THEORY OF COMPUTING, Volume 6 (2010), pp. 247–290
www.theoryofcomputing.org

The Submodular Welfare Problem with Demand Queries

Uriel Feige* Jan Vondrák†

Received: April 17, 2009; published: December 9, 2010.

Abstract: We consider the *Submodular Welfare Problem* where we have m items and n players with given utility functions $w_i : 2^{[m]} \to \mathbb{R}_+$. The utility functions are assumed to be monotone and submodular. We want to find an allocation of disjoint sets $S_1, S_2, \ldots, S_n$ of items maximizing $\sum_i w_i(S_i)$. A $(1 - 1/\mathrm{e})$-approximation for this problem in the *demand oracle model* has been given by Dobzinski and Schapira [5]. We improve this algorithm by presenting a $(1 - 1/\mathrm{e} + \varepsilon)$-approximation for some small fixed $\varepsilon > 0$.

We also show that the Submodular Welfare Problem is *NP*-hard to approximate within a ratio better than some $\rho < 1$. Moreover, this holds even when for each player there are only a constant number of items that have nonzero utility. The constant size restriction on utility functions makes it easy for players to efficiently answer any "reasonable" query about their utility functions. In contrast, for classes of instances that were used for previous hardness of approximation results, we present an incentive compatible (in expectation) mechanism based on *fair division queries* that achieves an optimal solution.

ACM Classification: F.2.2

AMS Classification: 68W25,68W20

Key words and phrases: combinatorial auction, welfare maximization, submodular function

1 Introduction

The following problem has appeared in the context of combinatorial auctions.

*Work was done while the author was at Microsoft Research, Redmond, WA.
†Work was done while the author was at Microsoft Research, Redmond, WA.

DOI: 10.4086/toc.2010.v006a011

Problem 1.1. Given m items and n players with utility functions $w_i : 2^{[m]} \to \mathbb{R}_+$, find a partition $[m] = S_1 \cup S_2 \cup \ldots \cup S_n$ that maximizes $\sum_{i=1}^{n} w_i(S_i)$.

This is what we call a *combinatorial allocation problem*, where items are to be allocated to players with different interests in different combinations of items, in order to maximize their total utility.

Oracle models Since an explicit description of a utility function requires exponential space, we have to clarify the issue of accessing the input. Unless the utility functions have a special form which allows us to encode them efficiently, we have to rely on *oracle access*. This means we have a "black box" for each player, which answers queries about her utility function. Two types of oracles have been considered in the literature.

Value oracle The most basic query is: *What is the value of $f(S)$?* An oracle answering such queries is called a *value oracle*.

Demand oracle Sometimes, a more powerful oracle is considered, which can answer queries of the following type: *Given an assignment of prices to items $p : [m] \to \mathbb{R}$, what is* $\max_S (f(S) - \sum_{j \in S} p_j)$*?* Such an oracle is called a *demand oracle*.

Whether we want to allow the use of a demand oracle depends on a particular setting. From an economic standpoint, it seems natural to assume that given an assignment of prices, a player can decide which set of items is the most valuable for her. On the other hand, from a computational point of view, this decision problem is *NP*-hard for some very natural submodular utility functions (e. g., coverage-type functions). Thus we can either assume that players have sufficient knowledge of their utility functions (or the utility functions are simple enough) so that they are able to answer demand queries, or we can restrict ourselves to the value oracle model. In this paper, we follow the first path and assume that a demand oracle is available for each player.

Submodularity By the Submodular Welfare Problem, we mean the combinatorial allocation problem where utility functions are *monotone and submodular*. A function is monotone if $f(S) \leq f(T)$ whenever $S \subseteq T$. Submodularity is a discrete analogue of concavity and can be defined in three equivalent ways:

Standard definition For any S, T,

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T).$$

Decreasing marginal values For any $S \subset T$ and $j \notin T$,

$$f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T).$$

Subadditive marginal values For any S, T, T',

$$f(S \cup T \cup T') - f(S) \leq (f(S \cup T) - f(S)) + (f(S \cup T') - f(S)).$$

The second definition has the natural interpretation that the marginal value of an item cannot increase by including some other additional items. This property has been known in economics as the property of *diminishing returns*. It is natural to assume this in certain settings, where items do not complement each other in any way.

Truthfulness We remark that in the context of combinatorial auctions, the utility functions are actually unknown and players are not assumed to be necessarily willing to reveal their true valuations. This leads to the design of *incentive-compatible mechanisms*, where players are not only queried about their valuations but also motivated to answer truthfully. We do not deal with this issue here and assume instead that players are willing to cooperate and give true answers to our queries.

Previous work Without any assumptions on the utility functions, the problem is at least as hard as Set Packing (which corresponds to "single-minded bidders" who are interested in exactly one set each). Hence, no reasonable approximation (significantly better than $m^{1/2}$) can be expected in general [11, 17]. Research has focused on classes of utility function that allow better positive results, in particular submodular utility functions. Lehmann, Lehmann and Nisan [14] provide an approximation ratio of $1/2$ for the Submodular Welfare Problem, using a simple greedy algorithm using only value queries. (This also follows from the work of Fisher, Nemhauser and Wolsey on submodular maximization subject to a matroid constraint [10].) A randomized version of this algorithm is shown in [5] to give a somewhat improved approximation ratio of $n/(2n-1)$. It is shown in [13] that if only value queries are allowed, then it is *NP*-hard to approximate Submodular Welfare within a ratio strictly better than $1-1/\mathrm{e}$.

Several subsequent works [4, 5, 7] considered the following linear programming relaxation of the problem, usually referred to as the *Configuration LP*. Here, $x_{i,S}$ is intended to be an indicator variable that specifies whether player i gets set S.

Configuration LP Maximize $\sum_{i,S} x_{i,S} w_i(S)$ subject to:

- Item constraints: $\sum_{i,S:j\in S} x_{i,S} \leq 1$ for every item j.
- Player constraints: $\sum_S x_{i,S} \leq 1$ for every player i.
- Nonnegativity constraints: $x_{i,S} \geq 0$.

This linear program has an exponential number of variables but only a polynomial number of constraints. Using the fact that the separation oracle for the dual is exactly the demand oracle, this LP can be solved optimally in the demand oracle model. We refer the reader to [16, 4] for more details.

The integrality gap of this LP is known (up to lower-order terms) for classes of utility functions that are more general than submodular. For subadditive utility functions,[1] it is $1/2$ [7], and for fractionally subadditive utility functions[2] it is $1-1/\mathrm{e}$ [5]. A $1/2$-approximation for subadditive utilities and a $(1-1/\mathrm{e})$-approximation for fractionally subadditive functions were given in [7]. Both algorithms run in the demand oracle model. For submodular utility functions, which are strictly contained in the classes we just mentioned, the positive results still apply and hence there is a $(1-1/\mathrm{e})$-approximation in the demand oracle model [5]. Recently, it has been shown that a $(1-1/\mathrm{e})$-approximation for Submodular Welfare is possible even in the value oracle model [19] which is optimal [13]. Prior to this work, however, it was not known whether this approximation factor can be improved in the demand oracle model. Also, it was not known whether the Configuration LP actually has an integrality gap arbitrarily close to $1-1/\mathrm{e}$ for submodular utility functions. An example of integrality gap $7/8$ was given in [5].

[1] f is subadditive if $f(A\cup B) \leq f(A)+f(B)$ for any A,B.

[2] f is fractionally subadditive if $f(S) \leq \sum_i \alpha_i f(A_i)$ for any positive linear combination such that $\sum_{i:j\in A_i} \alpha_i \geq 1$ for each $j \in S$.

Our results Our main result is the following.

Theorem 1.2. *There is some universal constant $\varepsilon > 0$ and a randomized rounding procedure for the Configuration LP, such that given any feasible fractional solution, the rounding procedure produces a feasible allocation whose expected value is at least a $(1-1/\mathrm{e}+\varepsilon)$-fraction of the value of the fractional solution.*

Our rounding procedure is oblivious in the sense of [7]: its only input is a fractional LP solution and it need not know anything about the actual utility functions of the players. We obtain the following corollary by combining Theorem 1.2 with the fact that demand queries suffice in order to find an optimal fractional solution to the Configuration LP.

Corollary 1.3. *The Submodular Welfare Problem can be approximated in the demand oracle model within a ratio of $1-1/\mathrm{e}+\varepsilon$ (in expectation), for some absolute constant $\varepsilon > 0$.*

The value of ε that we obtain is very small; in this paper we show roughly $\varepsilon \simeq 10^{-12}$. A more technical proof yielding $\varepsilon \geq 10^{-5}$ is presented in [18]. The significance of this result is that $1-1/\mathrm{e}$ is not the optimal answer, and hence it is likely that further improvements are possible. Another way to look at this result is that the integrality gap of the Configuration LP with submodular functions cannot be arbitrarily close to $1-1/\mathrm{e} \simeq 0.632$. We do not determine the worst case integrality gap; on the negative side, we improve the example of $7/8$ from [5] to an example with a gap of roughly 0.782 (see Section 3.4).

The reader who wishes to go straight to the proof of this statement can start reading from the beginning of Section 3. As a warm-up, we start with the case of two players (Section 2) which is instructive and conceptually important in the development of our ideas.

Hardness results This paper also contains some *APX*-hardness results. (A maximization problem is *APX*-hard if there is some constant $\rho < 1$ such that it is *NP*-hard to achieve approximation ratios better than ρ.) Contrasting Theorem 1.2 with the hardness results of [13] clearly shows that the best approximation ratio for Submodular Welfare depends on the query model that is allowed. Is there a limit to the best approximation ratio one can achieve, as we consider progressively stronger query models? Prior to our work, this was not known to hold even for the demand query model.

Concurrently and independently of our work, Dobzinski and Schapira [private communication] proved *APX*-hardness in the demand oracle model. More recently, Chakrabarty and Goel [3] proved that in the same model it is *NP*-hard to approximate the Submodular Welfare Problem within a ratio better than $15/16$.

In contrast to these results, we consider utility functions of a very restricted type. Hardness results using such utility functions are valid not only in the demand oracle model, but a much wider class of oracle models.

Constant size utility functions. Every player is interested only in a constant (say, t) number of the items. All other items have value 0 for the player. Hence the utility function of a player can be explicitly given by a table with a constant number of entries 2^t. This allows every player to answer demand queries (and essentially any other type of query) efficiently—in constant time. In particular, the Configuration LP then has only a polynomial number of "relevant" variables, and hence can be solved efficiently.

Theorem 1.4. *The Submodular Welfare Problem is APX-hard. Moreover, this holds even if every player has a constant size utility function.*

We remark that in previous *APX*-hardness results (say, for fractionally subadditive utility functions [4]), all players typically have the same utility function, and hence this utility function cannot be of constant size (as then the optimal allocation could be found by enumerating all possible allocations). To illustrate how certain query models might circumvent previous *NP*-hardness results we consider a class of queries that we call *fair division queries*. We give a polynomial time incentive compatible mechanism based on fair division queries that extracts the maximum welfare whenever all players have the same utility function, regardless of whether they are submodular or not. It is our opinion that no "reasonable" query model will be able to circumvent *NP*-hardness results when utility functions are of constant size. For this reason, in this paper we also indicate how previous *NP*-hardness results can be modified to hold also when utility functions are of constant size. In addition, this paper contains *NP*-hardness of approximation results in some cases where utility functions cannot be of constant size, for example, when there are only two players. We present our hardness results in Section 4.

Remark This is a detailed version of our extended abstract [9], where we presented a similar result also for the Generalized Assignment Problem. More details on the Generalized Assignment Problem can be found in [18]. Here, we focus on the Submodular Welfare Problem.

1.1 Overview of techniques

First, let us recall how one achieves an approximation ratio of $1-1/e$ (or in fact, $1-(1-1/n)^n$) for the Submodular Welfare Problem [5, 7]. One uses a given feasible solution to the Configuration LP in the following way. Every player independently selects a tentative set, where the probability that player i selects set S is precisely $x_{i,S}$. (If $\sum_S x_{i,S} < 1$ for some player, then it may happen that the player will select no tentative set at all.) Per player, the expected utility after this step is equal to her contribution to the value of the fractional solution of the Configuration LP. However, the tentative allocation may not be feasible, because some items may be in more than one tentative set. To resolve this issue, one uses a "contention resolution procedure." In [5], contention resolution is based on further queries to the players is order to determine which player will benefit the most from getting the contended item. In [7], contention resolution is done in an oblivious way, without further interaction with the players. In both cases it is shown that contention resolution can be done while preserving (in expectation) at least a $(1-1/e)$-fraction of the total utility.

The above two-step rounding procedure seems to have a lot of slackness. Namely, there are many items that are not allocated at all because they are not in any tentative set. In fact, on what appear to be worst case instances of the two-step rounding procedure, every item has probability roughly $1/e$ of not being allocated at all. It appears that adding a third step to this rounding procedure, in which the remaining items are allocated, could potentially allow one to improve the $1-1/e$ ratio.

Somewhat surprisingly, one can design instances (see Section 1.4) where the utility functions are submodular, and regardless of how contention resolution is performed, and how items outside the tentative sets are allocated, one cannot obtain an approximation ratio better than $1-1/e$.

We show in Section 1.3 that there is also a simpler *one-step* randomized rounding procedure (namely, item j is given to player i with probability $\sum_{j \in S} x_{i,S}$, without a need to first choose tentative sets), which achieves an approximation ratio of $1 - 1/\mathrm{e}$ (though not $(1-(1-1/n)^n)$). One may hope that on every instance, the best of the two rounding procedures (the three-step procedure and the one-step procedure) would give an approximation ratio better than $1 - 1/\mathrm{e}$. Nevertheless, again, this is not true.

Our new algorithm (that does improve upon $1 - 1/\mathrm{e}$) uses a combination of two rounding techniques. The analysis of our algorithm uses algebraic manipulations that might not be easy to follow. To help the reader see the reasoning behind our final algorithm, we break its derivation into stages.

Our first rounding technique is the *Fair Rounding Procedure* (Section 1.3) which achieves at least a $(1-1/\mathrm{e})$-approximation, and actually beats it if the fractional solution is "unbalanced." This stage is based on a *Fair Contention Resolution* technique which might be interesting on its own. It is a variation and simplification of the contention resolution technique used in [7]. Apart from other interesting features (that we discuss in Section 1.2), the Fair Rounding Procedure procedure has the following property. Call a fractional solution balanced if $y_{ij} = \sum_{S: j \in S} x_{i,S} = 1/n$ for each player i and item j. If the fractional solution deviates significantly from being balanced, then the Fair Contention Resolution technique already achieves a factor better than $1 - 1/\mathrm{e}$. Thus, the heart of the problem is to deal with fractional solutions which are balanced.

Section 2.3 is perhaps the most instructive one, as it addresses a simple case which is relatively easy to understand. There are only two players, and the fractional solution is half-integral. The goal is to design a rounding procedure that improves the ratio of $1-(1-1/2)^2 = 3/4$ guaranteed by previous rounding procedures. Neither the three-step rounding procedure (as described above, based on each player choosing one tentative set) nor the one-step rounding procedure achieve such an improvement. We present a new rounding procedure that achieves an approximation ratio of 5/6. Moreover, our analysis of the approximation ratio serves as an introduction to the kind of algebraic manipulations that will be used in the more complicated proofs. We also show that the 5/6 ratio for this case is best possible, by showing a matching integrality gap.

Section 2.4 deals with the case of two players and a balanced fractional solution, in the sense that for every item j, $\sum_{S: j \in S} x_{1,S} = \sum_{S: j \in S} x_{2,S} = 1/2$. We design an algorithm using two tentative sets S, S' for player 1 and T, T' for player 2, sampled independently according to the fractional solution. Using two sets instead of one allows us to allocate more items overall, and also it gives us more freedom in designing an allocation scheme. Using a submodularity argument inspired by the half-integral case, we show that either a player gains by taking the entire complement of the other player's tentative set, or she can combine items from S, S' to obtain what we call a "diagonal set" Y. Similarly, player 2 obtains a diagonal set Z. The sets Y and Z are designed so that their average overlap is less than that of S and T. Thus by resolving contention between Y and Z, we get a factor better than $3/4$.

