded instance by $I^R(\lambda)$, where λ is a positive constant and will be determined later (λ will be $O(1/\epsilon)$). We will often omit λ in the notation.

We first round the value of every good with relatively high value. In particular, for every good i with $v(i) > L/\lambda$, we round $v(i)$ to the next integer multiple of L/λ^2. Roughly this means that we round up the first few of the least significant digits of $v(i)$. We cannot afford to do the same for goods with small value since the error introduced by this process might be very big. Instead, let S be the sum of the values of the goods with value less than L/λ. We round S to the next integer multiple of L/λ, say S^R. Instance $I^R(\lambda)$ will have $S^R\lambda/L$ new goods with value L/λ. This completes the construction. Note that in $I^R(\lambda)$ all values are of the form kL/λ^2, where $\lambda \leq k \leq \lambda^2$. Hence we have only a constant number of distinct values.

Let M^R be the set of goods in the new instance and $v^R(i)$ be the value of each good. Let $L^R = \frac{1}{n}\sum_{j \in M^R} v^R(i)$. It is easy to see that $L \leq L^R$ and that all values in $I^R(\lambda)$ are at most L^R. Hence by Claim 9 there is an optimal solution $A^R = (A_1^R, ..., A_n^R)$ such that $\frac{1}{2}L^R < v(A_p^R) < 2L^R$ for every p. In the algorithm below we will search for such a solution.

We represent a player's bundle by a vector $t = (t_\lambda, ..., t_{\lambda^2})$, where t_k is the number of goods with value kL/λ^2 assigned to her. We will then say that the player is of type t. The utility derived from t is $v(t) = \sum_k t_k kL/\lambda^2$. Let U be the set of all possible types t, with $\frac{1}{2}L^R < v(t) < 2L^R$. It is easy to see that $|U|$ is bounded by a constant which is exponential in λ. Hence for a player of type $t \in U$, there is only a constant number of distinct values for her utility. Let $V(U)$ be the set of these values, i.e. $V(U) = \{u \ : \ v(t) = u \text{ for some } t \in U\}$.

We can now show how to find the optimal envy-ratio in $I^R(\lambda)$. For all pairs of values $u_1, u_2 \in V(U)$, we will solve the following decision problem: Is there an allocation in which the utility of every player is between u_1 and u_2? Since $|V(U)|$ is constant, after solving the above problem for all u_1, u_2 we can output the allocation corresponding to u_1^*, u_2^* for which the minimum ratio u_2^*/u_1^* is attained.

To solve the decision problem, we will write an integer program (IP) with a constant number of variables and use Lenstra's algorithm [20]. In the following IP, for each $t \in U$ we have an integer variable X_t indicating how many players are of type t. The first constraint implies that all players will obtain an allocation of type $t \in U$ and the second that all goods are assigned. It is obvious that the decision problem with inputs u_1, u_2 has a solution iff the corresponding integer program is feasible. Therefore we can find the optimal solution of $I^R(\lambda)$ in polynomial time.

In the following IP, $U_{u_1}^{u_2}$ is the set of all types $t \in U$ such that $u_1 \leq v(t) \leq u_2$ and n_k is the number of goods in $I^R(\lambda)$ of value kL/λ^2.

$$\begin{aligned}
&\sum_{t\in U} X_t = n \\
&\sum_{t\in U} X_t t_k = n_k \quad \forall k \\
&X_t \geq 0 \qquad\qquad \forall X_t \text{ with } t \in U^{u_2}_{u_1} \\
&X_t = 0 \qquad\qquad \forall X_t \text{ with } t \in U \setminus U^{u_2}_{u_1}
\end{aligned}$$

We need to see how the original instance is related to the rounded instance. The following Lemma has been proved in [2]:

Lemma 10 *Let $A = (A_1, ..., A_n)$ be an allocation in I, where $\frac{1}{2}L < v(A_i) < 2L$. Then there exists an allocation $B = (B_1, ..., B_n)$ in the rounded instance, I^R, such that:*

$$v(A_i) - \frac{1}{\lambda}L \leq v(B_i) \leq \frac{\lambda+1}{\lambda}v(A_i) + \frac{1}{\lambda}L$$

Also if $B = (B_1, ..., B_n)$ is an allocation in I^R such that $\frac{1}{2}L^R < v(B_i) < 2L^R$, then there exists an allocation $A = (A_1, ..., A_n)$, in I such that:

$$\frac{\lambda}{\lambda+1}v(B_i) - \frac{2}{\lambda}L \leq v(A_i) \leq v(B_i) + \frac{1}{\lambda}L$$

We are now ready to prove our Theorem. Our algorithm will be: Given instance I, compute the instance I^R, find an optimal allocation $A^R = (A^R_1, ..., A^R_n)$ for I^R, and then convert A^R to an allocation $A = (A_1, ..., A_n)$ for I using Lemma 10. Output A.

