

The allocation problem with submodular utility functions (preliminary version)

Uriel Feige and Jan Vondrak

February 6, 2006

Abstract

We consider the problem of maximizing welfare when allocating items to players with submodular utility functions. Our main result is a way of rounding any fractional solution to a linear programming relaxation to this problem so as to give a feasible solution of welfare at least $1 - 1/e + \epsilon$ of the value of the fractional solution, where $\epsilon > 0$ is some fixed constant. This resolves the question of whether the previous approximation ratio of $1 - 1/e$ (for which there now are three different proofs, one in Dobzinski and Schapira [Soda 2006], one in Feige [2006], and one mentioned in the current paper) is best possible.

We also show that the maximum welfare problem with submodular utilities is NP-hard to approximate within a ratio better than $\rho < 1$, and 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 maximum welfare.

1 Introduction

We consider the following problem. There are m items and n players. A feasible allocation allocates every item to at most one player. For every player i , her utility w_i depends only on the set S of items that she receives. Utility functions are nonnegative (namely, $w_i(S) \geq 0$ for every set S of items), and monotone ($w_i(S) \leq w_i(T)$ when $S \subset T$). Let S_i be the set of items allocated to player i . The goal is to find a feasible allocation that maximizes social welfare, namely, maximizes $\sum_i w_i(S_i)$.

This problem is sometimes referred to as *combinatorial auctions*, in which case the players are referred to as *bidders*. If there are no further restrictions on utility functions, the problem is NP-hard, and moreover, is NP-hard to approximate within any reasonable factor. Following [9], research has concentrated on restricted classes of utility functions. See further details and results in [2, 3, 6]. In this paper we consider the case when utility functions are *submodular*, which was also the focus of some earlier work [9, 3, 8].

Submodularity may be thought of as a discrete analog of convexity, and has several equivalent definitions. We list three such definitions.

- **The standard definition.** $w(S) + w(T) \geq w(S \cup T) + w(S \cap T)$ for all sets S and T .
- **Decreasing marginal value.** $w(T \cup \{j\}) - w(T) \leq w(S \cup \{j\}) - w(S)$ for all items j when $S \subset T$.

- **Subadditive marginal value.** $w(T \cup S \cup S') - w(T) \leq (w(T \cup S) - w(T)) + (w(T \cup S') - w(T))$ for all sets T, S, S' .

Before presenting known results for approximating the maximum welfare when utility functions are submodular (a problem that we shall call *max submodular welfare*), we briefly discuss the issue of how utility functions are represented. An explicit representation of utility functions, specifying the value of every set, might take space which is exponential in m . Hence in some cases one assumes that utility functions are not represented explicitly, but rather that a player may be required to answer queries regarding her utility function. For a utility function w , a *value query* specifies a set S and is answered by the value $w(S)$. A *demand query* specifies a list of nonnegative prices p_j for all items, and is answered by a set S that maximizes $w(S) - \sum_{j \in S} p_j$. Answering a demand query requires a player to solve an optimization problem, and hence demand queries are considered to be more costly than value queries. Other type of queries may also be considered (and in fact will be considered in this paper).

Lehmann, Lehmann and Nisan [9] provide an approximation ratio of $1/2$ for the max submodular welfare problem, using a simple greedy algorithm in which players need to answer only demand queries. A randomized version of this algorithm is shown in [3] to give a somewhat improved approximation ratio of $n/(2n - 1)$. It is shown in [8] that if only value queries are allowed, then the max submodular welfare problem cannot be approximated within a ratio better than $1 - 1/e$ (up to low order terms).

Several earlier works [2, 3, 6] considered the following linear programming relaxation of the problem, that we call the *welfare maximizing LP*. $x_{i,S}$ is intended to be an indicator variable that specifies whether player i gets set S .

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. It can be solved optimally in polynomial time, assuming that there is an oracle that can answer demand queries. See [11, 2] for more details.

The integrality gap of this LP is known (up to low order terms) for classes of utility functions that are more general than submodular. For subadditive utility functions, it is $1/2$ [6], and for fractionally subadditive utility functions (see Definition 2.1) it is $1 - (1 - 1/n)^n$ which tends from above to $1 - 1/e$ as n grows [3, 6]. For submodular utility functions, which is a less general class of utility functions, the positive results still apply (the integrality ratio is no worse than $1 - 1/e$), but the negative results do not. An example with $n = 2$ and integrality gap $7/8$ is given in [3].

It was known that even with demand queries, the max submodular welfare problem is NP-hard. However, prior to this work it was not known whether it has a PTAS. The problem when even more general classes of queries are allowed was studied in the context of communication complexity [11], but we are not aware of previous results in the context of hardness of approximation.

1.1 Our results

Our main result is the following.

Theorem 1.1. *There is some universal constant $\rho > 1 - 1/e$ and a randomized rounding procedure for the welfare maximizing LP, such that given any instance of the max submodular welfare problem and any feasible solution to the welfare maximizing LP, the rounding procedure produces a feasible allocation whose expected total welfare is at least a ρ fraction of the value of the feasible solution.*

Our rounding procedure is oblivious in the sense of [6]: its only input is a feasible solution to the LP, and it need not know anything about the actual utility functions of the players.

The following corollary follows from combining Theorem 1.1 with the fact that demand queries suffice in order to find an optimal fractional solution to the welfare maximizing LP.

Corollary 1.2. *The max submodular welfare problem can be approximated in random polynomial time within a ratio of $\rho > 1 - 1/e$ in a computational model that allows demand queries.*

Recall that in contrast, when only value queries are allowed it is NP-hard to approximate the max submodular welfare problem within a ratio better than $1 - 1/e$ [8]. Likewise, when utility functions may belong to a somewhat more general class, namely, can be fractionally subadditive (or equivalently, XOS), then even with demand queries an approximation ratio better than $1 - 1/e$ is NP-hard [2].

Theorem 1.1 is the main (and certainly most difficult) result in this paper. In the process of proving it, we prove a few other related results, and they will be mentioned in Section 1.2.

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 1.1 with the hardness results of [8] clearly shows that the best approximation ratio for max 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? Previous 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 for this case.) In an attempt to capture all reasonable query models simultaneously, we consider very restricted types of utility functions.

Constant size utility 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. In particular, the welfare maximizing LP then has only a polynomial number of "relevant" variables, and hence can be solved efficiently.

Theorem 1.3. *The maximum 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 [2]), 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 functions, 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.

1.2 Overview of techniques

The proof of Theorem 1.3 involves fairly standard techniques for proving hardness of approximation results. The requirement that utility functions are of constant size simplifies the proof rather than making it more difficult, because it makes it easier to complete a partly defined utility function into a submodular function.