Section 2.5 combines the balanced and unbalanced case for two players. We gain for different reasons in the balanced and unbalanced cases, but we always beat the factor of $3/4$. This result convinced us that an improvement over $1 - 1/\mathrm{e}$ in the general case of n players should be possible.

Considering the aforementioned Fair Rounding Procedure, it remains to handle the case of balanced fractional solutions. In Section 3.2, we develop the "Butterfly Rounding Procedure" which achieves an improvement over $1 - 1/\mathrm{e}$ for any balanced fractional solution for n players. This is the key part of the proof of Theorem 1.2. It is based on ideas used for the two-player case, but again, with added complications. The main structural difference between this rounding procedure and earlier ones is that we let every player choose two tentative sets rather than one. Thereafter, we perform contention resolution for every item that is in tentative sets of more than one player. The exact way in which we perform contention resolution is rather complicated.

Finally, the Fair and Butterfly Rounding Procedures are combined to obtain an approximation ratio strictly above $1-1/e$ for any fractional solution (Section 3.3).

1.2 Fair contention resolution

A key component in previous $(1-1/e)$-approximation algorithms for Submodular Welfare (and for more general classes of utility functions as well) is a method for resolving contention among several tentative sets that contain the same item. In our current work, we generalize and improve upon the method used for this purpose in [7] so that it can be combined more easily with other parts of our new rounding procedure. Our method gives a solution to a problem that we call *Fair Contention Resolution*. We now describe this problem.

There are n players and one item. Every player i is associated with a probability $0 \le p_i \le 1$ of requesting the item. Our goal is to allocate the item to at most one player, and have the probability that a player i receives the item be proportional to its respective p_i. We call this *balanced contention resolution*. Among all such contention resolution schemes, we wish to find one that maximizes for the players the probability that they get the item (the balancing requirement implies that maximizing this probability for one player maximizes it for all players). Given complete coordination among the players, we may assign the item to player i with probability $p_i/\sum_j p_j$, and this would be optimal. But we will be dealing with a two-step situation in which there is only partial coordination.

1. In step 1, there is no coordination among the players. Every player i independently requests the item with probability p_i.
2. In step 2, those players who requested the item in step 1 may coordinate a (randomized) strategy for allocating the item to one of them.

The probability that the item is allocated at all is at most the probability that the set of players reaching step 2 is nonempty, namely, $1-\prod_j(1-p_j)$. Hence in balanced contention resolution, player i can get the item with probability at most

$$\frac{p_i}{\sum_j p_j}\left(1-\prod_j(1-p_j)\right).$$

What we call *Fair Contention Resolution* is a method which indeed attains this maximum. In previous work [7], such contention resolution methods were designed for some special cases. Here, we design a general technique for an arbitrary number of players and any choice of probabilities p_j.

Fair Contention Resolution Suppose n players compete for an item independently with probabilities $p_1, p_2, \ldots, p_n$. Denote by A the random set of players who request the item, i. e., $\Pr[i \in A] = p_i$ independently for each i.

- If $A = \emptyset$, do not allocate the item.

- If $A = \{k\}$, allocate the item to player k.
- If $|A| > 1$, allocate the item to each $k \in A$ with probability

$$r_{A,k} = \frac{1}{\sum_{i=1}^{n} p_i} \left(\sum_{i \in A \setminus \{k\}} \frac{p_i}{|A|-1} + \sum_{i \notin A} \frac{p_i}{|A|} \right).$$

It can be seen that $r_{A,k} \geq 0$ and $\sum_{k \in A} r_{A,k} = 1$ so this is a valid probability distribution. The following lemma shows that this is indeed the best possible balanced contention resolution technique; we defer the proof to Section 3.1.

Lemma 1.5. *Conditioned on player k requesting the item, she obtains it with probability exactly*

$$\rho = \frac{1 - \prod_{i=1}^{n}(1-p_i)}{\sum_{i=1}^{n} p_i}.$$

1.3 A review of $(1-1/e)$-approximations

Prior to this work, there have been two known ways of achieving a $(1-1/e)$-approximation for the Submodular Welfare Problem [5, 7]. They are both based on solving the Configuration LP using demand queries and then rounding the fractional solution to an integral one. To put the question of improving $1-1/e$ in perspective, and to pave our way for our subsequent considerations, we present two new algorithms here. We assume in the following that $x_{i,S}$ is an optimal solution to the Configuration LP.

The fair rounding procedure

- Let each player i sample a set S_i from her probability distribution.
- For each item j, use the Fair Contention Resolution technique to decide which of the players i such that $j \in S_i$ receives the item.

Before analyzing this algorithm, let us introduce a class of utility functions that is more general than submodular functions.

Definition 1.6. A function w is *fractionally subadditive* if $w(S) \leq \sum \alpha_i w(T_i)$ for any collection of $\alpha_i \geq 0$ and sets T_i such that $\sum_{i: j \in T_i} \alpha_i \geq 1$ for all $j \in S$.

I. e., if the sets T_i form a “fractional cover” of S, then the sum of their utilities weighted by the corresponding coefficients is at least as large as that of S. It is known [7] that this is equivalent to the “XOS” property, where f is XOS if $f(S) = \max_j g_j(S)$ with each g_j linear, and also that submodular functions are a subclass of fractionally subadditive functions. In this paper, we use the terms *XOS* and *fractionally subadditive* interchangeably. The key to the analysis of this algorithm is the following lemma stated in [7].

Lemma 1.7. *Let $p \in [0,1]$ and $w : 2^{[m]} \to \mathbb{R}_+$ fractionally subadditive. For a set S, consider a random subset $S' \subseteq S$ such that each element of S is included in S' with probability at least p (not necessarily independently). Then*

$$\mathbf{E}[w(S')] \geq p\, w(S)\,.$$

Proof. For each $T \subseteq S$, let p_T denote the probability that $S' = T$. Define also $\alpha_T = p_T/p$. For each $i \in S$, we have

$$\sum_{T:i\in T} \alpha_T = \frac{1}{p}\sum_{T:i\in T} p_T = \frac{1}{p}\Pr[i \in S'] \geq 1$$

by the assumption that $\Pr[i \in S'] \geq p$. This means that $\{\alpha_T\}_{T\subseteq S}$ defines a fractional cover of S and by fractional subadditivity,

$$w(S) \leq \sum_{T\subseteq S} \alpha_T w(T) = \frac{1}{p}\sum_{T\subseteq S} p_T w(T) = \frac{1}{p}\mathbf{E}[w(S')]\,.$$

□

Using this lemma, we can show the following.

Lemma 1.8. *For n players with fractionally subadditive utility functions, the Fair Rounding Procedure delivers expected value at least $(1-(1-1/n)^n)\sum_S x_{i,S}\, w_i(S)$ to player i.*

Proof. Each player requests a set of expected value $\mathbf{E}[w_i(S_i)] = \sum_S x_{i,S}\, w_i(S)$. Define

- $y_{ij} = \sum_{S:j\in S} x_{i,S} = \Pr[j \in S_i]$

i. e., the probability that player i competes for item j. By Lemma 1.5, conditioned on player i competing for item j, the item is allocated to her with probability

$$\rho = \frac{1-\prod_{i=1}^{n}(1-y_{ij})}{\sum_{i=1}^{n} y_{ij}} \geq 1-\left(1-\frac{1}{n}\right)^n$$

since $\sum_{i=1}^{n} y_{ij} \leq 1$ and by the arithmetic-geometric mean inequality, the worst case occurs when $y_{ij} = 1/n$ for all i. This is done independently of the particular set S_i containing j that the player has chosen. Therefore, conditioned on any particular chosen set S_i, the player obtains each of its items with probability at least $1-(1-1/n)^n$. The result follows by Lemma 1.7. □

This improves the approximation factor of $1-1/e$ for any fixed number of players (which was known due to [5]). However, our goal is to obtain an absolute constant larger than $1-1/e$, independent of n. (Also, we know that $1-1/e$ cannot be improved for fractionally subadditive utility functions in general.) An even simpler $(1-1/e)$-approximation (only for submodular utilities) is the following.

The simple rounding procedure

- Let $y_{ij} = \sum_{S:j\in S} x_{i,S}$. For each j, we have $\sum_{i=1}^{n} y_{ij} \leq 1$.
- Assign each item independently, to player i with probability y_{ij}.

The fact that this gives a $(1-1/\mathrm{e})$-approximation can be seen in several ways. Given the recent work on submodular maximization subject to a matroid constraint [2], we can argue as follows. For a monotone submodular function w_i, define $f_i^+ : [0,1]^m \to \mathbb{R}_+$ by

$$f_i^+(\vec{y}_i) = \max\left\{\sum_S x_{i,S} w_i(S) : \sum_S x_{i,S} \leq 1, x_{i,S} \geq 0 \ \&\ \forall j, \sum_{S:j\in S} x_{i,S} \leq y_{ij}\right\}.$$

Here $\vec{y}_i$ is the vector with coordinates y_{ij} for $j = 1,\ldots,m$. So the Configuration LP can be written equivalently as

$$\begin{aligned} \max \sum_i f_i^+(\vec{y}_i), \\ \forall j, \sum_i y_{ij} \leq 1, \\ \forall i,j, y_{ij} \geq 0, \end{aligned}$$

and $f_i^+(\vec{y}_i)$ corresponds to the share of player i in the fractional solution. Our Simple Rounding Procedure allocates to player i a random set S_i which is obtained by rounding the coordinates of the vector $\vec{y}_i$ independently to 0 and 1, based on probabilities y_{ij}. The extension $F(y) = \mathbf{E}[f(\hat{y})]$ defined in [2] corresponds exactly to the expected value of such a set, $F_i(\vec{y}_i) = \mathbf{E}[w_i(S_i)]$. From [2] (Lemma 4 and 5), it follows that

$$\mathbf{E}[w_i(S_i)] = F_i(\vec{y}_i) \geq (1-1/\mathrm{e}) f_i^+(\vec{y}_i).$$

In other words, each player receives in expectation at least a $(1-1/\mathrm{e})$-fraction of her LP share.

1.4 Obstacles to improving $1-1/\mathrm{e}$

We show here that several natural approaches to improve the factor of $1-1/\mathrm{e}$ cannot succeed.

Example 1.9. Consider n^n items arranged in an n-dimensional cube, $Q_n = \{1,2,\ldots,n\}^n$. For a vector $\vec{y} \in Q_n$, let $\mathcal{F}_i(\vec{y})$ denote the "i-fiber" of $\vec{y}$ as the set of elements coinciding with $\vec{y}$ in all coordinates $j \neq i$:

$$\mathcal{F}_i(\vec{y}) = \{\vec{x} \in Q_n \mid \forall j \neq i; x_j = y_j\}.$$

The goal of player i is to obtain at least one item from each i-fiber. We define her utility function as

$$w_i(S) = \Pr_{\vec{y}}[\mathcal{F}_i(\vec{y}) \cap S \neq \emptyset]$$

where $\vec{y} \in Q_n$ is a uniformly random element, i. e., $\mathcal{F}_i(\vec{y})$ is a uniformly random i-fiber. This is a monotone submodular function, being the probability measure of a union of events $[j \in \mathcal{F}_i(\vec{y})]$ over $j \in S$.

One optimal fractional solution can be defined as follows. Each player i selects a combination of the layers orthogonal to dimension i,

$$H_{i,j} = \{\vec{x} \in Q_n : x_i = j\}\,.$$

Each of these sets has value $w_i(H_{i,j}) = 1$, because it intersects every i-fiber. In our optimal fractional solution, player i receives each set $H_{i,j}$ with fractional weight $x_{i,H_{i,j}} = 1/n$, for a value of 1.

Consider the Simple Rounding Procedure. It allocates each item independently and uniformly to a random player. For player i, the probability that she receives some item in a fixed i-fiber is $1-(1-1/n)^n$. Averaging over all fibers, the utility of each player is $1-(1-1/n)^n$.

However, the actual optimum gives value 1 to each player. This can be achieved by a "chessboard pattern" where item $\vec{x} \in Q_n$ is allocated to player $p(\vec{x}) = \sum_{i=1}^n x_i \bmod n$. So our one-step Simple Rounding Procedure gets only $1-(1-1/n)^n$ of the optimum.

As an alternative approach, consider a two-step rounding procedure (e. g., the Fair Rounding Procedure). Here, each player chooses a random set S with probability $x_{i,S}$. Then, conflicts between players are resolved somehow and finally, the remaining items are allocated. For our fractional solution, this would mean that each player chooses $H_{i,j}$ for a random j; we can assume WLOG that she chooses $H_{i,1}$. Regardless of how we resolve conflicts, only $n^n-(n-1)^n = (1-(1-1/n)^n)n^n$ items are allocated at this point, so we cannot get more value than $1-(1-1/n)^n$ per player. But the situation is even worse that this. We still have $(n-1)^n$ unallocated items, but regardless of how we assign them to players, we do not gain any additional value at all. This is because for any of these $(n-1)^n$ remaining items $\vec{y}$ and any player i, the fiber $\mathcal{F}_i(\vec{y})$ already has an item in the first layer which was allocated to player i and hence player i is not interested in any more items from this fiber. Thus again, this approach cannot achieve more than $1-(1-1/n)^n$ of the optimum. Instead, we have to return to the first step and redesign the rounding procedure in a different way.

2 The allocation problem for two players

In this section, we start working towards the goal of proving Theorem 1.2. First we consider a special case of the welfare maximization problem where only two players are interested in a given set of m items. The analysis of this case is not formally needed for the proof of Theorem 1.2 but we consider it instructive for the reader to understand this simpler case before proceeding to the proof of Theorem 1.2.

As we have shown in the previous section, in the setting with two players we can achieve an approximation factor of $3/4$ (even assuming only fractional subadditivity). Since our improvements are built on top of the $3/4$-approximation algorithm, let us present this special case first.

2.1 $3/4$-approximation for two players

The basic idea is to use the fractional LP solution to generate random sets suitable for each player. When we say that "player 1 samples a random set from his distribution," it means that he chooses set S with probability $x_{1,S}$. (Since $\sum_S x_{1,S} \le 1$, this is a valid probability distribution.) Similarly, player 2 samples a random set T from her distribution defined by $x_{2,T}$. We define

- $p_j = \Pr[j \in S] = \sum_{S:j\in S} x_{1,S}$.

- $q_j = \Pr[j \in T] = \sum_{T:j\in T} x_{2,T}$.

Ideally, we would like to assign S to player 1 and T to player 2 which would yield an expected value equal to the LP optimum. The only issue is that the sets S and T can overlap, so we cannot satisfy the players' requests exactly. One way to allocate the disputed items is by making a random decision. It turns out that the best way to do this is to allocate disputed items to one of the two players with *reversed probabilities* compared to the fractional solution (see [7]; also, this can be seen as a special case of the Fair Contention Resolution procedure, see Section 1.2). We do this using a random "splitting set" X which contains each item j with probability p_j.

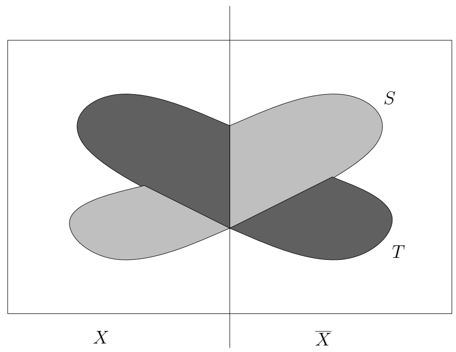

Figure 1: Algorithm 1.

Algorithm 1 (3/4-approximation for 2 players)

- Let player 1 sample a random set S and let player 2 sample a random set T from their respective distributions.
- Independently, generate a random set X which contains item j with probability p_j.
- Assign $S \setminus T$ to player 1.
- Assign $T \setminus S$ to player 2.
- Divide $S \cap T$ into $S \cap T \setminus X$ for player 1 and $S \cap T \cap X$ for player 2.