Suppose without loss of generality that $v(A^R_1) \leq ... \leq v(A^R_n)$ and $v(A_1) \leq ... \leq v(A_n)$. Let $A^* = (A^*_1, ..., A^*_n)$ be an optimal solution to I satisfying the conditions of Claim 9 and assume $v(A^*_1) \leq ... \leq v(A^*_n)$. We want to show:

$$\frac{v(A_n)}{v(A_1)} \leq (1+\epsilon)\frac{v(A^*_n)}{v(A^*_1})$$

By Lemma 10 we know that:

$$v(A_n) \leq v(A^R_n) + \frac{1}{\lambda}L \leq v(A^R_n) = \frac{1}{\lambda}L^R \leq v(A^R_n)(1+\frac{2}{\lambda})$$

Similar calculations yield: $v(A_1) \geq v(A^R_1)(\frac{\lambda}{\lambda+1} - \frac{4}{\lambda})$. Therefore:

$$\frac{v(A_n)}{v(A_1)} \leq \frac{v(A^R_n)}{v(A^R_1)}(\frac{1+\frac{2}{\lambda}}{\frac{\lambda}{\lambda+1} - \frac{4}{\lambda}})$$

We need to relate the optimal solution in I^R with the optimal solution in I. By using the first part of Lemma 10 and by performing similar calculations we have that there exists an allocation $A' = (A'_1, ..., A'_n)$ in I^R such that:

$$\frac{v(A'_n)}{v(A'_1)} \leq \frac{v(A^*_n)}{v(A^*_1)}(\frac{\frac{\lambda+1}{\lambda} + \frac{2}{\lambda}}{1-\frac{2}{\lambda}})$$

Since A^R is an optimal solution in I^R the ratio in A' will be at least as big as in A^R. Hence by combining the above equations we finally have:

$$
\begin{aligned}
\frac{v(A_n)}{v(A_1)} &\leq \frac{v(A_n^*)}{v(A_1^*)}\left(\frac{1+\frac{2}{\lambda}}{\frac{\lambda}{\lambda+1}-\frac{4}{\lambda}}\right)\left(\frac{\frac{\lambda+1}{\lambda}+\frac{2}{\lambda}}{1-\frac{2}{\lambda}}\right) \\
&\leq \frac{(\lambda+1)(\lambda+2)(\lambda+3)}{(\lambda-2)(\lambda^2-4\lambda-4)} OPT
\end{aligned}
$$

Thus, if we set $\lambda = 56/\epsilon$, it is easy to see that the factor will be at most $1+\epsilon$.

For the FPTAS in the case that the number of players is constant, the proof is by rounding the goods and running a dynamic programming algorithm [32]. □

4 Truthfulness

So far we have assumed that we can obtain the actual utilities of the players for the goods. However, in many situations players' utilities are private information and they might lie about their valuations in order to obtain a better bundle. We would like to investigate the question of whether we can have algorithms that elicit truthful utilities from the players and also produce allocations with minimum or bounded envy. A mechanism is truthful if for every player, her profit is maximized by declaring her true utility i.e. being truthful is a dominant strategy.

Truthful mechanisms have been extensively studied in other contexts such as auctions. However, unlike our problem, auction mechanisms are allowed to compensate the players with money whereas here we do not allow any money transfers between the players.

We will present a simple argument to prove that any mechanism that computes a minimum-envy allocation can not be truthful even in the special case where the utility functions are additive.

Theorem 11 *Any mechanism that returns an allocation with minimum possible envy cannot be truthful.*

Proof : Let $\mathcal{M}$ be a mechanism that outputs an allocation with minimum envy. Consider the following instance: We have two players, 1 and 2 and $k+2$ goods. In particular we have good a, good b and k eggs. Let's say $k = 100$. The k eggs are playing the role of an almost divisible good of value 0.2. Suppose the players have the following utilities for the goods:

$$v_1(a) = 0.45, v_1(b) = 0.35, v_1(\text{egg}) = 0.2/k,$$

$$v_2(a) = 0.35, v_2(b) = 0.45, v_2(\text{egg}) = 0.2/k$$

The specific instance admits an envy-free allocation: give to player 1 good a and 25 eggs and give the rest to player 2. Therefore in the allocation that $\mathcal{M}$ will output there will be no envy. Let A be the partition that $\mathcal{M}$ outputs for this instance. Note that in A each player receives exactly one of the goods a, b because if one player received both then the other player would envy her. Also we note that it is Player 1 who receives a. To see this, suppose on the contrary that player 1 receives b. Then in order for A to be envy-free player 1 should receive at least 75 eggs (otherwise the bundle S of player 1 is worth less than 1/2 and she will be envious). But then player 2 will receive a and at most 25 eggs so she will be envious, a contradiction. Therefore in A player 1 receives a and T eggs and player 2 receives b and $k-T$ eggs. It is also easy to see that $25 \leq T \leq 75$.