The proof of Theorem 1.1 considerably more challenging, and this section will attempt to present the thought process that led us to our new rounding procedure.

First, let us recall how one achieves an approximation ratio of $1 - 1/e$ (or in fact, $1 - (1 - 1/n)^n$) for the max submodular welfare problem. One uses the given feasible solution to the welfare maximizing 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 for some player, $\sum_S x_{i,S} < 1$, then it may happen that the player will select no tentative set at all.) Per player, the utility after this step is in expectation equal to its contribution to the value of the fractional solution of the welfare maximizing 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 [3], contention resolution is based on further queries to the players in order to determine which player will benefit most from getting the contended item. In [6], 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. Adding a third step to the rounding procedure, in which these items are allocated, would potentially allow one to improve over the $1 - 1/e$ ratio.

Somewhat surprisingly, one can design instances (see Section 2.2) where the utility functions are submodular, and regardless of how contention resolution is performed, and of how items outside the tentative sets are allocated, one cannot obtain an approximation ratio better than $1 - 1/e$.

We show (though the proof is omitted from the current version of this paper) that 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) achieves an approximation ratio of $1 - 1/e$ (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 $1 - 1/e$. However, again, this is not true.

The new rounding procedure that we design (that does improve over $1 - 1/e$) is also based on taking the best of two rounding procedures. (More accurately, it is a probability distribution over two rounding procedures, because taking the best of two outcomes would not give an oblivious rounding procedure.) One of them is a variation and simplification of the one used in [6] (see discussion in Section 1.3). This rounding procedure has the following property. Call a fractional solution to the LP *unbalanced* if for a "typical" item j , there is some player i (that may depend on j) for which $\sum_{S|j \in S} x_{i,S} > \epsilon$, where $\epsilon > 0$ is some fixed constant. Then the approximation ratio provided by our variation is $1 - 1/e + \epsilon^{O(1)}$. The second rounding procedure that we introduce handles the case when the fractional solution is balanced, and this is the most innovative part of our paper.

The main structural difference between our second 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, and the proof that it produces an approximation

ratio better than $1 - 1/e$ (when the fractional solution is balanced) uses algebraic manipulations that might not be easy to follow. To help the reader see the reasoning behind our new rounding procedure, we break its derivation into five stages.

The first stage is perhaps the most instructive one, as it addresses a simple and relatively easy to understand case. There are only two players, and the fractional solution is half integral. The goal is to design a rounding procedure that improves over 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.

The second stage 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 he/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 $3/4$. Specifically, we obtain a $7/9$ -approximation in this case.

The third stage combines the balanced and unbalanced case for two players, to obtain an algorithm for the general submodular welfare problem with two players. This deteriorates the approximation ratio slightly; we obtain a $13/17$ -approximation, which still beats $3/4$, however.

The fourth stage is the rounding procedure for the balanced case with n players, and is the key part of the proof of Theorem 1.1. It is based on ideas used for the two-player case, but again, with added complications.

In the fifth stage, the balanced and the unbalanced case are combined to obtain an approximation ratio strictly above $1 - 1/e$.

Though it may seem reasonable that every new stage can be accomplished based on previous stages, it was the experience of the authors that every stage required substantially new ideas in addition to those of previous stages.

1.3 Optimal balanced contention resolution

A key component in previous $1 - 1/e$ approximation ratios for max submodular welfare (and more generally, also for fractionally subadditive utility functions) 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 [6] 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 *optimal balanced contention resolution*. We now describe this problem.

There are n players and one item. Every player i is associated with a probability $0 \leq p_i \leq 1$ 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 . This will be called balanced contention resolution. Among all balanced 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 $\frac{p_i}{\sum_j p_j} (1 - \prod_j (1 - p_j))$. A balanced contention resolution method will be called *optimal* if indeed this maximum is attained by it. In previous work [6], optimal balanced contention resolution methods were designed for some special cases. In this work, we prove the following general lemma.

Lemma 1.4. *There is an explicit optimal balanced contention resolution method. More specifically, there is a simple formula that for every set A and player $i \in A$, determines the probability with which the item is given to i conditioned on A being the set of players that reach step 2. This formula depends on the size of A , on the values of p_j for those $j \in A$, and on the sum $\sum_{j=1}^n p_j$.*

See Lemma 3.1 for the proof of Lemma 1.4.

2 The allocation problem for two players

In this section, we start working towards the goal of proving Theorem 1.1. 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 1.1 but we consider it instructive for the reader to understand this simpler case before proceeding to the proof of Theorem 1.1. Also, our techniques are really geared to the case of two rather than n players. The improvement we achieve here (from $3/4$ to $13/17$) is relatively significant, as opposed to the purely theoretical improvement in Theorem 1.1.

In the setting with two players, it is not very difficult to achieve an approximation factor of $3/4$ (even assuming only fractional subadditivity), as shown in [6]. Since our improvements are built on top of the $3/4$ -approximation algorithm, let's review it 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 $\sum_S x_{1,S} \leq 1$, 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 [6]). For this purpose, we use a random “splitting set” X which contains each item j with probability p_j .

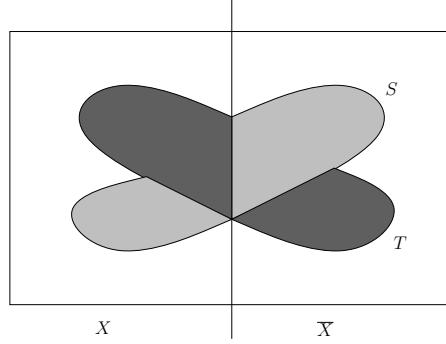


Figure 1: Algorithm 1.

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.

Before analysing Algorithm 1, let us introduce a class of utility functions that is more general than submodular functions (see more details in [6]).

Definition 2.1. A function w is fractionally subadditive if $w(S) \leq \sum \alpha_i w(T_i)$ with $0 \leq \alpha_i \leq 1$ for all i , whenever the following condition holds: for every item $j \in S$, $\sum_{i|j \in T_i} \alpha_i \geq 1$. (Namely, 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 .)

The key to the analysis of Algorithm 1 is the following lemma stated in [6].

Lemma 2.2. Let $p \in [0, 1]$ and $w : 2^{[m]} \rightarrow \mathbb{R}_+$ fractionally subadditive. For a set S , consider a probability distribution over subsets $S' \subseteq S$ such that each element of S is included in S' with probability at least p . Then

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

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 2.2,

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

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 $\frac{3}{4} \sum_T x_{2,T} w_2(T)$, since any element appears in $S \cap \bar{X}$ 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 [6], 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 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 $T_1 = \{a, c\}$, $T_2 = \{a, d\}$. 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 (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 $3/4 \cdot LP$. Moreover, even allocating the remaining item does not help! Regardless of who gets the last item, the objective value is still only $3/4 \cdot LP$. 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's 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 3 (5/6 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 $10/3 = 5/6 \cdot LP$.