Now consider Algorithm 1 from the point of view of player 1, conditioned on a specific choice of S. He receives each element of S, unless it also appears in $T \cap X$. This set is sampled independently of S and

$$\Pr[j \in T \cap X] = q_j p_j \leq \frac{1}{4}$$

because $p_j + q_j \leq 1$. Therefore, conditioned on S, each element is taken away with probability at most $1/4$. By Lemma 1.7,

$$\mathbf{E}[w_1(S \setminus (T \cap X)) \mid S] \geq \frac{3}{4} w_1(S)\,.$$

Taking the expectation over S, we get

$$\mathbf{E}[w_1(S\setminus(T\cap X))] \geq \frac{3}{4}\mathbf{E}[w_1(S)] = \frac{3}{4}\sum_S x_{1,S}w_1(S)\,.$$

Similarly, player 2 gets at least $(3/4)\sum_T x_{2,T}w_2(T)$, since any element appears in $S\cap\overline{X}$ with probability $p_j(1-p_j)\leq 1/4$. This shows that in expectation, we not only recover at least $3/4$ of the optimum, but each player individually obtains at least $3/4$ of his share in the LP.

2.2 Examples and integrality gaps

The $3/4$-approximation algorithm for two players is optimal for fractionally subadditive utility functions, in the sense that our LP can have an integrality gap equal to $3/4$. The proof is a simple example with 4 items which we present here. As shown in [7], the class of fractionally subadditive functions is equal to "XOS," the class of functions obtained as a maximum of several linear functions. We use this property here to define our example.

Example 2.1 (3/4 integrality gap for two players with XOS functions)**.**

	T_1	T_2
S_1	a	b
S_2	c	d

Consider 4 items $\{a,b,c,d\}$ partitioned in two ways: $S_1=\{a,b\},S_2=\{c,d\}$ and $T_1=\{a,c\},T_2=\{b,d\}$. For a set of items $A\subseteq\{a,b,c,d\}$, we define two utility functions by

- $w_1(A)=\max\{|A\cap S_1|,|A\cap S_2|\}$
- $w_2(A)=\max\{|A\cap T_1|,|A\cap T_2|\}$.

In other words, S_1,S_2 are the sets desired by player 1 and T_1,T_2 are the sets desired by player 2. The optimal LP solution is

$$x_{1,S_1}=x_{1,S_2}=x_{2,T_1}=x_{2,T_2}=\frac{1}{2}$$

which makes each player maximally happy and yields a total value of $LP=4$. On the other hand, there is no integral solution of value 4. Such a solution would require two disjoint sets S_i and T_j of value 2 but no such pair of disjoint sets exists. The optimum integral solution has value $3=3/4\cdot LP$.

The question arises whether $3/4$ is also optimal for submodular functions. It can be seen easily that the utility functions above are not submodular—for example $w_1(\{a,c\})+w_1(\{b,c\})=2$ but $w_1(\{c\})+w_1(\{a,b,c\})=3$. The easiest way to make the functions submodular is to increase the value of the diagonal sets $\{b,c\}$ and $\{a,d\}$ to 2.

Example 2.2 (two players with submodular functions)**.** Consider the items as above, where we also define $Y=\{a,d\}$ and $Z=\{b,c\}$. Each singleton has value 1 and any set of at least 3 elements has value 2. For pairs of items, define the utility function of player 1 as follows:

$w_1(Y) = w_1(Z) = 2$	$w_1(T_1) = 1$	$w_1(T_2) = 1$
$w_1(S_1) = 2$	a	b
$w_1(S_2) = 2$	c	d

In other words, player 1 wants at least one of items $\{a,c\}$ and at least one of $\{b,d\}$. Symmetrically, player 2 wants at least one of items $\{a,b\}$ and at least one of $\{c,d\}$. Her utility function is $w_2(S_1) = w_2(S_2) = 1$, $w_2(Y) = w_2(Z) = 2$ and $w_2(T_1) = w_2(T_2) = 2$. The functions w_1 and w_2 can be verified to be submodular (being the rank functions of partition matroids). As before, a fractional solution assigning each of the sets S_1, S_2, T_1, T_2 with weight $1/2$ has value $LP = 4$. However, here the integrality gap is equal to 1, since there is an integral solution (Y,Z) of value 4 as well.

This example illustrates a different phenomenon: Any optimal solution *must combine* items from the two sets desired by each player. If we allocate one set from the fractional solution to each player and resolve conflicts arbitrarily, we get only a value of $(3/4)LP$. Moreover, even allocating the remaining item does not help. Regardless of who gets the last item, the objective value is still only $(3/4)LP$. Therefore, we must combine different sets and the only optimal solution uses the two diagonals.

Observe that instead of increasing the value of the diagonals, we could have increased the value of the orthogonal sets (T_1, T_2 for player 1; S_1, S_2 for player 2). This produces submodular functions as well and again the integrality gap is 1. Here, it is enough to allocate for example S_1 to player 1 and the complement S_2 to player 2.

These examples might suggest that there is a chance to recover the full LP value by either taking sets from the fractional solution or "diagonals" as above. However, this is not the case. A linear combination of these two cases gives the following example.

Example 2.3 (5/6 integrality gap for two players with submodular functions)**.** Each singleton has value 1 and any set of at least 3 elements has value 2. For pairs of items, define the utility function of player 1 as follows:

$w_1(Y) = w_1(Z) = \frac{5}{3}$	$w_1(T_1) = \frac{4}{3}$	$w_1(T_2) = \frac{4}{3}$
$w_1(S_1) = 2$	a	b
$w_1(S_2) = 2$	c	d

Symmetrically, the utility function of player 2 is the same except that $w_2(S_1) = w_2(S_2) = 4/3$ and $w_2(T_1) = w_2(T_2) = 2$. This can be verified to be a submodular function.

As before, a fractional solution assigning each of the sets S_1, S_2, T_1, T_2 with weight $1/2$ has value $LP = 4$; whereas, any integral solution such as (S_1, S_2), (Y,Z) or (T_1, T_2) has value at most $10/3 = 5/6 \cdot LP$.

2.3 Two players with a half-integral solution

We have seen that the integrality gap for two players with submodular functions can be $5/6$. We show that an approximation factor of $5/6$ can be achieved in a special case as above, where the optimum fractional solution is *half-integral*: $x_{i,S} \in \{0, 1/2, 1\}$.

If there is a variable $x_{i,S} = 1$, we can assign S to player i and all the remaining items to the other player, in which case we recover the LP value without any loss. Therefore, we can assume that there are sets S_1, S_2, T_1, T_2 such that $x_{1,S_1} = x_{1,S_2} = x_{2,T_1} = x_{2,T_2} = 1/2$. Items that appear only in sets requested by one player can be simply assigned to the respective player, which can only improve the approximation factor. So let us assume that each item appears in two sets assigned to different players. I. e., (S_1, S_2) and (T_1, T_2) are two (typically different) partitions of the set of all items.

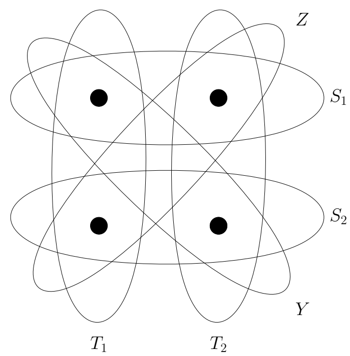

Figure 2: A half-integral solution.

It is necessary to allow the option to combine items from the two sets desired by each player, as described in Example 2.2. Any solution which starts by allocating one set to each player and resolving conflicts is limited by the factor of $3/4$. On the other hand, it is thanks to submodularity that we can extract improved profit by combining two different sets. In analogy with Example 2.2, we define two "diagonal sets"

- $Y = (S_1 \cap T_1) \cup (S_2 \cap T_2)$,
- $Z = (S_1 \cap T_2) \cup (S_2 \cap T_1)$.

Our algorithm then chooses a random allocation scheme according to the following table:

Probability	1/6	1/6	1/6	1/6	1/6	1/6
Player 1	S_1	S_2	Y	Z	T_1	T_2
Player 2	S_2	S_1	Z	Y	T_2	T_1

We use submodularity to analyze the expected profit of this allocation procedure. For player 1, submodularity yields

$$w_1(Y) + w_1(T_1) \geq w_1(S_1 \cap T_1) + w_1(S_2 \cup T_1)$$

and

$$w_1(Z) + w_1(T_2) \geq w_1(S_1 \cap T_2) + w_1(S_2 \cup T_2)\,.$$

Intuitively, T_1 and T_2 are not the sets desired by player 1 and their values could be as low as $w_1(S_1 \cap T_1)$ and $w_1(S_1 \cap T_2)$. However, if that is the case, submodularity implies that the diagonal sets Y, Z are very desirable for player 1.

Since $(S_1 \cap T_1) \cup (S_1 \cap T_2) = S_1$, we have $w_1(S_1 \cap T_1) + w_1(S_1 \cap T_2) \geq w_1(S_1)$, and by monotonicity $w_1(S_2 \cup T_j) \geq w_1(S_2)$. In total, player 1 gets expected profit

$$\frac{1}{6}(w_1(S_1) + w_1(S_2) + w_1(Y) + w_1(Z) + w_1(T_1) + w_1(T_2)) \geq \frac{1}{6}(2w_1(S_1) + 3w_1(S_2)))\,.$$

Alternatively, we can also estimate $w_1(Y) + w_1(T_2) \geq w_1(S_2 \cap T_2) + w_1(S_1 \cup T_2)$ and $w_1(Z) + w_1(T_1) \geq w_1(S_2 \cap T_1) + w_1(S_1 \cup T_1)$ which yields

$$\frac{1}{6}(w_1(S_1) + w_1(S_2) + w_1(Y) + w_1(Z) + w_1(T_1) + w_1(T_2)) \geq \frac{1}{6}(3w_1(S_1) + 2w_1(S_2)))\,.$$

By averaging the two estimates, player 1 gets expected profit at least $(5/12)(w_1(S_1) + w_1(S_2))$. We get a similar estimate for player 2, so the expected value of our integral solution is at least $(5/6)LP$.

2.4 Two players with a balanced fractional solution

Our goal now is to generalize the ideas of the previous section in order to improve the approximation factor of $3/4$ for two players in general. First, we relax the condition that the fractional solution is half-integral. Instead, we assume that for any item j,

$$\sum_{S:j \in S} x_{1,S} = \sum_{T:j \in T} x_{2,T} = \frac{1}{2}\,.$$

We call such a fractional solution *balanced*. In particular, we can make this assumption in case both players have the same utility function, since then any solution $x_{i,S}$ can be replaced by $\tilde{x}_{1,S} = \tilde{x}_{2,S} = (x_{1,S} + x_{2,S})/2$ without change of value.

Intuition Suppose that each player gets value 1 in the fractional solution. Let S denote a random set sampled from the distribution of player 1 and T a random set sampled from the distribution of player 2. Due to our assumption of balance, $\Pr[j \in S] = \Pr[j \in T] = 1/2$ for any item j. We can try to allocate S to player 1, and the entire complement of S to player 2. Then player 1 gets expected value 1, while player 2 obtains (using Lemma 1.7) $\mathbf{E}[w_2(\overline{S})] \geq \mathbf{E}[w_2(T \cap \overline{S})] \geq 1/2$, but this might be tight and we still don't achieve more than $3/4$ of the fractional value.

However, this is essentially the only case in which we do not gain compared to $3/4$. Assuming that the complement of S is not very valuable for player 2, let us consider another set: $Z = (T \cap S) \cup (T' \cap \overline{S})$ where T, T' are sets sampled independently from the distribution of player 2. (This is analogous to one of the diagonal sets in the previous section.) Linearity of expectation allows us to use submodularity for expected values of random sets just like for values of deterministic sets:

$$\mathbf{E}[w_2(Z)] + \mathbf{E}[w_2(\overline{S})] \geq \mathbf{E}[w_2(Z \cup \overline{S})] + \mathbf{E}[w_2(Z \cap \overline{S})] \geq \mathbf{E}[w_2(T)] + \mathbf{E}[w_2(T' \cap \overline{S})]\,.$$

In other words, if $\overline{S}$ presents no improvement on the average over $T' \cap \overline{S}$, then Z is a set as good as T which is exactly what player 2 would desire. Similarly, let player 1 generate two independent sets S, S'. If he does not gain by taking $\overline{T}$ instead of $S' \cap \overline{T}$, then $Y = (S \cap T) \cup (S' \cap \overline{T})$ is by submodularity just as good as S or S'. Note that each player uses the other player's set to "combine" two of her sets. Let us be more specific and define which of the other player's sets is used for this purpose:

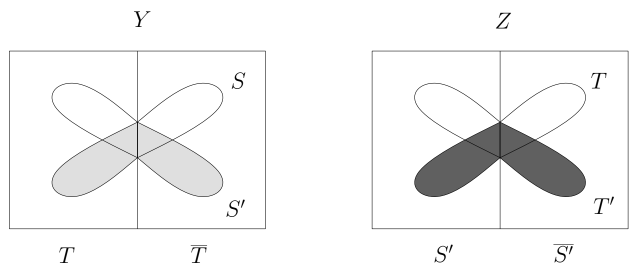

Figure 3: Diagonal sets Y and Z.

- $Y = (S \cap T) \cup (S' \cap \overline{T})$, $\quad Z = (T \cap S') \cup (T' \cap \overline{S'})$.

Note that unlike the diagonal sets in the previous section, Y and Z are not disjoint here. They are random sets, typically intersecting; however, the punch line is that Y and Z are not independent, and in fact the events $j \in Y, j \in Z$ are *negatively correlated*. Observe that Z intersects Y only inside the first part $(S \cap T)$, and more precisely

$$Y \cap Z = (S \cap T) \cap (S' \cup (T' \cap \overline{S'})) = (S \cap T) \cap (S' \cup T').$$

The sets S, S', T, T' are sampled independently and contain each item with probability $1/2$. Similarly, $\Pr[j \in Y] = \Pr[j \in Z] = 1/2$ for any item j, while

$$\Pr[j \in Y \cap Z] = \Pr[j \in (S \cap T) \cap (S' \cup T')] = \frac{3}{16}$$

rather than $1/4$ which is the probability of appearance in $S \cap T$. Thus the interests of the two players are closer to being disjoint and we are able to allocate more items to each of them, using Y and Z. Our next algorithm takes advantage of this fact, combining the two allocation schemes outlined above.

Algorithm 2 (37/48-approximation for 2 balanced players)

- Let player 1 sample independently random sets S, S'.
- Let player 2 sample independently random sets T, T'.
- Let X contain each item independently with probability $1/2$.
- Let $Y = (S \cap T) \cup (S' \cap \overline{T})$, $\ Z = (T \cap S') \cup (T' \cap \overline{S'})$.