Case 1: $T < 74$

In this case player 1 can increase her utility by lying and declaring that good a has less value for her. It is possible for her to lie in such a way to force the mechanism to give her the good a and at least $T+1$ eggs (assuming that 2 does not change her declaration). She can declare that her valuation function is: $v_1(a) = 0.45 - \delta, v_1(b) = 0.35 + \delta, v_1(egg) = 0.2/k$ where δ is such that:

$$0.45 - \delta + (T+1)0.2/k = 1/2$$

Notice that under this new declaration, there still exists an envy-free outcome. Let A' be the new output of $\mathcal{M}$. Again player 1 will receive a. This is true because if player 1 receives b then she has to receive at least $k-T-1$ eggs so that she is not jealous. But then player 2 will receive good a and at most $T+1 \leq 74$ eggs which in total is worth less than 1/2. Hence player 1 will get a and at least $T+1$ eggs (otherwise her bundle is worth less than 1/2) which is more than what she gets if she is honest.

Case 2: $T \geq 74$

Now it is player 2 who can try to cheat. By misreporting her utilities in a similar manner as in Case 1, she can obtain a higher utility than before. □

In the rest of the section, we present a simple truthful algorithm which allocates the goods to the players uniformly at random. We assume that the sum of the utilities of each player over all goods is one. We show that with high probability the maximum envy of the resulting allocation is no more than $O(\sqrt{\alpha} n^{1/2+\epsilon})$.

Theorem 12 *Suppose that $v_{p,i} \leq \alpha \quad \forall p \in N, j \in M$. Then for every $\epsilon > 0$, and for large enough n, there exists a truthful algorithm such that with high probability the allocation output by the algorithm has maximum envy at most $O(\sqrt{\alpha}\ n^{1/2+\epsilon})$.*

Proof : The proof is based on the probabilistic method. Allocate each good independently to player p with probability $1/n$. Clearly this is a truthful mechanism. We will show that with high probability, the allocation produced satisfies the desired bound. Fix two players p, q. Given p and q we define a random variable Y_j indicating the contribution of good j to the envy of player p for q. The variable Y_j is equal to 1, if good j is allocated to player q, -1, if it is allocated to player p, and 0 otherwise. Hence: $Y_j = 1$ with probability $1/n$, -1 w.p. $1/n$ and 0 w.p. $(n-2)/n$. We now define the random variable: $f_{pq} = \sum_j v_{p,j} Y_j$. Clearly the envy of p for q is $e_{pq} = max\{0, f_{pq}\}$. We will show that with high probability, for every p, q, $f_{pq} \leq O(\sqrt{\alpha}\ n^{1/2+\epsilon})$ and this will complete the proof.

The expectation of f_{pq} is:

$$E[f_{pq}] = \sum_j E[Y_j] v_{p,j} = 0$$

To compute the variance, note that the variables $\{Y_j\}$ are independent. Thus:

$$Var[f_{pq}] = \sum_j v_{p,j}^2 Var[Y_j] = \frac{2}{n}\sum_j v_{p,j}^2 \leq \frac{2\alpha}{n}\sum_j v_{p,j} = \frac{2\alpha}{n}$$

By using Chebyshev's inequality, we have that for any ordered pair of players p, q such that $p \neq q$ and for $t > 0$:

$$Pr[|f_{pq}| \geq t] \leq \frac{2\alpha}{nt^2}$$

Hence we have:

$$\begin{aligned} Pr[\max_{p,q} f_{pq} < t] &= Pr[\bigcap_{(p,q)} f_{pq} < t] = 1 - Pr[\bigcup_{(p,q)} f_{pq} \geq t] \\ &\geq 1 - \sum_{(p,q)} \frac{2\alpha}{nt^2} \geq 1 - \frac{2\alpha n}{t^2} \end{aligned}$$

If we set $t = 2\sqrt{\alpha}\ n^{1/2+\epsilon}$ we have that:

$$Pr[\text{max. envy } < 2\sqrt{\alpha}n^{1/2+\epsilon}] \geq 1 - n^{-2\epsilon} \qquad (1)$$

□

5 Discussion and Open Problems

Our algorithm for minimizing the envy-ratio works only if the utility functions of the players are the same. It would be very interesting to find an approximation algorithm for the general case. One approach is to use a linear programming relaxation similar to Lenstra et al. [21].