Examples for n players. These examples can be generalized to n players, by considering n^n items arranged in a hypercube $\{1, 2, \dots, n\}^n$. The sets desired by player i are the layers $H_{i,j}$ orthogonal to dimension i .

We can define XOS utility functions by considering the maximum number of items obtained by player i in some layer $H_{i,j}$ for $j = 1, 2, \dots, n$. This gives an integrality gap of $1 - (1 - 1/n)^n$.

Similarly, we can define submodular utility functions by considering the projection of the set of items obtained by player i onto a hyperplane orthogonal to dimension i . This does not give any

integrality gap, but again it implies that $1 - (1 - 1/n)^n$ cannot be improved by if we start by allocating one set $H_{i,j}$ to each player i . This example can also be modified to yield an integrality gap of 0.782 for submodular functions, but we leave the details outside of this paper.

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*: $x_{i,S} \in \{0, \frac{1}{2}, 1\}$.

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's 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 (possibly different) partitions of the set of all items.

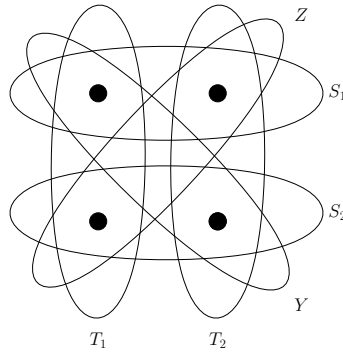


Figure 2: A half-integral solution.

It is necessary to allow the option to combine items from the two sets desired by each player, as described in Example 2. Any solution which starts by allocating one set to each player and resolving conflicts cannot beat 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, 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 $\frac{5}{12}(w_1(S_1) + w_1(S_2))$. We get a similar estimate for player 2, so the expected value of our integral solution is at least $\frac{5}{6}LP$.

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*. Observe that this is the worst case for Algorithm 1. If the fractional solution is significantly unbalanced, then even Algorithm 1 performs better than $3/4$, since items are removed from each player with probabilities smaller than $1/4$.

So, let's assume for now that the fractional solution is 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 from the fractional solution. Let S denote a random set sampled from the distribution of player 1 and T a random set sample 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 $\mathbf{E}[w_2(\bar{S})] \geq \mathbf{E}[w_2(T \cap \bar{S})] \geq 1/2$, 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's consider another set: $Z = (T \cap S) \cup (T' \cap \bar{S})$ 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:

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

In other words, if \bar{S} presents no improvement on the average over $T' \cap \bar{S}$, 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 \bar{T} instead of $S' \cap \bar{T}$, then $Y = (S \cap T) \cup (S' \cap \bar{T})$ is by submodularity just as good as S or S' . Note that each player uses the other player's set to "combine" two of his/her sets. Let's be more specific and define which of the other player's sets is used for this purpose:

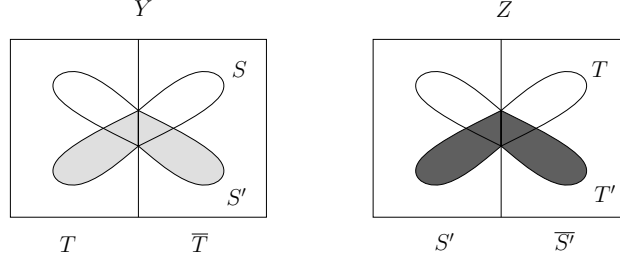


Figure 3: Diagonal sets Y and Z .

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

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

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

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

We assign items randomly based on the following table:

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

Theorem 2.3. *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, \bar{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.

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

By submodularity,

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

Now we use the linearity of expectation and Lemma 2.2: each item appears in $(S' \cup T') \cap X \cap T$ with probability $3/16$, and in \bar{T} with probability $1/2$. Therefore

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

The expected profit of player 1 is

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

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

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

Submodularity yields

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

Finally, we apply Lemma 2.2:

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

and the rest follows as for player 1. \square

For two players with a balanced fractional solution, we also have a slightly improved algorithm which has an approximation factor $7/9 \doteq 0.777$ (as compared to $37/48 \doteq 0.771$). Its analysis is slightly more involved; it uses a new trick, the application of Lemma 2.2 to marginal values of subsets. This trick and the structure of the next algorithm will be useful for our final solution for n players.

Algorithm 2'. (7/9-approximation for 2 balanced players)

- Let player 1 sample independently random sets S, S' .
- Let player 2 sample independently random sets T, T' .
- Let $Y = (S \cap T) \cup (S' \cap \bar{T})$, $Z = (T \cap S) \cup (T' \cap \bar{S})$.
- Let $Y' = (S \cap T') \cup (S' \cap \bar{T}')$, $Z' = (T \cap S') \cup (T' \cap \bar{S}')$.

We assign items according to the following table:

Probability	5/18	5/18	4/18	4/18
Player 1	S'	\bar{T}	Y'	$Y \setminus Z'$
Player 2	\bar{S}'	T	$Z \setminus Y'$	Z'

Theorem 2.4. For any balanced fractional solution $x_{i,S}$, Algorithm 2' gives expected profit at least $7/9 \sum_S x_{i,S} w_i(S)$ to each player i .

Proof. Let's define $\alpha = \mathbf{E}[w_1(S)]$, $\beta = \mathbf{E}[w_2(T)]$, $\gamma = \mathbf{E}[w_1(\bar{T})]$ and $\delta = \mathbf{E}[w_2(\bar{S})]$. I.e., $\alpha + \beta = \sum_{i,S} x_{i,S} w_i(S)$ is the value of the fractional solution, while $\alpha + \delta$ and $\beta + \gamma$ are the expected values achieved by the first two allocation schemes. It remains to estimate the expected profit obtained in the last two schemes. By submodularity and linearity of expectation, we get

$$\mathbf{E}[w_2(Z')] + \mathbf{E}[w_2(\bar{S}')] \geq \mathbf{E}[w_2(T \cup \bar{S}')] + \mathbf{E}[w_2(T' \cap \bar{S}')] \geq \mathbf{E}[w_2(T)] + \frac{1}{2} \mathbf{E}[w_2(T')]$$

using Lemma 2.2. I.e.,

$$\mathbf{E}[w_2(Z')] \geq \frac{3}{2} \beta - \delta.$$

It is a little bit more complicated to estimate the value of $Y \setminus Z'$. We use the fact that $Y \setminus Z'$ simplifies to $(S \cap T) \setminus (S' \cup T') \cup (S' \cap \bar{T})$, and then by submodularity:

$$\begin{aligned} \mathbf{E}[w_1(Y \setminus Z')] + \mathbf{E}[w_1(\bar{T})] &= \mathbf{E}[w_1(((S \cap T) \setminus (S' \cup T')) \cup (S' \cap \bar{T}))] + \mathbf{E}[w_1(\bar{T})] \\ &\geq \mathbf{E}[w_1(((S \cap T) \setminus (S' \cup T')) \cup \bar{T})] + \mathbf{E}[w_1(S' \cap \bar{T})] \\ &\geq \mathbf{E}[w_1(((S \cap T) \setminus (S' \cup T')) \cup \bar{T})] + \frac{1}{2} \alpha. \end{aligned}$$