We assign items randomly based on the following table:

Probability	$1/3$	$1/3$	$1/3$
Player 1	S'	$\overline{T}$	$Y \setminus (Z \cap X)$
Player 2	$\overline{S'}$	T	$Z \setminus (Y \cap \overline{X})$

Theorem 2.4. *For any balanced fractional solution $x_{i,S}$, Algorithm 2 gives expected profit at least $37/48 \sum_S x_{i,S}\, w_i(S)$ to each player i.*

Proof. Consider player 1. The sets S and S' are sampled from the same distribution. $\mathbf{E}[w_1(S)] = \mathbf{E}[w_1(S')] = \sum_S x_{1,S} w_1(S)$ is the share of player 1 in the fractional solution, ideally what we would like player 1 to receive. For the sets allocated to him in the second and third scheme, $\overline{T}$ and $Y \setminus (Z \cap X)$, we use submodularity. We use the fact that $Y \cap Z = (S \cap T) \cap (S' \cup T')$, i. e.,

$$Y \setminus (Z \cap X) = ((S \cap T) \setminus ((S' \cup T') \cap X)) \cup (S' \cap \overline{T}).$$

By submodularity,

$$\begin{aligned} w_1(\overline{T}) + w_1(Y \setminus (Z \cap X)) &\geq w_1(((S \cap T) \setminus ((S' \cup T') \cap X)) \cup \overline{T}) + w_1(S' \cap \overline{T}) \\ &\geq w_1(S \setminus ((S' \cup T') \cap X \cap T)) + w_1(S' \cap \overline{T}). \end{aligned}$$

Now we use the linearity of expectation and Lemma 1.7: each item appears in $(S' \cup T') \cap X \cap T$ with probability $3/16$, and in $\overline{T}$ with probability $1/2$. Therefore

$$\mathbf{E}[w_1(\overline{T})] + \mathbf{E}[w_1(Y \setminus (Z \cap X))] \geq \frac{13}{16}\mathbf{E}[w_1(S)] + \frac{1}{2}\mathbf{E}[w_1(S')] = \frac{21}{16}\mathbf{E}[w_1(S)].$$

The expected profit of player 1 is

$$\frac{1}{3}\mathbf{E}[w_1(S)] + \frac{1}{3}\mathbf{E}[w_1(\overline{T})] + \frac{1}{3}\mathbf{E}[w_1(Y \setminus (Z \cap X))] \geq \frac{37}{48}\mathbf{E}[w_1(S)] = \frac{37}{48}\sum_S x_{i,S}\, w_1(S).$$

For player 2, the analysis is similar, although not exactly symmetric: here, we have

$$Z \setminus (Y \cap \overline{X}) = (T \cap S' \setminus (S \cap \overline{X})) \cup (T' \cap \overline{S'} \setminus (S \cap T \cap \overline{X})).$$

Submodularity yields

$$\begin{aligned} w_2(\overline{S'}) + w_2(Z \setminus (Y \cap \overline{X})) &\geq w_2((T \cap S' \setminus (S \cap \overline{X})) \cup \overline{S'}) + w_2(T' \cap \overline{S'} \setminus (S \cap T \cap \overline{X})) \\ &\geq w_2(T \setminus (S' \cap S \cap \overline{X})) + w_2(T' \cap \overline{S'} \setminus (S \cap T \cap \overline{X})). \end{aligned}$$

Finally, we apply Lemma 1.7:

$$\mathbf{E}[w_2(\overline{S'})] + \mathbf{E}[w_2(Z \setminus (Y \cap \overline{X}))] \geq \frac{7}{8}\mathbf{E}[w_2(T)] + \frac{7}{16}\mathbf{E}[w_2(T')] = \frac{21}{16}\mathbf{E}[w_2(T)]$$

and the rest follows as for player 1. □

We remark that with a slightly more involved rounding scheme, it is possible to achieve a 7/9-approximation in the case of two balanced players. Details can be found in [18].

2.5 Two players with an arbitrary fractional solution

Given our algorithm for balanced fractional solutions, it seems plausible that we should be able to obtain an improvement in the general case as well. This is because Algorithm 1 gives a better approximation than $3/4$ if the fractional solution is unbalanced. However, a fractional solution can be balanced on some items and unbalanced on others, so we have to analyze our profit more carefully, item by item. For this purpose, we prove the following generalization of Lemma 1.7.

Lemma 2.5. *Fix an ordering of items $[m] = \{1,2,\ldots,m\}$; we denote by $[j] = \{1,2,\ldots,j\}$ the first j items in this ordering. Let S and X be random subsets of $[m]$ such that for any S_1 and any j, $\Pr[j \in X \mid S = S_1] \geq p_j$. Let w be a monotone submodular function and define*

$$\sigma_j = \mathbf{E}[w(S \cap [j]) - w(S \cap [j-1])].$$

Then

$$\mathbf{E}[w(S \cap X)] \geq \sum_{j=1}^{m} p_j \sigma_j.$$

Proof. Using the marginal-value definition of submodularity, we obtain

$$\begin{aligned} w(S \cap X) &= \sum_{j=1}^{m} (w(S \cap X \cap [j]) - w(S \cap X \cap [j-1])) \\ &\geq \sum_{j=1}^{m} \mathbf{1}_{j \in X} (w(S \cap [j]) - w(S \cap [j-1])) . \end{aligned}$$

Conditioned on $S = S_1$, j appears in X with probability at least p_j, so taking expectation over X yields

$$\mathbf{E}_X[w(S \cap X) \mid S = S_1] \geq \sum_{j=1}^{m} p_j (w(S_1 \cap [j]) - w(S_1 \cap [j-1]))$$

and finally

$$\mathbf{E}[w(S \cap X)] \geq \sum_{j=1}^{m} p_j \mathbf{E}[w(S \cap [j]) - w(S \cap [j-1])] = \sum_{j=1}^{m} p_j \sigma_j .$$

$\square$

In the following, we assume that S is a set sampled by player 1, T is a set sampled by player 2, and we set

- $p_j = \Pr[j \in S]$, $\sigma_j = \mathbf{E}[w_1(S \cap [j]) - w_1(S \cap [j-1])]$,
- $q_j = \Pr[j \in T]$, $\tau_j = \mathbf{E}[w_2(T \cap [j]) - w_2(T \cap [j-1])]$.

We seek to estimate our profit in terms of $\mathbf{E}[w_1(S)] = \sum_j \sigma_j$ and $\mathbf{E}[w_2(T)] = \sum_j \tau_j$ which are the shares of the two players in the fractional solution.

Let us give a sketch of an argument that a strict improvement over $3/4$ is possible in the general case. Let us choose a very small constant $\varepsilon > 0$ and define "unbalanced items" by $U = \{j : |p_j - 1/2| > \varepsilon\}$. We distinguish two cases:

- If a non-negligible value comes from unbalanced items, e. g., $\sum_{j\in U}(\sigma_j+\tau_j)\geq\varepsilon\cdot LP$, then we use Algorithm 1. A refined analysis using Lemma 2.5 implies than the algorithm yields profit at least

$$\begin{aligned}\sum_j(1-p_j(1-p_j))\sigma_j+\sum_j(1-p_j(1-p_j))\tau_j &\geq \frac{3}{4}LP+\sum_{j\in U}\Big(\frac{1}{4}-p_j(1-p_j)\Big)(\sigma_j+\tau_j)\\ &\geq\Big(\frac{3}{4}+\varepsilon^3\Big)LP.\end{aligned}$$

- If the contribution of unbalanced items is negligible, $\sum_{j\in U}(\sigma_j+\tau_j)<\varepsilon\cdot LP$, then let us remove the unbalanced items. Also, we scale the remaining fractional solution by $1/(1+2\varepsilon)$ and possibly extend some sets so that we get a balanced solution with $p_j=q_j=1/2$. We incur at most a factor of $(1-3\varepsilon)$ compared to the original fractional solution. Then we run Algorithm 2 on this balanced fractional solution which yields expected value at least $(37/48)(1-3\varepsilon)\cdot LP$.

Choosing for example $\varepsilon=1/150$ gives an approximation factor slightly better than $3/4$. A more careful combination of the balanced and unbalanced case yields a $13/17$-approximation for 2 players in the general case but we omit the details here. The algorithm and its analysis can be found in [18].

3 The allocation problem for n players

As we discussed, our final algorithm will use a combination of two rounding techniques. The first technique is the Fair Rounding Procedure that we presented in Section 1.3.

3.1 The fair rounding procedure

Algorithm 3 (Fair Rounding Procedure)

- Let each player i sample a set S_i from her probability distribution.
- Using the Fair Contention Resolution technique (below), resolve contention independently for each item, and allocate each item to the respective winner.

Fair contention resolution Suppose n players compete for an item independently with probabilities $p_1,p_2,\dots,p_n$. Denote by A the random set of players who request the item, i. e., $\Pr[i\in A]=p_i$ independently for each i.

- If $A=\emptyset$, do not allocate the item.
- If $A=\{k\}$, allocate the item to player k.
- If $|A|>1$, allocate the item to each $k\in A$ with probability

$$r_{A,k}=\frac{1}{\sum_{i=1}^n p_i}\left(\sum_{i\in A\setminus\{k\}}\frac{p_i}{|A|-1}+\sum_{i\notin A}\frac{p_i}{|A|}\right).$$

Note that $r_{A,k} \geq 0$ and $\sum_{k \in A} r_{A,k} = 1$ so this is a valid probability distribution. The formula has the property that conditioned on a particular set of players A competing for the item, some preference is given to players with smaller probabilities p_i. This is necessary since these players have to compete more often with other players, and they would be at a disadvantage if we chose among the players in A uniformly. (Compare with the case of 2 players in Section 2.)

Here we analyze the Fair Contention Resolution technique and prove Lemma 1.5. Let us restate it here.

Lemma 3.1. *Conditioned on player k requesting the item, she obtains it with probability exactly*

$$\rho = \frac{1 - \prod_{i=1}^{n}(1-p_i)}{\sum_{i=1}^{n} p_i}.$$

Proof. First, suppose that we allocate the item to player k with probability $r_{A,k}$ for any A containing k, even $A = \{k\}$. (For the sake of the proof, we interpret the sum over $A \setminus \{k\}$ for $A = \{k\}$ as zero, although the summand is undefined.) Then the conditional probability that player k receives the item, when she competes for it, would be $\mathbf{E}[r_{A,k} \mid k \in A]$.

However, when $A = \{k\}$, our technique actually allocates the item to player k with probability 1, rather than

$$r_{\{k\},k} = \frac{\sum_{i \neq k} p_i}{\sum_{i=1}^{n} p_i} = 1 - \frac{p_k}{\sum_{i=1}^{n} p_i}.$$

So player k gains an additional probability

$$\Pr[A = \{k\}](1 - r_{\{k\},k}) = \Pr[A = \{k\}] \cdot \frac{p_k}{\sum_i p_i}$$

which makes the total probability that player k obtains the item equal to

$$q_k = p_k \, \mathbf{E}[r_{A,k} \mid k \in A] + \frac{p_k}{\sum_{i=1}^{n} p_i} \Pr[A = \{k\}]. \tag{3.1}$$

We would like to show that

$$q_k = \frac{p_k}{\sum p_i} \Pr[A \neq \emptyset].$$

This means that $\mathbf{E}[r_{A,k} \mid k \in A]$ should be equal to

$$\frac{1}{\sum p_i} \Pr[A \setminus \{k\} \neq \emptyset].$$

Let us define $B = [n] \setminus \{k\}$ and let $A' = A \setminus \{k\}$ be the set of players competing with k. The probability of a particular set A' occurring is $p(A') = \prod_{i \in A'} p_i \prod_{i \in B \setminus A'}(1-p_i)$. Let us write $\mathbf{E}[r_{A,k} \mid k \in A]$ as a weighted sum over all possible subsets $A' \subseteq B$:

$$\begin{aligned}
\mathbf{E}[r_{A,k} \mid k \in A] &= \sum_{A' \subseteq B} p(A') \; r_{A' \cup \{k\},k} \\
&= \frac{1}{\sum_{i=1}^{n} p_i} \sum_{A' \subseteq B} p(A') \left(\sum_{i \in A'} \frac{p_i}{|A'|} + \sum_{i \in B \setminus A'} \frac{p_i}{|A'|+1} \right).
\end{aligned}$$

Ideally, we would like to see

$$\frac{1}{\sum p_i} \sum_{A' \neq \emptyset} p(A') = \frac{1}{\sum p_i} \Pr[A' \neq \emptyset]$$

instead, but we have to perform a certain redistribution of terms to achieve this. Observe that for i, A' such that $i \in B \setminus A'$, the contribution can be also written as

$$p(A') \frac{p_i}{|A'|+1} = \frac{1-p_i}{p_i} p(A' \cup \{i\}) \frac{p_i}{|A'|+1} = p(A' \cup \{i\}) \frac{1-p_i}{|A' \cup \{i\}|}.$$

Using this equality to replace all the terms for $i \in B \setminus A'$, we get

$$\begin{aligned}
\mathbf{E}[r_{A,k} \mid k \in A] &= \frac{1}{\sum_{i=1}^n p_i} \left(\sum_{A' \subseteq B} p(A') \sum_{i \in A'} \frac{p_i}{|A'|} + \sum_{A' \subseteq B} \sum_{i \in B \setminus A'} p(A' \cup \{i\}) \frac{1-p_i}{|A' \cup \{i\}|} \right) \\
&= \frac{1}{\sum_{i=1}^n p_i} \left(\sum_{A' \subseteq B} p(A') \sum_{i \in A'} \frac{p_i}{|A'|} + \sum_{\emptyset \neq A'' \subseteq B} \sum_{i \in A''} p(A'') \frac{1-p_i}{|A''|} \right) \\
&= \frac{1}{\sum_{i=1}^n p_i} \sum_{\emptyset \neq A' \subseteq B} p(A') \sum_{i \in A'} \left(\frac{p_i}{|A'|} + \frac{1-p_i}{|A'|} \right) \\
&= \frac{1}{\sum_{i=1}^n p_i} \sum_{\emptyset \neq A' \subseteq B} p(A') = \frac{1}{\sum_{i=1}^n p_i} \Pr[A \setminus \{k\} \neq \emptyset].
\end{aligned}$$

So indeed, from (3.1), player k receives the item with probability

$$q_k = \frac{p_k}{\sum_{i=1}^n p_i} \big(\Pr[A \setminus \{k\} \neq \emptyset] + \Pr[A = \{k\}] \big) = \frac{p_k}{\sum_{i=1}^n p_i} \Pr[A \neq \emptyset].$$

□

As we showed in Lemma 1.8, this already proves that the Fair Rounding Procedure achieves a $(1-1/e)$-approximation. In fact, the Fair Rounding Procedure achieves a factor better than $1-1/e$ whenever the fractional solution is *unbalanced*, similarly to the case of two players. Therefore, the most difficult case to deal with is when the fractional solution is *balanced*.

3.2 The butterfly rounding technique

Let us assume for now that the fractional solution is balanced in the sense that for each player i and item j, we have $y_{ij} = \sum_{S: j \in S} x_{i,S} = 1/n$, which is the worst case for the Fair Rounding Procedure.[3] We also assume that the number of players n is very large; otherwise, we have an improvement over $1-1/e$ already. In other words, we consider the variables y_{ij} infinitesimal and we write $(1-1/n)^n \simeq e^{-1}$, $(1-1/n)^{n/2} \simeq e^{-1/2}$, etc.

In this case, we use ideas from the two-player case in order to obtain a small improvement over $1-1/e$. Roughly speaking, we divide the players into two groups and treat them as two super-players,

[3]There is always a balanced optimal LP solution when all players have the same utility function. Hence the results in this section can be applied directly to this special case.

using our algorithm for two players. The items obtained by the super-players are allocated within each group. We would have liked to employ Algorithm 2 as a black box for the two super-players, but the additional complication of conflicts inside each group forces us to modify the previous algorithms slightly.

Let us assume that we can split the players evenly into two groups $\mathcal{A},\mathcal{B}$ such that for each item j,

$$\sum_{i\in\mathcal{A}} y_{ij} = \sum_{i\in\mathcal{B}} y_{ij} = \frac{1}{2}\,.$$

For a collection of sets $\{S_i : i \in \mathcal{A}\}$ sampled by players in one group, we will use Fair Contention Resolution to make the sets disjoint. Recall that players in a group $\mathcal{A}$ such that $\sum_{i\in\mathcal{A}} y_{ij} = 1/2$ can resolve contention in such a way that each requested item is allocated with probability

$$\frac{1-\prod_{i\in\mathcal{A}}(1-y_{ij})}{\sum_{i\in\mathcal{A}} y_{ij}} \geq 2(1-e^{-1/2}) \simeq 0.787\,.$$

This is significantly better than $1-e^{-1} \simeq 0.632$; however, 0.787 is not the approximation factor we can achieve. First we have to distribute items between the two groups, and for this purpose we use ideas inspired by the two-player case. In the end, we recover roughly 0.645 of the LP value for each player.

Algorithm 4 (The Butterfly Rounding Procedure)

- Let each player in group $\mathcal{A}$ sample two independent random sets S_i, S'_i.
- Let each player in group $\mathcal{B}$ sample two independent random sets T_i, T'_i.
- Let $U = \bigcup_{i\in\mathcal{A}} S_i,\;\; U' = \bigcup_{i\in\mathcal{A}} S'_i,\;\; V = \bigcup_{i\in\mathcal{B}} T_i,\;\; V' = \bigcup_{i\in\mathcal{B}} T'_i$.
- Let the players in $\mathcal{A}$ apply Fair Contention Resolution to sets S_i to obtain disjoint sets $S^*_i \subseteq S_i$. Similarly, let them resolve contention among S'_i to obtain disjoint sets $S'^*_i \subseteq S'_i$.
- Let the players in $\mathcal{B}$ use Fair Contention Resolution among T_i to obtain disjoint sets $T^*_i \subseteq T_i$. Similarly, let them resolve contention among T'_i to obtain disjoint sets $T'^*_i \subseteq T'_i$.
- Let $Y^*_i = (S^*_i \cap V) \cup (S'^*_i \cap \overline{V}),\;\; Z^*_i = (T^*_i \cap U) \cup (T'^*_i \cap \overline{U})$.
- Let $Y'^*_i = (S^*_i \cap V') \cup (S'^*_i \cap \overline{V'}),\;\; Z'^*_i = (T^*_i \cap U') \cup (T'^*_i \cap \overline{U'})$.