There are many related notions of fairness such as max-min fairness or proportional fairness and we would like to know the complexity of these solution concepts as well. For max-min fairness some recent progress has been made in [5, 4, 11].

Another interesting question concerns obtaining tradeoffs between fairness and optimality of a solution. We call an allocation optimal if it maximizes the social welfare of the players. Such a tradeoff can be seen as the social cost of fairness.

Finally, there might be an interesting connection between finding a market equilibrium and minimizing the envy. Imagine that we give one dollar to each one of the players and have them buy their favorite goods in the market. If the market clears (it might not, because the goods are not divisible) then the allocation is envy-free. If the market does not clear, the deficiency of the market according to [12] is closely related to the maximum envy-ratio.

6 Acknowledgments

We would like to thank Ted Hill for many discussions and encouraging comments. We would also like to thank Steven Brams, Kamal Jain, David Shmoys, Vijay Vazirani and Gerhard Woeginger for useful discussions and pointers to the literature.

References

[1] A. Alkan, G. Demange, and D. Gale. Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59:1023–1039, 1991.

[2] N. Alon, Y. Azar, G. Woeginger, and T. Yadid. Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, 98:55–66, 1998.

[3] E. Aragones. A derivation of the money rawlsian solution. *Social Choice and Welfare*, 12:267–276, 1995.

[4] A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *ACM Symposium on Theory of Computing (STOC)*, pages 114–121, 2007.

[5] N. Bansal and M. Sviridenko. The santa claus problem. In *ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2006.

[6] C. Bazkan, M. Santha, and Z. Tuza. Efficient approximation algorithms for the subset-sum problem. In *ICALP*, pages 387–396, 1991.

[7] S. J. Brams and A. D. Taylor. *Fair Division: from Cake Cutting to Dispute Resolution.* Cambrige University press, 1986.

[8] S. J. Brams and A. D. Taylor. An envy-free cake division protocol. *American Mathematical Monthly*, 102(1):9–18, 1995.

[9] E. G. Coffman and M. A. Langston. A performance guarantee for the greedy set-partitioning algorithm. *Acta Informatica*, 21:409–415, 1984.

[10] M. Dall'Aglio and T. Hill. Maximin-share and minimax-envy in fair division problems. *Journal of Mathematical Analysis and Applications*, 2003.

[11] M. Dall'Aglio and R. Mosca. How to allocate hard candies fairly. To appear in *Mathematical Social Sciences*, 2008.

[12] X. Deng, C. H. Papadimitriou, and M. Safra. On the complexity of equilibria. In *ACM Symposium on Theory of Computing (STOC)*, pages 67–71, 2002.

[13] B. L. Deuermeyer, D. K. Friesen, and M. A. Langston. Scheduling to maximize the minimum procesor finish time in a multiprocessor system. *SIAM Journal of Algebraic and Discrete Methods*, 3:190–196, 1982.

[14] J. Edmonds and K. Pruhs. Cake cutting really is not a piece of cake. In *SODA*, pages 271–278, 2006.

[15] S. Even and A. Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7:285–296, 1984.

[16] D. Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7:45–98, 1967.

[17] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics*, 17:416–429, 1969.

[18] T. Hill. Partitioning general probability measures. *The Annals of Probability*, 15(2):804–813, 1987.

[19] D. S. Hochbaum and D. B. Shmoys. Using dual approximation algorithms for scheduling problems. *Journal of the ACM*, 34:144–162, 1987.

[20] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8:538–548, 1983.

[21] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.

[22] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[23] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors. *Algorithmic game theory.* Springer-Verlag, 2007.

[24] N. Nisan and I. Segal. The comminication requirements of efficient allocations and supporting lindahl prices. To appear in Journal of Economic Theory, preliminary version available at http://www.cs.huji.ac.il/ noam/mkts.html, 2004.

[25] J. M. Robertson and W. A. Webb. Approximating fair division with a limited number of cuts. *Journal of Combinatorial Theory, Series A*, 72:340–344, 1995.

[26] J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: be fair if you can.* AK Peters, 1998.

[27] H. Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[28] W. Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.

[29] K. Tadenuma and W. Thomson. The fair allocation of an indivisible good when monetary compensations are possible. *Mathematical Social Sciences*, 25(2):117–132, 1993.

[30] A. D. Taylor. Algorithic approximations of simultaneous equal division. Preprint, Dept. of Mathematics, Union College, 1993.

[31] H. Varian. Equity, envy and efficiency. *Journal of Economic Theory*, 9:63–91, 1974.

[32] V. V. Vazirani. *Approximation Algorithms.* Springer-Verlag, 2001.

[33] G. Woeginger. A polynomial time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20:149–154, 1997.

[34] G. Woeginger and J. Sgall. On the complexity of cake cutting. *Discrete Optimization*, 4(2):213–220, 2007.