To estimate the last expectation, let's consider $g(A) = w_1(A \cup \bar{T}) - w_1(\bar{T})$, a function on subsets $A \subseteq T$ which gives the marginal value of adding some items to \bar{T} . This is also a submodular function and therefore Lemma 2.2 applies to it; in particular, conditioning on S, T , we get

$$\mathbf{E}_{S', T'}[g((S \cap T) \setminus (S' \cup T')) \mid S, T] \geq \frac{1}{4} g(S \cap T)$$

because $S' \cup T'$ contains each element with probability 3/4. This means,

$$\mathbf{E}[w_1(((S \cap T) \setminus (S' \cup T')) \cup \bar{T}) - w_1(\bar{T}) \mid S, T] \geq \frac{1}{4} (w_1((S \cap T) \cup \bar{T}) - w_1(\bar{T})) \geq \frac{1}{4} (w_1(S) - w_1(\bar{T})),$$

$$\mathbf{E}[w_1(((S \cap T) \setminus (S' \cup T')) \cup \bar{T}) - w_1(\bar{T})] \geq \frac{1}{4} \mathbf{E}[w_1(S) - w_1(\bar{T})] = \frac{1}{4} (\alpha - \gamma)$$

and therefore

$$\mathbf{E}[w_1(Y \setminus Z')] \geq \frac{1}{4}(\alpha - \gamma) + \frac{1}{2}\alpha = \frac{3}{4}\alpha - \frac{1}{4}\gamma.$$

By symmetry, we also get $\mathbf{E}[w_1(Y')] \geq \frac{3}{2}\alpha - \gamma$ and $\mathbf{E}[w_2(Z \setminus Y')] \geq \frac{3}{4}\beta - \frac{1}{4}\delta$. By combining all of the above, we obtain that the total expected profit of player 1 is

$$\begin{aligned} & \frac{5}{18}\mathbf{E}[w_1(S)] + \frac{5}{18}\mathbf{E}[w_1(\bar{T})] + \frac{4}{18}\mathbf{E}[w_1(Y \setminus Z')] + \frac{4}{18}\mathbf{E}[w_1(Y')] \\ & \geq \frac{5}{18}\alpha + \frac{5}{18}\gamma + \frac{4}{18}\left(\frac{3}{4}\alpha - \frac{1}{4}\gamma\right) + \frac{4}{18}\left(\frac{3}{2}\alpha - \gamma\right) = \frac{7}{9}\alpha. \end{aligned}$$

By symmetry, the expected profit of player 2 is at least $\frac{7}{9}\beta$. \square

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 2.2.

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

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

Then

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

This implies Lemma 2.2 as a special case, since we can have X contain each item with probability $p_j = p$ and then $\mathbf{E}[w(S')] \geq \mathbf{E}[w(S \cap X)] \geq \sum_j p_j \sigma_j = p w(S)$.

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

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

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

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

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.

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

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

$$\sum_j (1 - p_j(1 - p_j))\sigma_j + \sum_j (1 - p_j(1 - p_j))\tau_j \geq \frac{3}{4}LP + \sum_{j \in U} (\frac{1}{4} - p_j(1 - p_j))(\sigma_j + \tau_j) \geq (\frac{3}{4} + \epsilon^3)LP.$$

- If the contribution of unbalanced items is negligible, $\sum_{j \in U} (\sigma_j + \tau_j) < \epsilon \cdot LP$, then let's remove the unbalanced items. Also, we scale the remaining fractional solution by $1/(1 + 2\epsilon)$ 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 $(1 - 3\epsilon)$ compared to the original fractional solution. Then we run Algorithm 2 on this balanced fractional solution which yields expected value at least $\frac{37}{48}(1 - 3\epsilon) \cdot LP$.

Choosing for example $\epsilon = 1/150$ gives an approximation factor slightly better than $3/4$. However, we can do better than this, by analyzing more carefully how Algorithm 2 performs on unbalanced items. For balanced items ($p_j = 1/2$) Algorithm 1 gives factor $3/4$, while Algorithm 2 gives factor $37/48$. On the other hand, observe that items which are extremely unbalanced ($p_j \rightarrow 0$) are recovered by Algorithm 1 with probability almost 1, whereas Algorithm 2 recovers them with probability only $2/3$. The best we can hope for (by combining these two algorithms) is to take a convex combination which optimizes the minimum of these two cases. This convex combination takes Algorithm 1 with probability $5/17$ and Algorithm 2 with probability $12/17$ which yields a factor of $13/17$ in the two extreme cases. We show that this is indeed possible in the entire range of probabilities p_j . Since Algorithm 2 turns out to favor the player whose share (p_j or q_j) is higher than $1/2$, we offset this advantage by generating a splitting set X which gives more advantage to the player with a smaller share.

Algorithm 3. (13/17-approximation for 2 players)

- Let player 1 sample independently random sets S, S' .
- Let player 2 sample independently random sets T, T' .
- Let

$$f(x) = \frac{4x}{(1-x)(9-4x(1-x))}.$$

Generate independently a random set X containing item j with probability $\phi(p_j) = f(p_j)$ for $p_j \leq 1/2$, or with probability $\phi(p_j) = 1 - f(1 - p_j)$ for $p_j > 1/2$.