We assign the items using one of four allocation schemes:

1. With probability $e^{1/2}/(1+2e^{1/2})$:
   Each player $i \in \mathcal{A}$ receives S^*_i. Each player $i \in \mathcal{B}$ receives $(T^*_i \setminus U) \cup (T'^*_i \setminus U \setminus V)$.
2. With probability $e^{1/2}/(1+2e^{1/2})$:
   Each player $i \in \mathcal{B}$ receives T^*_i. Each player $i \in \mathcal{A}$ receives $(S^*_i \setminus V) \cup (S'^*_i \setminus V \setminus U)$.
3. With probability $1/(2+4e^{1/2})$:
   Each player $i \in \mathcal{A}$ receives Y'^*_i. Each player $i \in \mathcal{B}$ receives $Z^*_i \setminus \bigcup_{i\in\mathcal{A}} Y'^*_i$.

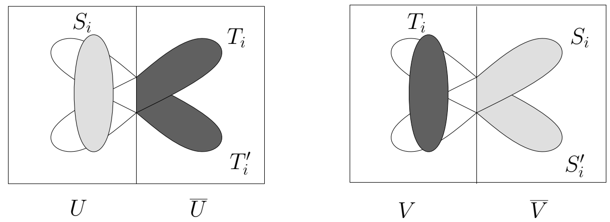

Figure 4: Allocation schemes 1 and 2.

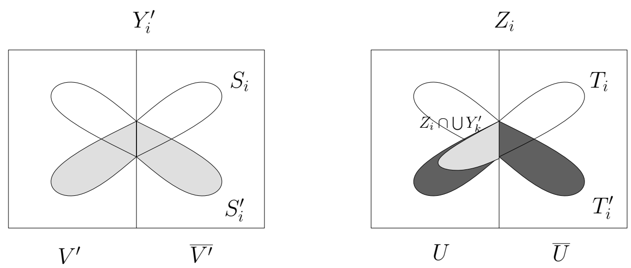

Figure 5: Allocation scheme 3.

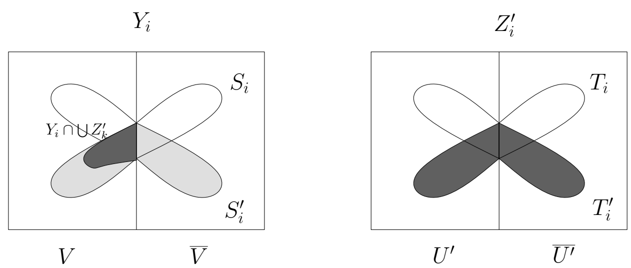

Figure 6: Allocation scheme 4.

4. With probability $1/(2+4e^{1/2})$:
   Each player $i \in \mathcal{B}$ receives $Z_i'^*$. Each player $i \in \mathcal{A}$ receives $Y_i^* \setminus \bigcup_{i\in\mathcal{B}} Z_i'^*$.

See the figures depicting the four allocation schemes (without considering contention inside each group of players). In the following, we also use $Y_i = (S_i \cap V) \cup (S_i' \cap \overline{V})$, $Y_i' = (S_i \cap V') \cup (S_i' \cap \overline{V'})$, etc., to denote the "diagonal sets" before resolving contention. Taking the union over each group, we get the same set regardless of resolving contention or not:

$$\bigcup_{i\in\mathcal{A}} S_i = \bigcup_{i\in\mathcal{A}} S_i^* = U\,, \quad \bigcup_{i\in\mathcal{A}} Y_i = \bigcup_{i\in\mathcal{A}} Y_i^* = (U \cap V) \cup (U' \cap \overline{V})\,, \quad \text{etc.}$$

Intuition Note that in a set like S_i or T_i, each element appears with probability $1/n$. The sets U, U', V, V' are formed by taking a union of $n/2$ such independent sets, and hence contain each element with probability $1-(1-1/n)^{n/2} \simeq 1-e^{-1/2}$. If we use only the first two schemes, it can be shown that we would get an approximation factor at least $1-1/e$. In fact, we get $1-1/e$ even without using the sets S'_i, T'_i - if each player chooses one set and we resolve contention first in one preferred group and then in the second group, we get exactly $1-1/e$ (see the proof below). Adding some elements of T'_i to a player $i \in \mathcal{B}$ and some elements of S'_i to a player $i \in \mathcal{A}$ might or might not help—but if it doesn't help, we prove that we can construct other good sets Y_i, Y'_i for player $i \in \mathcal{A}$ and Z_i, Z'_i for $i \in \mathcal{B}$, which have the property of negative correlation (we repeat the trick of Algorithm 2). Then we can extract more than $1-1/e$ of their value for each player.

Theorem 3.2. *For n players with a balanced fractional solution and $n \to \infty$, Algorithm 4 yields expected profit at least $0.645 \sum_S x_{i,S} w_i(S)$ for player i.*

Proof. We use the following notation:

- For $i \in \mathcal{A}$, $\alpha_i = \mathbf{E}[w_i(S_i)]$.
  For $i \in \mathcal{B}$, $\beta_i = \mathbf{E}[w_i(T_i)]$.
- For $i \in \mathcal{A}$, $\gamma_i = \mathbf{E}[w_i((S_i \cup S'_i) \setminus V) - w_i(S_i \setminus V)]$.
  For $i \in \mathcal{B}$, $\delta_i = \mathbf{E}[w_i((T_i \cup T'_i) \setminus U) - w_i(T_i \setminus U)]$.
- For every i, j: $y_{ij} = \sum_{S: j \in S} x_{i,S}$; i. e., $\sum_{i \in \mathcal{A}} y_{ij} = \sum_{i \in \mathcal{B}} y_{ij} = 1/2$.

First, recall that in each group of sets like $\{S_i : i \in \mathcal{A}\}$, Lemma 1.5 allows us to resolve contention in such a way that each item in S_i is retained in S^*_i with conditional probability at least $2(1-e^{-1/2})$. We will postpone this step until the end, which will incur a factor of $2(1-e^{-1/2})$ on the expected value of all allocated sets. Instead, we analyze the sets "requested" by each player, which are formally obtained by removing the stars from the sets appearing in each allocation scheme. Note that some requested sets are formed by combining S_i and S'_i, such as $Y_i = (S_i \cap V) \cup (S'_i \cap \overline{V})$; however, contention resolution for each fixed item requested by player i involves only one of the sets S_i, S'_i.

Consider a player $i \in \mathcal{A}$. In the first allocation scheme, he requests the set S_i of expected value $\mathbf{E}[w_i(S_i)] = \alpha_i$. In the second allocation scheme, he requests $(S_i \setminus V) \cup (S'_i \setminus V \setminus U)$. First, let us just consider $S_i \setminus V$. Each item survives outside of V with probability $(1-1/n)^{n/2} \simeq e^{-1/2}$, therefore

$$\mathbf{E}[w_i(S_i \setminus V)] \geq e^{-1/2} \alpha_i .$$

Observe that at this point, we already have an approximation factor of $1-1/e$: By averaging the two cases, we get

$$\frac{1}{2}\mathbf{E}[w_i(S_i) + w_i(S_i \setminus V)] \geq \frac{1}{2}(1+e^{-1/2})\alpha_i .$$

Player i actually receives each requested item with probability at least $2(1-e^{-1/2})$, so his expected profit is at least

$$2(1-e^{-1/2}) \cdot \frac{1}{2}(1+e^{-1/2})\alpha_i = (1-e^{-1})\alpha_i .$$

However, rather than $S_i \setminus V$, player i requests $(S_i \setminus V) \cup (S'_i \setminus V \setminus U)$. This might yield some gain or not; we would like to express this gain in terms of γ_i. Let us write $\tilde{U} = \bigcup_{k \in \mathcal{A} \setminus \{i\}} S_k$; we can use this instead of $U = \bigcup_{k \in \mathcal{A}} S_k$ here, since

$$(S_i \setminus V) \cup (S'_i \setminus V \setminus \tilde{U}) = (S_i \setminus V) \cup (S'_i \setminus V \setminus U).$$

The way we analyze the contribution of $S'_i \setminus V \setminus \tilde{U}$ is that we look at the marginal value

$$\gamma'_i = \mathbf{E}[w_i((S_i \setminus V) \cup (S'_i \setminus V \setminus \tilde{U})) - w_i(S_i \setminus V)]$$

as opposed to

$$\gamma_i = \mathbf{E}[w_i((S_i \setminus V) \cup (S'_i \setminus V)) - w_i(S_i \setminus V)].$$

Let us fix S_i, V and define

$$g_i(A) = w_i((S_i \setminus V) \cup A) - w_i(S_i \setminus V),$$

the marginal value of a set added to $S_i \setminus V$. This is also a submodular function. We are interested in $g_i(S'_i \setminus V \setminus \tilde{U})$, as opposed to $g_i(S'_i \setminus V)$. Each item present in $S'_i \setminus V$ survives in $S'_i \setminus V \setminus \tilde{U}$ with conditional probability $\prod_{k \in \mathcal{A} \setminus \{i\}} (1 - y_{kj}) \simeq e^{-1/2}$. Since $\tilde{U}$ is sampled independently of S_i, S'_i and V, Lemma 2.5 implies

$$\mathbf{E}_{\tilde{U}}[g_i(S'_i \setminus V \setminus \tilde{U})] \geq e^{-1/2} g_i(S'_i \setminus V) = e^{-1/2}(w_i((S_i \setminus V) \cup (S'_i \setminus V)) - w_i(S_i \setminus V)).$$

Taking expectation over the remaining random sets, we get

$$\gamma'_i = \mathbf{E}[g_i(S'_i \setminus V \setminus \tilde{U})] \geq e^{-1/2} \mathbf{E}[w_i((S_i \setminus V) \cup (S'_i \setminus V)) - w_i(S_i \setminus V)] = e^{-1/2} \gamma_i.$$

To summarize,

$$\mathbf{E}[w_i((S_i \setminus V) \cup (S'_i \setminus V \setminus U))] = \mathbf{E}[w_i(S_i \setminus V)] + \gamma'_i \geq e^{-1/2}(\alpha_i + \gamma_i).$$

Now, let us turn to the third allocation scheme. Player i requests $Y'_i = (S_i \cap V') \cup (S'_i \cap \overline{V'})$ which by submodularity and monotonicity satisfies

$$\begin{aligned}
\mathbf{E}[w_i(Y'_i)] + \mathbf{E}[w_i((S_i \cup S'_i) \cap \overline{V'})] &\geq \mathbf{E}[w_i(Y'_i \cup (S_i \cap \overline{V'}))] + \mathbf{E}[w_i(S'_i \cap \overline{V'})] \\
&\geq \mathbf{E}[w_i(S_i)] + \mathbf{E}[w_i(S'_i \cap \overline{V'})],
\end{aligned}$$

i. e.,

$$\mathbf{E}[w_i(Y'_i)] \geq \mathbf{E}[w_i(S_i)] - (\mathbf{E}[w_i((S_i \cup S'_i) \cap \overline{V'}) - \mathbf{E}[w_i(S'_i \cap \overline{V'})]) = \alpha_i - \gamma_i.$$

Note that either player i gains in the second allocation scheme (when γ_i is large), or otherwise Y'_i has very good expected value, close to α_i.

In the fourth allocation scheme, player i requests $Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k$, where $Y_i = (S_i \cap V) \cup (S'_i \cap \overline{V})$ and $\bigcup_{k \in \mathcal{B}} Z'_k = (V \cap U') \cup (V' \cap \overline{U'})$, and so

$$\begin{aligned}
Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k &= ((S_i \cap V) \cup (S'_i \cap \overline{V})) \setminus ((V \cap U') \cup (V' \cap \overline{U'})) \\
&= ((S_i \cap V) \setminus ((V \cap U') \cup (V' \cap \overline{U'}))) \cup ((S'_i \cap \overline{V}) \setminus ((V \cap U') \cup (V' \cap \overline{U'})) \\
&= ((S_i \cap V) \setminus (U' \cup V')) \cup (S'_i \cap \overline{V})
\end{aligned}$$

since $S_i \cap V$ is contained in V, and $S'_i \cap \overline{V}$ is disjoint from both $V \cap U'$ and $V' \cap \overline{U'}$. By submodularity and monotonicity,

$$\begin{aligned}
& \mathbf{E}[w_i(Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k)] + \mathbf{E}[w_i((S_i \cup S'_i) \cap \overline{V})] \\
= \; & \mathbf{E}[w_i(((S_i \cap V) \setminus (U' \cup V')) \cup (S'_i \cap \overline{V}))] + \mathbf{E}[w_i((S_i \cup S'_i) \cap \overline{V})] \\
\geq \; & \mathbf{E}[w_i(((S_i \cap V) \setminus (U' \cup V')) \cup ((S_i \cup S'_i) \cap \overline{V}))] + \mathbf{E}[w_i(S'_i \cap \overline{V})] \\
\geq \; & \mathbf{E}[w_i(S_i \setminus (V \cap (U' \cup V')))] + \mathbf{E}[w_i(S'_i \cap \overline{V})]
\end{aligned}$$

and therefore, using $\gamma_i = \mathbf{E}[w_i((S_i \cup S'_i) \cap \overline{V})] - \mathbf{E}[w_i(S'_i \cap \overline{V})]$,

$$\begin{aligned}
\mathbf{E}[w_i(Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k)] & \geq \mathbf{E}[w_i(S_i \setminus (V \cap (U' \cup V')))] - \gamma_i \\
& \geq (1 - (1 - e^{-1/2})(1 - e^{-1}))\alpha_i - \gamma_i
\end{aligned}$$

since item j appears in $V \cap (U' \cup V')$ with probability $(1 - e^{-1/2})(1 - e^{-1})$. This makes the profit of player i significantly smaller compared to the third allocation scheme; nonetheless, he does not lose as much as if we removed from him a union of independently sampled sets for players in $\mathcal{B}$ (which would contain each element with probability $(1 - e^{-1/2})$ rather than $(1 - e^{-1/2})(1 - e^{-1})$). Here, we benefit from the negative correlation between sets Y_i and Z'_k.

Finally, contention is resolved within each group which incurs an additional factor of $2(1 - e^{-1/2})$. The overall expected profit of player $i \in \mathcal{A}$ is

$$\begin{aligned}
& \frac{e^{1/2}}{1 + 2e^{1/2}} 2(1 - e^{-1/2}) \left(\alpha_i + e^{-1/2}(\alpha_i + \gamma_i) \right) + \frac{1 - e^{-1/2}}{1 + 2e^{1/2}} \left(\alpha_i - \gamma_i + (1 - (1 - e^{-1/2})(1 - e^{-1}))\alpha_i - \gamma_i \right) \\
& \qquad = \frac{1 - e^{-1/2}}{1 + 2e^{1/2}} \left(2e^{1/2} + 4 - (1 - e^{-1/2})(1 - e^{-1}) \right) \alpha_i \geq 0.645 \; \alpha_i \, .
\end{aligned}$$

The analysis for a player $i \in \mathcal{B}$ would be exactly the same, yielding expected profit at least $0.645 \; \beta_i$. □

3.3 A small improvement in the general case

Here we finally prove that a tiny (but constant) improvement over $1 - 1/e$ in the general case is possible. We have the Butterfly Rounding Procedure which requires that players be divided into groups $\mathcal{A}, \mathcal{B}$ with balanced interest in each item, namely

$$\sum_{i \in \mathcal{A}} y_{ij} = \sum_{i \in \mathcal{B}} y_{ij} = \frac{1}{2} \, ,$$

and we regard the values y_{ij} as infinitesimal. In fact, the analysis of the Butterfly Rounding Procedure is quite exact, provided that the values y_{ij} are not too large. Also, in this case we can argue that a random partition is likely to be approximately balanced. So, let us propose a variant of the Butterfly Rounding Procedure for the general case.

Algorithm 4'

- Partition the players into groups $\mathcal{A}, \mathcal{B}$ by assigning each player uniformly and independently to $\mathcal{A}$ or $\mathcal{B}$. Define
$$z_j = \max\left\{\sum_{i\in\mathcal{A}} y_{ij},\ \sum_{i\in\mathcal{B}} y_{ij},\ \frac{1}{2}\right\}.$$
- Consider the fractional solution as a probability distribution over subsets S for each player. Let X be an independently sampled random set, where item j is present with probability $1/(2z_j)$. We modify the probability distribution of each player by taking $S \cap X$ instead of S. This defines a probability distribution corresponding to a new fractional solution $\tilde{x}_{i,\tilde{S}}$ where
$$\tilde{x}_{i,\tilde{S}} = \sum_{S,X:S\cap X=\tilde{S}} x_{i,S} \Pr[X] = \sum_{S,X:S\cap X=\tilde{S}} x_{i,S} \prod_{j\in X} \frac{1}{2z_j} \prod_{j\notin X}\left(1-\frac{1}{2z_j}\right).$$
Then, we get
$$\tilde{y}_{ij} = \sum_{\tilde{S}:j\in S} \tilde{x}_{i,\tilde{S}} = \sum_{S,X:j\in S\cap X} x_{i,S}\Pr[X] = \frac{y_{ij}}{2z_j}$$
so that
$$\max\left\{\sum_{i\in\mathcal{A}} \tilde{y}_{ij}, \sum_{i\in\mathcal{B}} \tilde{y}_{ij}\right\} \le \frac{1}{2}.$$
- Then run Algorithm 4 on the fractional solution $\tilde{x}_{i,S}$.

Let us fix a very small $\varepsilon > 0$ and call an item "unbalanced" if some player requests it with probability $y_{ij} > \varepsilon$. We claim that Algorithm 4' works well for fractional solutions where no item is unbalanced. This is true because then, $\sum_{i\in\mathcal{A}} y_{ij}$ and $\sum_{i\in\mathcal{B}} y_{ij}$ are random variables well concentrated around $(1/2)\sum_{i=1}^n y_{ij}$; more precisely, their variance is
$$\mathrm{Var}\left[\sum_{i\in\mathcal{A}} y_{ij}\right] = \sum_{i=1}^n \frac{1}{4} y_{ij}^2 \le \frac{\varepsilon}{4}\sum_{i=1}^n y_{ij} \le \frac{\varepsilon}{4}.$$
Therefore, the expected amount by which either sum exceeds $1/2$ is
$$\mathbf{E}[z_j - 1/2] \le \mathbf{E}\left[\left|\sum_{i\in\mathcal{A}} y_{ij} - \frac{1}{2}\sum_{i=1}^n y_{ij}\right|\right] \le \sqrt{\mathrm{Var}\left[\sum_{i\in\mathcal{A}} y_{ij}\right]} = \frac{1}{2}\sqrt{\varepsilon}.$$
The way we obtain the new fractional solution $\tilde{x}_{i,S}$ corresponds to a sampling procedure where each item remains with probability $1/(2z_j)$. Therefore, the expected factor we lose here is
$$\mathbf{E}\left[\frac{1}{2z_j}\right] \ge \frac{1}{2\mathbf{E}[z_j]} \ge \frac{1}{1+\sqrt{\varepsilon}}.$$

Moreover, the analysis of Algorithm 4 (which assumed infinitesimal values of y_{ij}) can be used with small modifications for such a fractional solution. For $0 \le y_{ij} \le \varepsilon$, we have $-y_{ij} \ge \log(1-y_{ij}) \ge -y_{ij}/(1-\varepsilon)$, i. e.,

$$e^{-\sum_i y_{ij}} \ge \prod_i (1-y_{ij}) \ge e^{-\frac{1}{1-\varepsilon}\sum_i y_{ij}} \ge (1-2\varepsilon)e^{-\sum_i y_{ij}}.$$

It can be verified that all the estimates in the analysis of Algorithm 4 are precise up to a relative error of $O(\varepsilon)$ in this case. (Note that we do not use the assumption that $n \to \infty$ anymore.) The fact that the solution may be "sub-balanced" ($\sum_{i\in\mathcal{A}} y_{ij}, \sum_{i\in\mathcal{B}} y_{ij} < 1/2$) can only help. Accounting for the balancing step, we get a solution of expected value $(0.645 - O(\sqrt{\varepsilon}))LP$.

If some items are unbalanced, then running Algorithm 4' might present a problem. However, then we gain by running Algorithm 3. As in Section 2.5, we decide which algorithm to use based on the importance of unbalanced items in the fractional solution. Let U denote the set of unbalanced items, and define

$$\sigma_{ij} = \mathbf{E}[w_i(S_i \cap [j]) - w_i(S_i \cap [j-1])],$$

the expected contribution of item j to player i. Then we distinguish two cases:

- If a non-negligible value comes from unbalanced items, e. g., $\sum_i \sum_{j\in U} \sigma_{ij} \ge \varepsilon \cdot LP$, then we use Algorithm 3. For each unbalanced item $j \in U$, since $y_{ij} > \varepsilon$ for some i, Lemma 1.5 allocates the item to each player with conditional probability

$$\rho_j \ge 1 - \prod_i (1-y_{ij}) \ge 1 - \frac{1-\varepsilon}{e^{-\varepsilon}} e^{-\sum_i y_{ij}} \ge 1 - \left(1 - \frac{1}{2}\varepsilon^2\right) e^{-1}.$$

  By Lemma 2.5, the expected value of our solution is at least

$$\sum_i \sum_j \rho_j \sigma_{ij} \ge \sum_i \sum_j \left(1 - \frac{1}{e}\right)\sigma_{ij} + \sum_i \sum_{j\in U} \frac{1}{2e}\varepsilon^2 \sigma_{ij} \ge \left(1 - \frac{1}{e} + \frac{\varepsilon^3}{2e}\right) \cdot LP.$$

- If the contribution of unbalanced items is negligible, $\sum_i \sum_{j\in U} \sigma_{ij} < \varepsilon \cdot LP$, then let us remove the unbalanced items. This incurs a factor of $(1-\varepsilon)$ on the value of the fractional solution. Then we run Algorithm 4' which yields expected value at least $(0.645 - O(\sqrt{\varepsilon})) \cdot LP$.

For a very small $\varepsilon > 0$, one of the two algorithms beats $1 - 1/e$ by a positive constant amount. A rough estimate shows that we should choose $\varepsilon \simeq 10^{-4}$ in order to keep $0.645 - O(\sqrt{\varepsilon})$ above $1 - 1/e$, and then the first case gives an improvement on the order of 10^{-12}. This finishes the proof of Theorem 1.2.

3.4 Integrality gap of 0.782

We provide an instance of the Configuration LP where the integrality gap is rougly 0.782, thus improving our example of $5/6$ for two players (Section 2.2). We remark that our example generalizes the $7/8$ integrality gap example for two players rather than the $5/6$ gap example (see Section 2.2).

Example 3.3. Consider a set of items $Q_n = [n]^n$. For each $i \in [n]$, consider a random set T_i which contains independently one random item from each layer $H_{i,j}$, $j = 1, \dots, n$. Then we define the utility function of player i as

$$w_i(S) = \Pr_{T_i}[S \cap T_i \neq \emptyset] .$$

We remark that the randomization is just a convenient tool to define the utility functions; however, the functions themselves are deterministic. Observe that each layer $H_{i,j}$ is still optimal for player i, because the random set T_i intersects it with probability 1. We consider the same fractional solution, $x_{i,H_{i,j}} = 1/n$ for each i, j. This gives value 1 to each player, for a total LP optimum of n.

Lemma 3.4. *The optimal integral solution for Example 3.3 has value $OPT \leq (0.782 + o(1))n$.*

Consider a partition where S_i is the set obtained by player i. Let

$$z_{ij} = \frac{1}{n^{n-1}} |S_i \cap H_{i,j}| ,$$

the fraction of layer $H_{i,j}$ obtained by player i. Recall that T_i contains one random element independently from each layer $H_{i,j}$ for $j = 1, \dots, n$. Hence, we have

$$w_i(S_i) = \Pr_{T_i}[S_i \cap T_i \neq \emptyset] = 1 - \prod_{j=1}^{n} (1 - z_{ij}) .$$

What can be a possible assignment of the variables z_{ij}? For a selection of subsets $K_1, K_2, \dots, K_n \subseteq [n]$, the total number of items in the union of layers $\bigcup_{i=1}^{n} \bigcup_{j \in K_i} H_{i,j}$ is $n^n - \prod_{i=1}^{n}(n - |K_i|)$. All the players together cannot obtain more than this many items from the respective layers, which yields

$$\sum_{i=1}^{n} \sum_{j \in K_i} z_{ij} n^{n-1} \leq n^n - \prod_{i=1}^{n} (n - |K_i|) .$$

We can also assume that the variables $z_{i1}, z_{i2}, \dots$ are ordered decreasingly for each i, which makes it sufficient to consider $K_i = \{1, 2, \dots, k_i\}$. Thus the maximum value of an integral solution is $OPT \leq OPT_1$ where

$$\begin{aligned} OPT_1 \quad &= \quad \max \Big\{ n^{n-1} \sum_{i=1}^{n} \left(1 - \prod_{j=1}^{n} (1 - z_{ij}) \right) : \\ \forall\, 1 \leq k_1, k_2, \dots, k_n \leq n, \qquad & \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k_i} z_{ij} \leq 1 - \prod_{i=1}^{n} \left(1 - \frac{k_i}{n} \right) \\ \forall\, 1 \leq i \leq n, \qquad & 1 \geq z_{i1} \geq z_{i2} \geq \cdots \geq z_{in} \geq 0 \Big\} . \end{aligned}$$

In the following, we estimate OPT_1 from above.

Lemma 3.5. $(1-\varepsilon(n))OPT_1 \leq OPT_2$ *where* $\varepsilon(n) \to 0$ *and*

$$\begin{aligned} OPT_2 &= \max\left\{ n^{n-1}\sum_{i=1}^{n}\left(1-\prod_{j=1}^{n}(1-y_{ij})\right) : \right. \\ \forall\, 1 \leq k_1,k_2,\dots,k_n \leq n, &\qquad \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k_i} y_{ij} \leq 1-e^{-\sum_{i=1}^{n}k_i/n} \\ \forall\, 1 \leq i \leq n, &\qquad \left. 1 \geq y_{i1} \geq y_{i2} \geq \dots \geq y_{in} \geq 0 \right\}. \end{aligned}$$

Proof. Note that we are replacing the contraints by stronger conditions, so $OPT_1 \geq OPT_2$. But we are more interested in the opposite inequality. Choose an arbitrarily small $\varepsilon > 0$ and consider an optimal solution z_{ij} to OPT_1. We modify this to a near-optimal solution z'_{ij} with a bounded sum $\sum_{j=1}^{n} z'_{ij}$ for each i. Recall that $z_{i1} \geq z_{i2} \geq \dots$; set $C_\varepsilon = \log(1/\varepsilon)$ and if $\sum_{j=1}^{n} z_{ij} > C_\varepsilon$, take k_i minimum such that $\sum_{j=1}^{k_i} z_{ij} > C_\varepsilon$. Keep only these k_i values, $z'_{ij} = z_{ij}$ for $j \leq k_i$, and set the rest to $z'_{ij} = 0$ (for $j > k_i$). Now,

$$\sum_{j=1}^{n} z'_{ij} \leq C_\varepsilon + 1 \leq 2C_\varepsilon\,.$$

On the other hand, since we have $\sum_{j=1}^{n} z'_{ij} > C_\varepsilon$ for any i where we modified the solution, and then

$$\prod_{j=1}^{n}(1-z_{ij}) \leq \prod_{j=1}^{n}(1-z'_{ij}) \leq e^{-C_\varepsilon} = \varepsilon\,,$$

we didn't lose more than εn^n in the objective function. We will see that the optimum is $OPT_1 = \Omega(n^n)$, so we lose only $O(\varepsilon)OPT_1$.

We claim that for n sufficiently large and any $\ell \leq n$,

$$\frac{1-\varepsilon}{n}\sum_{i=1}^{\ell}\sum_{j=1}^{k_i} z'_{ij} \leq 1-e^{-\sum_{i=1}^{\ell}k_i/n}\,. \tag{3.2}$$

If this holds, then $y_{ij} = (1-\varepsilon)z'_{ij}$ is a feasible solution to OPT_2, which means that $OPT_2 \geq (1-O(\varepsilon))OPT_1$.

Assume WLOG that $1 \leq k_1 \leq k_2 \leq \dots \leq k_\ell$. If all the k_i's are bounded by a constant, for instance $(4/\varepsilon)C_\varepsilon$, then we are done, since then $1-k_i/n = (1-O(n^{-2}))e^{-k_i/n}$ and

$$1-\prod_{i=1}^{\ell}\left(1-\frac{k_i}{n}\right) = 1-(1-O(\ell n^{-2}))e^{-\sum_{i=1}^{\ell}k_i/n} \leq \frac{1}{1-\varepsilon}\left(1-e^{-\sum_{i=1}^{\ell}k_i/n}\right)$$

for a sufficiently large n.

If $k_\ell > (4/\varepsilon)C_\varepsilon$, we use induction on ℓ. We assume that (3.2) holds for $\ell-1$. Going from $\ell-1$ to ℓ, the right-hand side of (3.2) increases by

$$\left(1-e^{-\sum_{i=1}^{\ell}k_i/n}\right)-\left(1-e^{-\sum_{i=1}^{\ell-1}k_i/n}\right) = e^{-\sum_{i=1}^{\ell-1}k_i/n}\left(1-e^{-k_\ell/n}\right).$$

We can assume that $e^{-\sum_{i=1}^{\ell} k_i/n} \geq \varepsilon$, otherwise (3.2) is trivial. Also, we assumed $k_\ell > \frac{4}{\varepsilon} C_\varepsilon$, so the RHS increases by at least

$$\varepsilon(1-e^{-k_\ell/n}) \geq \varepsilon(1-e^{-1})\frac{k_\ell}{n} \geq \frac{2C_\varepsilon}{n}\,.$$

Meanwhile, the LHS increases by

$$\frac{1-\varepsilon}{n}\sum_{j=1}^{k_\ell} z'_{ij} \leq \frac{1-\varepsilon}{n} \cdot 2C_\varepsilon$$

which proves (3.2). □

Now we estimate OPT_2. The constraints imply that when we order the y_{ij}'s in a decreasing sequence, the sum of the m largest ones is bounded by $n(1-e^{-m/n})$. We claim that the optimum is attained when this bound is tight for any m, i. e., the m-th largest y_{ij} is equal to $n(e^{-(m-1)/n} - e^{-m/n}) \simeq e^{-m/n}$. This can be seen by considering the first y_{ij}, being the m-th largest, which violates this condition and is smaller than $e^{-m/n}$. Denote by $P_m = \prod_{j' \neq j}(1-y_{ij'})$ the product for the remaining entries in the i-th row, apart from y_{ij}. Similarly, let $y_{k\ell}$ be the $(m+1)$-th largest entry and denote by $P_{m+1} = \prod_{\ell' \neq \ell}(1-y_{k\ell'})$ the product for the remaining entries in its row. If $P_m > P_{m+1}$ then we can switch y_{ij} and $y_{k\ell}$ and decrease the sum of row products $\prod_j(1-y_{ij}) + \prod_\ell(1-y_{k\ell})$. If $P_m \leq P_{m+1}$ then we can increase y_{ij} slightly and decrease $y_{k\ell}$ by the same amount, which again decreases the average of the two row products. In both cases, we increase the objective function.