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

We assign items randomly based on the following table:

Probability	5/17	4/17	4/17	4/17
Player 1	$S \setminus (T \cap X)$	S'	\bar{T}	$Y \setminus (Z \cap X)$
Player 2	$T \setminus (S \cap \bar{X})$	S'	T	$Z \setminus (Y \cap \bar{X})$

Theorem 2.6. *For 2 players with an arbitrary fractional solution, Algorithm 3 yields expected profit at least $13/17 \sum_S x_{i,S} w_i(S)$ for player i .*

Proof. By definition, we have $\mathbf{E}[w_1(S)] = \mathbf{E}[w_1(S')] = \sum_j \sigma_j$ and $\mathbf{E}[w_2(T)] = \mathbf{E}[w_2(T')] = \sum_j \tau_j$. Now consider the first allocation scheme. The definition of set X is symmetric in the sense that $\phi(1 - p_j) = 1 - \phi(p_j)$, i.e. if $p_j + q_j = 1$, the set is generated equivalently from the point of view of either player. Since item j appears in $T \cap X$ with probability $q_j \phi(p_j) \leq (1 - p_j) \phi(p_j)$, and in $S \cap \bar{X}$ with probability $p_j(1 - \phi(p_j)) = p_j \phi(1 - p_j)$, Lemma 2.5 implies

$$\begin{aligned} \mathbf{E}[w_1(S \setminus (T \cap X))] &\geq \sum_j (1 - (1 - p_j) \phi(p_j)) \sigma_j, \\ \mathbf{E}[w_2(T \setminus (S \cap \bar{X}))] &\geq \sum_j (1 - p_j \phi(1 - p_j)) \tau_j. \end{aligned}$$

To estimate the combined profit of the remaining allocation schemes, we use submodularity and Lemma 2.5:

$$\begin{aligned} \mathbf{E}[w_1(Y \setminus (Z \cap X))] + \mathbf{E}[w_1(\bar{T})] &\geq \mathbf{E}[w_1((S \cap T) \setminus ((S' \cup T') \cap X) \cup \bar{T})] + \mathbf{E}[w_1(S' \cap \bar{T})] \\ &\geq \mathbf{E}[w_1(S \setminus (T \cap X \cap (S' \cup T')))] + \mathbf{E}[w_1(S' \cap \bar{T})] \\ &\geq \sum_j (1 - q_j \phi(p_j)(1 - (1 - p_j)(1 - q_j))) \sigma_j + \sum_j (1 - q_j) \sigma_j \\ &\geq \sum_j (1 + p_j - (1 - p_j) \phi(p_j)(1 - p_j(1 - p_j))) \sigma_j. \end{aligned}$$

The total expected profit of player 1 is:

$$\begin{aligned} &\frac{5}{17} \mathbf{E}[w_1(S \setminus (T \cap X))] + \frac{4}{17} \mathbf{E}[w_1(S)] + \frac{4}{17} \mathbf{E}[w_1(\bar{T})] + \frac{4}{17} \mathbf{E}[w_1(Y \setminus (Z \cap X))] \\ &\geq \frac{5}{17} \sum_j (1 - \phi(p_j)(1 - p_j)) \sigma_j + \frac{4}{17} \sum_j (2 + p_j - (1 - p_j) \phi(p_j)(1 - p_j(1 - p_j))) \sigma_j \\ &= \frac{13}{17} \sum_j \sigma_j + \frac{1}{17} \sum_j (4p_j - \phi(p_j)(1 - p_j)(9 - 4p_j(1 - p_j))) \sigma_j. \end{aligned}$$

We show that the last sum is nonnegative. It can be verified that the function

$$f(x) = \frac{4x}{(1-x)(9-4x(1-x))}$$

is increasing and convex on the interval $(0, 1)$. Also, $f(1/2) = 1/2$ and by convexity $f(p_j) + f(1 - p_j) \geq 1$. We have $\phi(p_j) = f(p_j)$ for $p_j \in [0, \frac{1}{2}]$ and $\phi(p_j) = 1 - f(1 - p_j)$ for $p_j \in [\frac{1}{2}, 1]$; i.e., $\phi(p_j) \leq f(p_j)$

for any $p_j \in (0, 1)$. Consequently, $\phi(p_j)(1 - p_j)(9 - 4p_j(1 - p_j)) \leq 4p_j$ and so player 1 gets expected profit at least $\frac{13}{17} \sum_j \sigma_j$.

For player 2, we get the following:

$$\begin{aligned}
\mathbf{E}[w_2(Z \setminus (Y \cap \bar{X}))] + \mathbf{E}[w_2(\bar{S}')] &\geq \mathbf{E}[w_2(((S' \cap T) \setminus (Y \cap \bar{X})) \cup \bar{S}')] + \mathbf{E}[w_2((T' \cap \bar{S}') \setminus (Y \cap \bar{X}))] \\
&\geq \mathbf{E}[w_2(T \setminus (S \cap S' \cap \bar{X}))] + \mathbf{E}[w_2(T' \cap \bar{S}' \setminus (S \cap T \cap \bar{X}))] \\
&\geq \sum_j (1 - \phi(1 - p_j)p_j^2)\tau_j + \sum_j (1 - p_j)(1 - \phi(1 - p_j)p_j q_j)\tau_j \\
&\geq \sum_j (1 + (1 - p_j) - \phi(1 - p_j)p_j(1 - p_j(1 - p_j)))\tau_j.
\end{aligned}$$

This is just like the expression for player 1 after substituting $p_j \rightarrow 1 - p_j$. The same analysis gives that the total expected profit of player 2 is at least $\frac{13}{17} \sum_j \tau_j$. \square

3 The allocation problem for n players

Our algorithms for allocation between two players allow us to achieve a (definite, albeit small) improvement over $1 - 1/e$ in the general case. Roughly speaking, we divide the players into two groups and treat them as two super-players, 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 3 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.

In order to distribute items within each group of players, we use the following procedure. This is a refinement and simplification of the previously used techniques for fractionally subadditive utility functions [6].

The optimal technique to resolve conflicts. 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

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

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.

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

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

This is the best possible value of ρ , since the total probability that the item is allocated should be $\rho \sum_i p_i$ and this cannot be higher than the probability that somebody requests the item, $1 - \prod_i (1 - p_i)$.

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 $\Pr[A = \{k\}](1 - r_{\{k\},k}) = \Pr[A = \{k\}] p_k / \sum_i p_i$ which makes the total probability that player k obtains the item equal to

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

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's 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's write $\mathbf{E}[r_{A,k} \mid k \in A]$ as a weighted sum over all possible subsets $A' \subseteq B$:

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

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 a set A' and player $i \in B \setminus A'$, 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, we obtain from (1) that player k receives the item with probability

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

This gives immediately an approximation factor of $1 - (1 - 1/n)^n > 1 - 1/e$, assuming only fractional additivity.

Algorithm 4. $((1 - (1 - 1/n)^n)$ -approximation for n players)

- Let each player i sample a set S_i from his/her probability distribution.
- Using the optimal technique to resolve conflicts, obtain disjoint sets S_i^* and allocate them to the respective players.