Thus we know exactly the optimal sequence of entries y_{ij}: the m-th largest one is roughly $e^{-m/n}$. We only have to find their optimal placement in the $n \times n$ matrix y_{ij}. Since the product $\prod_{i,j}(1-y_{ij})$ is fixed, and we minimize the *sum of the row products* $\sum_i \prod_j(1-y_{ij})$, we try to make these products as uniform as possible. We claim that $\sum_{i=1}^n \prod_{j=1}^n (1-y_{ij})$ is minimized under these conditions, when:

- There is m such that the first m rows contain $y_{i1} = e^{-i/n}$, and $y_{ij} \simeq 0$ for $j > 1$ (i. e., the smallest possible entries are placed here).

- The remaining entries are distributed in the remaining $n-m$ rows so that their products $\prod_{j=1}^n (1-y_{ij})$ are all approximately equal to

$$\rho = \left(\prod_{r=m+1}^{n^2-m(n-1)} \left(1-e^{-r/n}\right)\right)^{1/(n-m)}.$$

- m is chosen so that $\rho \simeq 1-e^{-m/n}$.

To see this, denote by M the rows containing the m largest entries (it's easy to see that these should be in m distinct rows). Any of these rows has a product $\prod_{j=1}^n (1-y_{ij}) \leq 1-e^{-m/n} = \rho$. Outside of M, consider the row with the largest product $\prod_{j=1}^n (1-y_{ij})$. By averaging, this must be larger than ρ. If there is any entry in it smaller than some entry in M, switch these two entries and gain in the objective function. Therefore, all the smallest entries must be next to the m largest entries in M.

Outside of M, the optimal way is to make the products $\prod_{j=1}^{n}(1-y_{ij})$ as uniform as possible, since we are minimizing their sum. So we make them all approximately equal to ρ. The value of m is chosen so that the last row in M has a product approximately equal to ρ as well.

We find the approximate solution of

$$1-e^{-m/n} = \left(\prod_{r=m+1}^{n^2-m(n-1)} \left(1-e^{-r/n}\right)\right)^{1/(n-m)}$$

by taking logs and replacing a sum by an integral. We substitute $m=\mu n$ and $r=xn$, which yields

$$\log\left(1-e^{-\mu}\right) = \frac{1}{1-\mu}\int_{\mu}^{\infty} \log\left(1-e^{-x}\right) dx\,. \tag{3.3}$$

The numerical solution of this equation is $\mu \simeq 0.292$. The value of our optimal solution is then

$$\begin{aligned} OPT_2 &= n^{n-1}\left(\sum_{i=1}^{m} e^{-i/n} + (n-m)e^{-m/n}\right) \\ &\simeq n^n\left(\int_0^{\mu} e^{-x}dx + (1-\mu)e^{-\mu}\right) = n^n(1-\mu e^{-\mu})\,. \end{aligned}$$

Substituting the solution of (3.3) yields $OPT_2 \simeq 0.782\, n^n$.

4 Hardness results

In this section we show that there is some constant $\rho < 1$ such that it is NP-hard to approximate the Submodular Welfare Problem within a ratio better than ρ. We shall present several proofs of this fact that differ from each other in the number of players involved and in the type of submodular functions involved.

Previously it was shown in [13] that the maximum submodular welfare problem in the *value oracle model* is hard to approximate within a ratio better than $1-1/e$. In [13] the source of the hardness result is the complexity of individual utility functions: given k, it is already NP-hard to approximate within a ratio better than $1-1/e$ the maximum utility that a single player can derive by choosing at most k items (even if no other player exists). In particular, it is NP-hard for players to answer demand queries (in the construction of [13]). In contrast, we are interested in cases where the utility functions of individual players are simple and every player can easily answer any reasonable query about her utility function, including demand queries. Hence our hardness of approximation results highlight the difficulty of coordinating the wishes of different players, rather than the difficulty of a single player figuring out what she actually wants.

We shall be considering utility functions that come from one of the following three families.

1. *Constant size functions.* Every player is interested only in a constant (say, t) number of the items. All other items have 0 value for the player. Hence the utility function of a player can be explicitly given by a table with a constant 2^t number of entries. This allows every player to answer demand queries (and essentially any other type of query) efficiently—in constant time.

2. *Bounded.* For some constant t, the utility function of a player can be represented in full by a table specifying the utility of those sets of items that have size bounded by t. The value of any other set T is $w(T) = \max_{S \subset T} w(S)$, that is, the value of the most valuable bounded set that is fully contained in T. Hence, for every set T, its utility is determined by at most t items within the set, and all other items have zero marginal utility. However, unlike the case of constant size functions, the set of items that has zero marginal utility depends on the set T and is not universal for all choices of T. The number of entries in the table describing a bounded utility function is at most m^t, which is polynomial when t is constant. Clearly, both value queries and demand queries can be answered in polynomial time. However, as we shall see in Section 4.1, there are other types of queries that cannot be answered efficiently.

3. *Separable.* The items can be partitioned into disjoint classes C_i of constant size. The utility of set S is computed as $w(S) = \sum_i w(S \cap C_i)$. Namely, it is the sum of constant size functions over disjoint sets of items. Again, both value queries and demand queries can be answered in polynomial time.

In the following theorem we state our main results for the three families of utility functions that are described above.

Theorem 4.1. *There is some constant $\rho < 1$ such that it is NP-hard to approximate the Submodular Welfare Problem within a ratio better than ρ in the following cases.*

1. *When all players have constant size utility functions. Specifically, we will show a proof with $t = 15$.*

2. *When all players have the same utility function and it is bounded. In this case we also present explicit values for ρ. Specifically, we shall show a proof with $t = 7$ and $\rho \simeq 0.9964$.*

3. *When there are only two players and their utility functions are separable.*

We remark that in case 2 of Theorem 4.1 we could not have used constant size utility functions as in case 1, because when all players have the same utility function and it is of constant size the Submodular Welfare Problem can be solved in polynomial time. Likewise, in case 3 we could not have used bounded utility functions as in case 2, because when there are only a constant number of players and the utility functions are bounded, the Submodular Welfare Problem can be solved in polynomial time. (In both cases, only a constant number of items play a role and all allocations can be enumerated.)

Subsequently to our work, a hardness of $(15/16 + \varepsilon)$-approximation was proved for the Submodular Welfare Problem in the demand oracle model [3]. We note that in this result, the utility functions are of the "budget-additive" form, $w_i(S) = \min\{\sum_{j \in S} w_{ij}, B_i\}$, and in particular such that demand queries can be answered efficiently. However, they do not fall in any of the classes discussed above. For budget-additive utility functions, a $3/4$-approximation was given in [3]. We are not aware of improved approximation results for the classes of constant size, bounded or separable utility functions.

An interesting feature of our constructions (as well as of the results of [13]) is that they have *perfect completeness*. On positive instances, there is no conflict between the players, in the sense that there is an allocation that gives every player the maximum utility she could have got have there been no other players. (Remark: perfect completeness does not necessarily require all items to be allocated, as some items that have zero marginal value may remain unallocated.) Our proofs show that it is *NP*-hard to distinguish such instances from instances in which only a $\rho < 1$ fraction of the welfare can be recovered.

The rest of the section is organized as follows. Section 4.1 is somewhat of a digression. It presents an incentive compatible mechanism that achieves maximum welfare when all players have the same utility function. This mechanism highlights the significance of constant size utility functions. Section 4.2 shows our methodology for achieving the submodularity property for utility functions. Section 4.3 proves part 1 of Theorem 4.1. Section 4.4 proves part 2 of Theorem 4.1. Section 4.5 reviews the hardness of approximation result of [6] for max k-cover and the hardness of approximation result of [4] for maximum welfare with XOS utility functions. Section 4.6 presents a new hardness of approximation result for welfare maximization with XOS utility functions, but this time with constant size utility functions (see Theorem 4.4). The motivation for such a result is explained in Section 4.1. Section 4.7 proves part 3 of Theorem 4.1, based on special properties of known hardness results for max k-cover.

4.1 Fair division queries

Inspired by methods for the fair division of goods (sometimes referred to as *cake cutting* theorems, see [1] for example), we propose the following allocation algorithm for the maximum welfare problem. The version of the algorithm presented here is most useful when all players have exactly the same utility function. This is an important special case, and may arise naturally when players have the ability to resell items that are allocated to them. We make no assumptions about the nature of the utility function, and in particular, it need not be submodular.

1. Initialization. Let P denote the set of players, and let S denote the set of all items.
2. If S is empty, allocate no items to the players and end.
3. If $|P| = 1$, allocate all of S to the player and end.
4. Pick an arbitrary player $p \in P$ and ask her to answer the following *fair division query*: partition S into $|P|$ parts (some of which may be empty). Denote the reply of p by $S_1, \dots, S_{|P|}$.
5. Pick one part (say, S_i) at random and allocate the items in it to player p. Remove p from P, remove S_i from S, and return to step 2.

The above algorithm is incentive compatible in the following sense. If a player wishes to maximize her expected welfare, then her answer to a fair division query must maximize the sum of utilities of the parts in the partition. We say that a player is *truthful in expectation* if indeed she follows such a strategy. Now it easily follows that if all players have the same utility function, and all players are truthful in expectation, then the above algorithm actually produces the maximum welfare (regardless of whether the utility function is submodular or not). This should be contrasted with the fact that the published hardness of approximation results (e. g., within ratio of $1 - 1/e$ for XOS utility functions [4], within ratio of 1/2 for subadditive utility function, see for example [7], or within ratio of roughly $1/\sqrt{m}$ for general utility functions) are all proved for cases in which all players have the same utility functions. In all these proofs, the utility functions are simple in the sense that they allow the players to efficiently answer demand queries, but clearly, as our algorithm above implies, the utility functions are still too complicated so as to allow players to answer fair division queries.

Based on the above, one may argue that existing hardness of approximation results involve players who do not fully understand their own utility function, in the sense that there are queries that the player has incentive to answer truthfully, but finds it *NP*-hard to answer them. On the other hand, one may argue that it is easy to modify the existing hardness of approximation results so that not all players have the same utility functions (and indeed this is what we shall do...), and then the algorithm presented above would not necessarily provide a good approximation ratio. Taking these arguments into account, the points that we wish to make here are the following.

1. We showed an explicit example where the query model allows one to solve *NP*-hard allocation problems by transferring the computational difficulties to the players.

2. In some existing *NP*-hardness results, it is hard to distinguish whether the hardness is a consequence of the nature of individual utility functions, or whether it is the consequence of interplay between several simple utility functions.

It is our belief that the use of constant size utility functions is the most natural way to address the two points above. It is hard to imagine any reasonable query that would be difficult to answer regarding such a utility function, and hence queries will not transfer the computational burden to the players. Likewise, hardness results proved when utility functions have constant size are more obviously a consequence of there being multiple players, rather than being a consequence of the difficulty to reason about a single utility function.

4.2 Simple submodular extensions

A recurring theme in our proofs is that for each player, there will be certain sets of items that are *special*. The special sets will typically all be of the same size. If a player gets one of her special sets, her utility is maximized. On positive instances, there will be an allocation in which every player gets a special set. On negative instances, for every allocation some players do not get a respective special set, due to conflicts with other players. To prove a hardness of approximation result, we need to define the utility that a player gets out of sets that are not special. Hence one needs to extend the utility function originally defined only on the special sets to all sets. This needs to be done while maintaining the following properties:

1. The resulting utility function is submodular.

2. The resulting utility function is "simple" in the sense that one can efficiently answer demand queries.

3. The utility per item ($w(S)/|S|$) of nonspecial sets is "significantly" lower than that of special sets.

If one replaces the requirement that utility functions are submodular by the weaker requirement that utility functions are fractionally subadditive (or equivalently, XOS, see [7]), then one can use the following *simple XOS extension*. The value of a set T is $\max[|T \cap S|]$, where S ranges over all special sets.

That is, special sets have maximum value and every item in them contributes 1 towards this value. Other sets have value that depends on the maximum size of their intersection with a special set. Those items in the intersection may be thought of as having value 1, and the other items may be thought of as having 0 marginal value. It is not hard to see that the simple XOS extension does not result in a submodular utility function.

Achieving the above three properties when utility functions are submodular has been the stumbling block of extending known hardness of approximation results from more general classes of utility functions (e. g., the $1-1/e$ hardness result for XOS utility functions [4]) to the case of submodular utility functions. The key observation that allows us to achieve this in our work is the fact that we need to consider only sets of items of size bounded by t. Specifically, we use the following approach.

Let $\{S_i\}$ be a collection of special sets, all of size b. Let w be a partially defined utility function, giving utility b to every special set. We define the *simple submodular extension* of w to be as follows:

1. For every set S with $|S| < b$, $w(S) = |S|$.
2. For every set S with $|S| > b$, $w(S) = b$.
3. For every set S with $|S| = b$, $w(S) = b$ if S is a special set, and $w(s) = b - 1/2$ otherwise.

The three properties above are satisfied. It is not hard to see that w as above is indeed submodular. When b is constant, then w is also "simple." And nonspecial sets of size b have value a factor of $(1-1/2b)$ lower than that of the special sets, which is significant when b is constant.

We now explain our convention of how to apply the simple submodular extension to each one of the three families of utility functions that we consider.

1. *Constant size.* We will be interested in cases where special sets have size b, and the union of all special sets has constant size, at most t. In this case the extension is defined only on the union of the special sets, and the remaining items have zero value.
2. *Bounded.* We will be interested in the case in which all special sets have size b, and then we set $t = b+1$. Only the first t items in a set matter, because any set of cardinality t has value b, which is the maximum value a set may have.
3. *Separable.* We will take the simple submodular extension on each class separately (as done for constant size functions), and then the submodular utility function will be the sum of the utility functions over all classes.

4.3 Constant size utility functions

Here we prove part 1 of Theorem 4.1. (An alternative proof will also appear in Section 4.6.) Our proof will be via a reduction from the problem *max 3-coloring-5*. Specifically, we shall use the following *NP*-hardness result whose proof appears in [8].

Lemma 4.2. *There is some $\varepsilon > 0$ such that given a 5-regular graph, it is NP-hard to distinguish between the case that it can be legally 3-colored, and the case in which every 3-coloring of its vertices leaves an ε fraction of its edges illegally colored (both endpoints have the same color).*

Given a 5-regular graph with n vertices and m edges (hence $2m = 5n$) we reduce it to the following submodular welfare maximization problem. With every edge e we associate three items, e_1, e_2 and e_3, corresponding to the three "colors" $\{1,2,3\}$. Hence there are $3m$ items. There will be m edge players, one for every edge, and n vertex players, one for every vertex. The utility function of the player p_e who is associated with edge e gives the player utility 1 if she receives at least one of the three items associated with the edge, and utility 0 otherwise. Hence it is a constant size submodular utility function. The utility function of the player p_v who is associated with vertex v will be nonzero on 15 items, and will have three special sets of size 5: the set of all 5 items of color 1 associated with edges incident with v, and likewise for colors 2 and 3. The utility function of p_v is the simple submodular extension (as described in Section 4.2) of this function.

On positive instances, we can legally 3-color the graph. Then each vertex player gets the five items associated with her chosen color, giving her utility 5. Each edge player can get the item not allocated to the two players at the endpoint of the edge, giving her utility 1. Altogether, the total welfare is $3m$ (all items are allocated and give utility 1 per item), and all players are maximally happy.

On negative instances, we use the following analysis. Without loss of generality, we may assume that every edge player gets one item (because then the item contributes marginal utility 1, and there is no way by which it can contribute larger marginal utility). Likewise, it is is not hard to see that we may assume that every vertex player gets exactly 5 items, one from every incident edge (otherwise a shifting argument would yield an allocation with at least as much welfare). We distinguish between "legally colored vertices" for which all allocated items have the same color, and "illegally colored vertices" for which not all allocated items have the same color. The number of illegally colored vertices is at least $\varepsilon m/3$ (in a coloring that maximizes the number of legally colored edges, every vertex is incident with at most 3 illegally colored edges, by giving it the majority color of its neighbors), and every illegally colored vertex has utility 9/2 rather than 5 (by the simple submodular extension). Hence the maximum welfare is at most $3m - \varepsilon m/6$, showing that Submodular Welfare cannot be approximated within a factor better than $\rho = 1 - \varepsilon/18$.