Lemma 3.2. *For n players with fractionally subadditive utility functions, Algorithm 4 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 3.1, conditioned on player i competing for item j , the item is allocated to him/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 $\sum_{i=1}^n y_{ij} \leq 1$. 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 2.2. \square

This already improves the approximation factor of $1 - 1/e$ for any fixed number of players. However, our goal is to obtain an absolute constant larger than $1 - 1/e$, independent of n .

3.1 n balanced infinitesimal players

First, let's assume again 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 Algorithm 4. (This can be assumed WLOG if all the players have the same utility function.) 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 $(1 - 1/n)^n \doteq e^{-1}$, $(1 - 1/n)^{n/2} \doteq e^{-1/2}$, etc. We split the players evenly into groups \mathcal{A}, \mathcal{B} such that

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

For a collection of sets $\{S_i : i \in \mathcal{A}\}$ sampled by players in one group, we will use Lemma 3.1 to make the sets disjoint; we call this “resolving conflicts among a group of players”. Recall that players in a group \mathcal{A} such that $\sum_{i \in \mathcal{A}} y_{ij} = 1/2$ can resolve conflicts in such a way that each requested item is allocated with probability $(1 - \prod_{i \in \mathcal{A}} (1 - y_{ij})) / (\sum_{i \in \mathcal{A}} y_{ij}) \geq 2(1 - e^{-1/2}) \doteq 0.787$. This is significantly better than $1 - e^{-1} \doteq 0.632$; 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 5. (0.645-approximation for n balanced infinitesimal players)

- 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} resolve conflicts among S_i to obtain disjoint sets S_i^* . Similarly, resolve conflicts among S'_i to obtain disjoint sets $S_i'^*$.
- Let the players in \mathcal{B} resolve conflicts among T_i to obtain disjoint sets T_i^* . Similarly, resolve conflicts among T'_i to obtain disjoint sets $T_i'^*$.
- Let $Y_i^* = (S_i^* \cap V) \cup (S_i'^* \cap \bar{V})$, $Z_i^* = (T_i^* \cap U) \cup (T_i'^* \cap \bar{U})$.
- Let $Y_i'^* = (S_i^* \cap V') \cup (S_i'^* \cap \bar{V}')$, $Z_i'^* = (T_i^* \cap U') \cup (T_i'^* \cap \bar{U}')$.

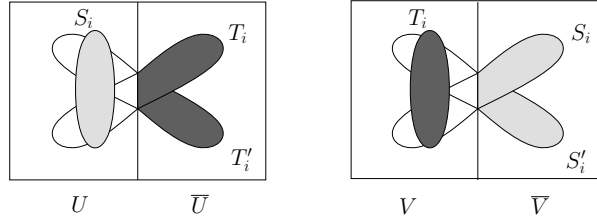


Figure 4: Allocation schemes 1 and 2.

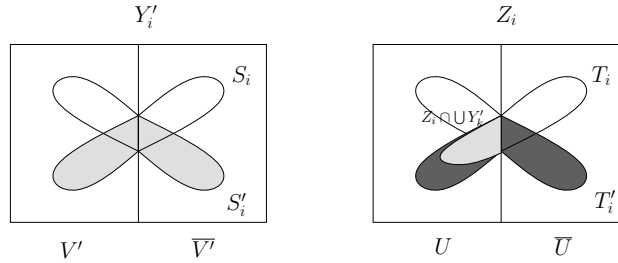


Figure 5: Allocation scheme 3.

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 $Z_i^* \setminus \bigcup_{i \in \mathcal{A}} Y_i'^*$.

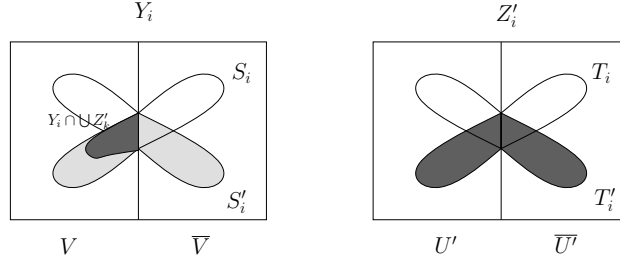


Figure 6: Allocation scheme 4.

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 conflicts inside each group of players). In the following, we also use $Y_i = (S_i \cap V) \cup (S'_i \cap \bar{V})$, $Y'_i = (S_i \cap V') \cup (S'_i \cap \bar{V}')$, etc., to denote sets before resolving conflicts. Taking the union over each group, we get the same set regardless of resolving conflicts or not: $\bigcup_{i \in \mathcal{A}} S_i = \bigcup_{i \in \mathcal{A}} S_i^* = U$, $\bigcup_{i \in \mathcal{A}} Y_i = \bigcup_{i \in \mathcal{A}} Y_i^* = (U \cap V) \cup (U' \cap \bar{V})$, etc.

Intuition. If we use only the first two schemes, we get an approximation factor of 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 conflicts 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 3.3. *For n players with a balanced fractional solution and $n \rightarrow \infty$, Algorithm 5 yields expected profit at least $0.645 \sum_S x_{i,S} w_i(S)$ for player i .*

Proof. We set 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 3.1 allows us to resolve conflicts 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 $Y_i = (S_i \cap V) \cup (S'_i \cap \bar{V})$; however, the resolution of conflicts 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's just consider $S_i \setminus V$. Each item survives outside of V with probability $(1 - 1/n)^{n/2} \doteq e^{-1/2}$, 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's write $\tilde{U} = \bigcup_{k \in \mathcal{A} \setminus \{i\}} S_k$; we can use this instead of $U = \tilde{U} \cup S_i$ 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's 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 $\prod_{k \in \mathcal{A} \setminus \{i\}} (1 - y_{kj}) \doteq e^{-1/2}$. 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's turn to the third allocation scheme. Player i requests $Y'_i = (S_i \cap V') \cup (S'_i \cap \bar{V}')$ which by submodularity satisfies

$$\mathbf{E}[w_i(Y'_i)] + \mathbf{E}[w_i((S_i \cup S'_i) \cap \bar{V}')] \geq \mathbf{E}[w_i(Y'_i \cup (S_i \cap \bar{V}'))] + \mathbf{E}[w_i(S'_i \cap \bar{V}')] \geq \mathbf{E}[w_i(S_i)] + \mathbf{E}[w_i(S'_i \cap \bar{V}')],$$

i.e.

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

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 $Y_i = (S_i \cap V) \cup (S'_i \cap \bar{V})$ and $\bigcup_{k \in \mathcal{B}} Z'_k = (V \cap U') \cup (V' \cap \bar{U}')$, and so

$$Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k = ((S_i \cap V) \cup (S'_i \cap \bar{V})) \setminus ((V \cap U') \cup (V' \cap \bar{U}')) = ((S_i \cap V) \setminus (U' \cup V')) \cup (S'_i \cap \bar{V}).$$

By submodularity,

$$\mathbf{E}[w_i(Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k)] + \mathbf{E}[w_i((S_i \cup S'_i) \cap \bar{V})] \geq \mathbf{E}[w_i(S_i \setminus (V \cap (U' \cup V')))] + \mathbf{E}[w_i(S'_i \cap \bar{V})]$$

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

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

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, conflicts are 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

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

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

3.2 Sketch of a small improvement in the general case

Let's convince the reader that a tiny (but constant) improvement over $1 - 1/e$ in the general case is possible. We have Algorithm 5 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 Algorithm 5 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's propose a variant of Algorithm 5 for the general case.

Algorithm 5'.

- Partition the players into groups \mathcal{A}, \mathcal{B} by assigning each player uniformly and independently to \mathcal{A} or \mathcal{B} . Define

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

- Now 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 $\tilde{x}_{i, S \cap X}$ where

$$\tilde{x}_{i, S \cap X} = x_{i, S} \Pr[X] = x_{i, S} \prod_{j \in X} \frac{1}{2z_j} \prod_{j \notin X} \left(1 - \frac{1}{2z_j} \right).$$

Then, we get $\tilde{y}_{ij} = \sum_{S: j \in S} \tilde{x}_{i, S} = y_{ij}/(2z_j)$ so that $\max(\sum_{i \in \mathcal{A}} \tilde{y}_{ij}, \sum_{i \in \mathcal{B}} \tilde{y}_{ij}) \leq 1/2$.

- Then run Algorithm 5 on the fractional solution $\tilde{x}_{i, S}$. Note that the fractional solution $\tilde{x}_{i, S}$ is "sub-balanced" rather than balanced, but this is sufficient for Algorithm 5 to work properly.

Let's fix a very small $\epsilon > 0$ and call an item "unbalanced" if some player requests it with probability $y_{ij} > \epsilon$. We claim that Algorithm 5' 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 $\frac{1}{2} \sum_{i=1}^n y_{ij}$; more precisely, their variance is

$$\text{Var} \left[\sum_{i \in \mathcal{A}} y_{ij} \right] = \sum_{i=1}^n \frac{1}{4} y_{ij}^2 \leq \frac{\epsilon}{4} \sum_{i=1}^n y_{ij} \leq \frac{\epsilon}{4}.$$

Therefore, the expected amount by which either sum exceeds $1/2$ is

$$\mathbf{E}[z_j - 1/2] \leq \mathbf{E} \left[\left| \sum_{i \in \mathcal{A}} y_{ij} - \frac{1}{2} \sum_{i=1}^n y_{ij} \right| \right] \leq \sqrt{\text{Var} \left[\sum_{i \in \mathcal{A}} y_{ij} \right]} = \frac{1}{2} \sqrt{\epsilon}.$$

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

$$\mathbf{E} \left[\frac{1}{2z_j} \right] \geq \frac{1}{2\mathbf{E}[z_j]} \geq \frac{1}{1 + \sqrt{\epsilon}}.$$

Moreover, the analysis of Algorithm 5 (which assumed infinitesimal values of y_{ij}) is quite precise for such a fractional solution. For $0 \leq y_{ij} \leq \epsilon$, we have $-y_{ij} \geq \log(1 - y_{ij}) \geq -y_{ij}/(1 - \epsilon)$, i.e.

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

Thus all the estimates in the analysis of Algorithm 5 are precise up to an error of $O(\epsilon)$. Accounting for the balancing step, we get a solution of expected value $(0.645 - O(\sqrt{\epsilon}))LP$.

If some items are unbalanced, then running Algorithm 5' might present a problem. However, then we gain by running Algorithm 4. 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. $\sum_i \sum_{j \in U} \sigma_{ij} \geq \epsilon \cdot LP$, then we use Algorithm 4. For each unbalanced item $j \in U$, since $y_{ij} > \epsilon$ for some i , Lemma 3.1 allocates the item to each player with conditional probability

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

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

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

- If the contribution of unbalanced items is negligible, $\sum_i \sum_{j \in U} \sigma_{ij} < \epsilon \cdot LP$, then let's remove the unbalanced items. This incurs a factor of $(1 - \epsilon)$ on the value of the fractional solution. Then we run Algorithm 5' which yields expected value at least $(0.645 - O(\sqrt{\epsilon})) \cdot LP$.

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

3.3 Our currently best algorithm in the general case

Let's analyze Algorithm 5' more precisely. For each item j , we measure the "granularity" of the fractional solution by $\sum_{i=1}^n y_{ij}^2$, which could range from $1/n$ (for a perfectly balanced solution) to 1 (when only one player requests item j). We show later that the analysis of Algorithm 5 can be carried through for any balanced fractional solution, with error terms depending on the granularity of the solution. But first, let's look at the balancing procedure.

Lemma 3.4. *Let player i have a distribution over sets S_i with expected value $\mathbf{E}[w_i(S_i)] = \sum_j \sigma_{ij}$ where*

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

Then after the balancing procedure executed for a given partitioning of players, the new marginal values for player i are

$$\tilde{\sigma}_{ij} = \mathbf{E}_{S_i, X}[w_i(S_i \cap X \cap [j]) - w_i(S_i \cap X \cap [j-1])]$$

such that averaging over random partitions,

$$\mathbf{E}_{(\mathcal{A}, \mathcal{B})}[\tilde{\sigma}_{ij}] \geq \frac{\sigma_{ij}}{1 + \sqrt{\sum_{i=1}^n y_{ij}^2}}.$$

Proof. Conditioned on a specific partition $(\mathcal{A}, \mathcal{B})$, the balancing procedure gives to player i a random set $S_i \cap X$ where $\Pr[j \in X] = 1/(2z_j)$. The proof of Lemma 2.5 implies that the new marginal values will be at least $\tilde{\sigma}_{ij} \geq \sigma_{ij}/(2z_j)$. Averaging over random partitions $(\mathcal{A}, \mathcal{B})$, and using $\mathbf{E}[1/z_j] \geq 1/\mathbf{E}[z_j]$, we get

$$\mathbf{E}_{(\mathcal{A}, \mathcal{B})}[\tilde{\sigma}_{ij}] \geq \frac{\sigma_{ij}}{2\mathbf{E}[z_j]}.$$