We remark that rather than use the simple submodular extension, one may use a different submodular extension that gives a value somewhat better than $1 - \varepsilon/18$ for ρ. Details are omitted.

4.4 Bounded utility functions

Here we prove part 2 of Theorem 4.1. Our proof is by reduction from the problem of finding a maximum matching in k-uniform hypergraphs. Specifically, we choose the value of $k = 6$ because then we can use as a blackbox the following result of [12]. Recall that a *matching* in a hypergraph is a collection of hyperedges that do not intersect.

Lemma 4.3. *For every $\varepsilon > 0$, given a 6-uniform hypergraph with n vertices, it is NP-hard to distinguish between the case that it has a matching covering at least $(1-\varepsilon)n$ vertices, and the case in which every matching covers at most $22n/23$ vertices.*

Given an instance of 6-uniform hypergraph matching, we reduce it to an instance of Submodular Welfare as follows. The vertices of the graph are the items. There are $(1-\varepsilon)n/6$ players. All players have exactly the same utility function, and it is bounded. The hyperedges of the graph are the special sets of the utility function. Hence in its simple submodular extension (as in Section 4.2), sets corresponding to hyperedges have value 6, other sets of size 6 have value 5.5, and sets of size 7 or more have value 6.

On positive instances, every player gets a hyperedge of the maximum matching, giving utility $(1-\varepsilon)n$, and every player is maximally happy. On negative instances, it is not hard to see that the best allocation will give $22n/(23\cdot 6)$ players a hyperedge (giving utility $22n/23$), εn players will get 7 vertices each (giving utility $6\varepsilon n$), and $(1/23-2\varepsilon)n$ players will get 6 vertices that do not form a hyperedge (giving utility 5.5 per player). When ε tends to 0, the ratio between the negative and positive instances tends to $1-1/(12\cdot 23)\simeq 0.9964$.

4.5 Review of hardness for Max k-cover

Recall that in the max k-cover problem, we are given a collection $\mathcal{S}$ of sets and a parameter k, and the objective is to choose k sets whose union covers the maximum number of items. A straightforward greedy algorithm (or alternatively, use of a linear programming relaxation) achieves an approximation ratio of $1-1/\mathrm{e}\simeq 0.632$ for this problem. In [6] it is shown that for every ε it is *NP*-hard to distinguish between the case that there are k sets that disjointly cover all items, and the case in which every collection of k sets cover only a $1-1/\mathrm{e}+\varepsilon$ fraction of the items. Hence it is *NP*-hard to approximate the max k-cover problem within a ratio better than $1-1/\mathrm{e}+\varepsilon$.

Let us now recall how the hardness of max k-cover is used in [4] to show hardness for the maximum welfare problem with XOS utility functions. There are k players. They all have the same utility function. In this utility function, the special sets are exactly those in the collection $\mathcal{S}$ of sets of the max k-cover problem, and the utility function is the simple XOS extension of this set system. On positive instances, allocating to the k players the k sets of the optimal solution gives utility 1 per item. On negative instances, the k players may get k arbitrary sets (not necessarily from the collection $\mathcal{S}$). For each such set consider its maximum intersection with a set $S\subset\mathcal{S}$. The union of all these maximum intersection covers at most an $1-1/\mathrm{e}+\varepsilon$ fraction of the items, which for the simple XOS extension means that the utility per item is only $1-1/\mathrm{e}+\varepsilon$ on average.

The utility function based on simple XOS extension is simple in the sense that a player can answer demand queries efficiently. However, it is *NP*-hard to answer fair division queries. This can be seen either directly, as these queries amount to solving the max k-cover problem, or indirectly, as an implication of the algorithm of Section 4.1.

We now show how one may obtain a $1-1/\mathrm{e}+\varepsilon$ hardness of approximation result for the maximum welfare when utility functions are XOS and of constant size. We shall use certain properties of the set systems constructed in [6]. Unfortunately, there is no explicit theorem in [6] that lists all these properties. We do not wish to reproduce the rather lengthy proof of [6] in detail here. Hence we shall only list the properties that we use, together with some clearly marked "hints" regarding why the construction of [6] has these properties.

Fix an arbitrarily small value of $\varepsilon>0$. The polynomial time reduction from max-3SAT-5 to max k-cover described in [6] has the following properties.

1. It constructs a set system. The set system can be viewed as being composed of *groups* of sets, where each set belongs to exactly one group. (Hint: recall that each set in [6] corresponds to a possible answer of a prover to a possible query. Those sets that correspond to the possible answers

of the same prover to the same query will form a group.) The number of groups will be denoted here by k (though in [6] it is denoted by k', and k denotes there the number of provers).

2. The number of sets in a group is bounded above by some constant that depends only on ε. (Hint: this is the consequence of the fact that to prove hardness of approximation for max k-cover it suffices to have a constant number of parallel repetitions.)

3. All sets are exactly of the same size. (Hint: this uses the strong regularity properties of the construction, such as the fact that we use max 3SAT-5 rather than just max 3SAT, the use of Hadamard codes in the multi-prover system described in Section 2.3 in [6], and the partition system described in the proof of Theorem 5.3 in [6].) Likewise, all groups contain exactly the same number of sets (though we do not need to use this fact).

4. The size of every set can be bounded from above by some constant that depends only on ε. (Hint: since the number of parallel repetitions is constant, each query to a prover can be completed in only a constant number of ways to queries to the other provers. Moreover, one can choose the number of points in every partition system to be a large constant.)

5. For "yes" instances of max 3SAT-5, all points can be covered by k mutually disjoint sets, one from every group.

6. For "no" instances of max 3SAT-5, every collection of k sets cover at most a fraction of $(1-1/\mathrm{e}+\varepsilon)$ of the points.

4.6 From max k-cover to maximum welfare

In this section, we prove the following results.

Theorem 4.4. *For welfare maximization with constant-size XOS utility functions, it is NP-hard to achieve a $(1-1/\mathrm{e}+\varepsilon)$-approximation; with constant-size subadditive utility functions, it is NP-hard to achieve a $(1/2+\varepsilon)$-approximation, for any fixed $\varepsilon > 0$.*

We interpret instances of max k-cover as described in Section 4.5 as instances of the maximum welfare problem. The points in the set system will be the items. There will be k players, one for every group of sets. Consider a player p that is associated with a group g_p of sets. Let s be the size of a single set (recall that all sets have the same size and that this size is a constant that depends only on ε). Let ℓ be the number of sets in a group (recall that all groups have the same number of sets and that this number is a constant that depends only on ε). Let us define $t = s\ell$, and observe that t is a constant that depends only on ε. Let $G_p = \bigcup_{S \in g_p} S$ be the union of all sets in g_p, and observe that $|G_p| \leq t$. Using the above notation, we describe the utility function w_p of player p. This function will be bounded above by 1 (and below by 0).

Only points in G_p will have nonzero utility for player p. Hence to completely specify w_p it suffices to specify $w_p(S)$ only for those sets $S \subset G_p$. As there are at most 2^t such sets S, the utility function w_s can be described in full by an explicit table with a constant number of entries.

We call the sets in g_p the special sets, and they each have s items and utility s. We first address the case of XOS utility functions. In this case, the utility function w_p of player p is the simple XOS extension of the special sets. Equivalently, for every set T we have that $w_p(T) = \max_{S \in g_p}[|T \cap S|]$. As in the case of [4], this reduction gives a hardness of approximation within a ratio of $1 - 1/\mathrm{e} + \varepsilon$, but the advantage of our reduction is that each of the utility functions is of constant size.

Using additional properties of the reduction of [6] (on negative instances, the fraction of players that get a special set is in fact arbitrarily small), one can show hardness of approximation within a ratio of $1/2 + \varepsilon$ when utility functions are subadditive. The following subadditive utility function can be used: special sets have value 2 to the respective player, and other sets (that do not contain a special set) have value 1.

For the case of submodular utility functions, we obtain an alternative way to prove the first part of Theorem 4.1. We can use the simple submodular extension. Specifically, this gives:

1. For every set T for which $|T \cap G_p| < s$, we have $w_p(T) = |T \cap G_p|$.
2. For every set T for which $|T \cap G_p| > s$, we have $w_p(T) = s$.
3. For every set T for which $|T \cap G_p| = s$, we have $w_p(T) = s$ if $T \cap G_p \in g_p$ and $w_p(T) = s - 1/2$ if $T \cap G_p \notin g_p$.

On positive instances, it is again true that every player can get utility s. To analyze negative instances, it suffices to observe the following property: the fraction of players p for which the allocated set T_p fully contains some set $S \in g_p$ is at most $(1 - 1/\mathrm{e} + \varepsilon)$. (In fact, it is much smaller, but this is not needed here.) It is not hard to see that at best the other players get average utility $s - 1/2$. Hence the maximum welfare in this case gives average utility at most $s - (1/\mathrm{e} - \varepsilon)/2$ per player, and hence it is *NP*-hard to approximate Submodular Welfare within ratio better than $1 - (1/\mathrm{e} - \varepsilon)/2s$. Observe that this ratio is bounded away from 1 (but not by much) because s is some constant.

4.7 Two players

In this section we sketch the proof the third part of Theorem 4.1. Our proof is again based on a reduction from max k-cover, and again we will use the special properties of the reduction as described in Section 4.5. However, we shall use a simpler version of the proof of [6]. Namely, rather than use a k prover proof system (which was perhaps the central aspect in which [6] improves over the previous reduction of [15]), we shall limit the number of provers to two (as in the proof in [15]). This has the effect that the hardness of approximation ratio for max k-cover becomes $3/4 + \varepsilon$ rather than $1 - 1/\mathrm{e} + \varepsilon$, which is not of significant importance in our context. More importantly, this allows the reduction to have the following property (in addition to the properties listed in Section 4.5).

- The groups can be partitioned into two collections. Within a collection, all groups are disjoint in the sense that two sets in different groups within the same collection cannot share an item. (Hint: one collection corresponds to answers of one prover, the other collection corresponds to answers of the other prover.)

In the reduction to Submodular Welfare with two players, each player will be associated with one collection of groups. Its utility function will be the separable function defined on the collection, where each group serves as a constant size utility function as in Section 4.6, and the total utility is the sum of utilities over groups in the collection. We shall use a simple submodular extension as explained in Section 4.2. In a sense, a player is simply the union of $k/2$ players from Section 4.6. The analysis of the hardness of approximation result is essentially as in Section 4.6, except that $1/e$ changes to $1/4$.

Acknowledgment

We would like to thank the anonymous referees for detailed comments, which helped improve the exposition.

References

[1] STEVEN J. BRAMS AND ALAN D. TAYLOR: *Fair division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996. 281

[2] GRUIA CALINESCU, CHANDRA CHEKURI, MARTIN PÁL, AND JAN VONDRÁK: Maximizing a submodular set function subject to a matroid constraint. In *Proc. 12th Intern. Conf. Integer Program. Comb. Opt. (IPCO)*, volume 4513 of *LNCS*, pp. 182–196. Springer, 2007. [doi:10.1007/978-3-540-72792-7_15] 256

[3] DEEPARNAB CHAKRABARTY AND GAGAN GOEL: On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. *SIAM J. Comput.*, 39(6):2189–2211, 2010. Preliminary version appeared in Proc. 49th FOCS, 2008. [doi:10.1137/080735503] 250, 280

[4] SHAHAR DOBZINSKI, NOAM NISAN, AND MICHAEL SCHAPIRA: Approximation algorithms for combinatorial auctions with complement-free bidders. *Math. Oper. Res.*, 35(1):1–13, 2010. Preliminary version appeared in Proc. 37th STOC, 2005. [doi:10.1287/moor.1090.0436] 249, 251, 281, 283, 285, 287

[5] SHAHAR DOBZINSKI AND MICHAEL SCHAPIRA: An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proc. 17th SODA*, pp. 1064–1073, 2006. [doi:10.1145/1109557.1109675] 247, 249, 250, 251, 254, 255

[6] URIEL FEIGE: A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998. [doi:10.1145/285055.285059] 281, 285, 286, 287

[7] URIEL FEIGE: Maximizing social welfare when utility functions are subadditive. In *Proc. 38th STOC*, pp. 41–50. ACM Press, 2006. [doi:10.1145/285055.285059] 249, 250, 251, 252, 253, 254, 258, 259, 281, 282

[8] URIEL FEIGE, MAGNUS HALLDORSSON, GUY KORTSARZ, AND ARAVIND SRINIVASAN: Approximating the domatic number. *SIAM J. Comput.*, 32(1):172–195, 2002. [doi:10.1137/S0097539700380754] 283

[9] URIEL FEIGE AND JAN VONDRÁK: Approximation algorithms for combinatorial allocation problems: Improving the factor of $1-1/e$. In *Proc. 47th FOCS*, pp. 667–676. IEEE Comp. Soc. Press, 2006. [doi:10.1109/FOCS.2006.14] 251

[10] MARSHALL L. FISHER, GEORGE L. NEMHAUSER, AND LAWRENCE A. WOLSEY: An analysis of approximations for maximizing submodular set functions—II. *Math. Program. Stud.*, 8:73–87, 1978. [doi:10.1007/BFb0121195] 249

[11] JOHAN HÅSTAD: Clique is hard to approximate within n to the power $1-\varepsilon$. *Acta Math.*, 182(1):105–142, 1999. [doi:10.1007/BF02392825] 249

[12] ELAD HAZAN, SHMUEL SAFRA, AND ODED SCHWARTZ: On the complexity of approximating k-dimensional matching. In *Proc. 6th Intern. Workshop Approx. Algorithms for Comb. Opt. (APPROX)*, volume 2764 of *LNCS*, pp. 83–97. Springer, 2003. [doi:10.1007/978-3-540-45198-3_8] 284

[13] SUBHASH KHOT, RICHARD LIPTON, EVANGELOS MARKAKIS, AND ARANYAK MEHTA: Inapproximability results for combinatorial auctions with submodular utility functions. *Algorithmica*, 52(1):3–18, 2008. Preliminary version appeared in Proc. 1st Intern. Workshop Internet and Network Economics (WINE), 2005. [doi:10.1007/s00453-007-9105-7] 249, 250, 279, 280

[14] BENNY LEHMANN, DANIEL LEHMANN, AND NOAM NISAN: Combinatorial auctions with decreasing marginal utilities. *Games Econom. Behav.*, 55(2):270–296, 2006. Preliminary version appeared in Proc. Conf. Electron. Commerce 2001. [doi:10.1016/j.geb.2005.02.006] 249

[15] CARSTEN LUND AND MIHALIS YANNAKAKIS: On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994. [doi:10.1145/185675.306789] 287

[16] NOAM NISAN AND ILYA SEGAL: The communication requirements of efficient allocations and supporting prices. *J. Econom. Theory*, 129(1):192–224, 2006. [doi:10.1016/j.jet.2004.10.007] 249

[17] TUOMAS SANDHOLM: Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002. [doi:10.1016/S0004-3702(01)00159-X] 249

[18] JAN VONDRÁK: *Submodularity in combinatorial optimization*. PhD thesis, Charles University, Prague, Czech republic, 2007. 250, 251, 264, 266

[19] JAN VONDRÁK: Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. 40th STOC*, pp. 67–74. ACM Press, 2008. [doi:10.1145/1374376.1374389] 249

AUTHORS

Uriel Feige
Professor
The Weizmann Institute
Rehovot, Israel
uriel.feige@weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~feige/

Jan Vondrák
Research staff member
IBM Almaden Research Center
San Jose, CA
jvondrak@us.ibm.com
http://math.princeton.edu/~jvondrak

ABOUT THE AUTHORS

URIEL FEIGE is a professor at the Weizmann Institute, though this work was done while he was a member of the theory group at Microsoft Research. His main research interests involve studying the borderline between *P* and *NP* as it manifests itself in approximation algorithms, heuristics, and exact algorithms that are not necessarily polynomial time.

JAN VONDRÁK is a researcher at the IBM Almaden research center. His interests are in approximation algorithms and probabilistic combinatorics. He graduated from M.I.T. in 2005 under the supervision of Michel Goemans. Subsequently he spent a year at Microsoft Research in Redmond where this work originated. Matroids and submodular functions have pervaded his research ever since.