To estimate $\mathbf{E}[z_j]$, we use the second moment. Let Y_{ij} be independent random variables that take values 0 or y_{ij} with probability $1/2$, corresponding to player i being assigned to group \mathcal{A} or \mathcal{B} . We have $\sum_{i=1}^n \mathbf{E}[Y_{ij}] = 1/2$ and $z_j = \max(\sum_{i=1}^n Y_{ij}, 1 - \sum_{i=1}^n Y_{ij}) = 1/2 + |\sum_{i=1}^n Y_{ij} - 1/2|$. Due to independence, $\text{Var}[\sum_{i=1}^n Y_{ij}] = \sum_{i=1}^n \text{Var}[Y_{ij}] = \frac{1}{4} \sum_{i=1}^n y_{ij}^2$ and therefore

$$\mathbf{E}[z_j] = \frac{1}{2} + \mathbf{E}\left[\left|\sum_{i=1}^n Y_{ij} - \frac{1}{2}\right|\right] \leq \frac{1}{2} + \sqrt{\text{Var}\left[\sum_{i=1}^n Y_{ij}\right]} = \frac{1}{2} + \frac{1}{2}\sqrt{\sum_{i=1}^n y_{ij}^2}.$$

□

Thus, we are likely to get a good balanced solution for fractional solutions with low granularity. Now we apply Algorithm 5 to this modified fractional solution, and estimate the expected profit. In the analysis, we need the following bounds.

Lemma 3.5. *For any $y_{1j}, \dots, y_{nj} \geq 0$ such that $\sum_{i=1}^n y_{ij} \leq 1$,*

$$e^{-\sum_{i=1}^n y_{ij}} \left(1 - \sum_{i=1}^n y_{ij}^2\right) \leq \prod_{i=1}^n (1 - y_{ij}) \leq e^{-\sum_{i=1}^n y_{ij}} \left(1 - \frac{1}{2} \sum_{i=1}^n y_{ij}^2\right).$$

Proof. By taking logs and Taylor's series,

$$\begin{aligned}\log \prod_{i=1}^n (1 - y_{ij}) &= \sum_{i=1}^n \log(1 - y_{ij}) = -\sum_{i=1}^n y_{ij} - \frac{1}{2} \sum_{i=1}^n y_{ij}^2 - \frac{1}{3} \sum_{i=1}^n y_{ij}^3 - \dots, \\ \log \left(e^{-\sum_{i=1}^n y_{ij}} \left(1 - \sum_{i=1}^n y_{ij}^2 \right) \right) &= -\sum_{i=1}^n y_{ij} - \sum_{i=1}^n y_{ij}^2 - \frac{1}{2} \left(\sum_{i=1}^n y_{ij}^2 \right)^2 - \frac{1}{3} \left(\sum_{i=1}^n y_{ij}^2 \right)^3 - \dots, \\ \log \left(e^{-\sum_{i=1}^n y_{ij}} \left(1 - \frac{1}{2} \sum_{i=1}^n y_{ij}^2 \right) \right) &= -\sum_{i=1}^n y_{ij} - \frac{1}{2} \sum_{i=1}^n y_{ij}^2 - \frac{1}{8} \left(\sum_{i=1}^n y_{ij}^2 \right)^2 - \frac{1}{24} \left(\sum_{i=1}^n y_{ij}^2 \right)^3 - \dots\end{aligned}$$

To compare the first two series, we simply note that

$$\frac{1}{k} \left(\sum y_{ij}^2 \right)^k \geq \frac{1}{k} \sum y_{ij}^{2k} \geq \frac{1}{2k} \sum y_{ij}^{2k} + \frac{1}{2k+1} \sum y_{ij}^{2k+1}$$

and so the first inequality in the lemma follows.

To compare the first and the third series, observe that the first two terms match, while starting from the third we have to compare $\sum_i y_{ij}^k$ with $(\sum_i y_{ij}^2)^{k-1}$. We use Hölder's inequality

$$\left(\sum a_i^p \right)^{1/p} \left(\sum b_i^q \right)^{1/q} \geq \sum a_i b_i$$

with $a_i = y_{ij}^{k/(k-1)}$, $b_i = y_{ij}^{(k-2)/(k-1)}$, $p = k-1$ and $q = (k-1)/(k-2)$:

$$\left(\sum_{i=1}^n y_{ij}^k \right)^{\frac{1}{k-1}} \left(\sum_{i=1}^n y_{ij} \right)^{\frac{k-2}{k-1}} \geq \sum_{i=1}^n y_{ij}^{\frac{k-2}{k-1}} y_{ij}^{\frac{k}{k-1}} = \sum_{i=1}^n y_{ij}^2.$$

Using $\sum_{i=1}^n y_{ij} \leq 1$, we get $\sum_i y_{ij}^k \geq (\sum_i y_{ij}^2)^{k-1}$, and therefore the second inequality holds as well. \square

Lemma 3.6. *For n players with an arbitrary fractional solution, the balancing procedure followed by Algorithm 5 gives player i expected profit at least*

$$0.645 \sum_j \frac{1 - \sum_{i=1}^n y_{ij}^2}{1 + \sqrt{\sum_{i=1}^n y_{ij}^2}} \sigma_{ij}.$$

Proof. Let player i have a fractional solution of value $\mathbf{E}[w_i(S_i)] = \sum_j \sigma_{ij}$. We proved that after the balancing procedure, this solution is modified to a new (random) one with marginal values $\tilde{\sigma}_{ij}$ where $\mathbf{E}[\tilde{\sigma}_{ij}] \geq \sigma_{ij} / (1 + \sqrt{\sum_{i=1}^n y_{ij}^2})$. We denote the new fractional solution by $\tilde{x}_{i,S}$ and $\tilde{y}_{ij} = \sum_{S:j \in S} \tilde{x}_{i,S}$.

Now we apply Algorithm 5. We have to go through the analysis of Algorithm 5 more carefully, keeping in mind that the solution has finite granularity. We need to be especially cautious about estimates like $\prod(1 - y_{ij}) \geq e^{-\sum y_{ij}}$ which incur an error term depending on the granularity (see Lemma 3.5). Inequalities like $\prod(1 - y_{ij}) \leq e^{-\sum y_{ij}}$ hold without change, as well as $(1 - \prod(1 - y_{ij})) / \sum y_{ij} \geq 2(1 - e^{-1/2})$ for $\sum y_{ij} \leq 1/2$.

In the first allocation scheme, player $i \in \mathcal{A}$ receives

$$\mathbf{E}[w_i(S_i^*)] \geq \sum_j \frac{1 - \prod_{i \in \mathcal{A}} (1 - \tilde{y}_{ij})}{\sum_{i \in \mathcal{A}} \tilde{y}_{ij}} \tilde{\sigma}_{ij} \geq 2(1 - e^{-1/2}) \sum_j \tilde{\sigma}_{ij};$$

here, the inequality goes the right way.

In the second allocation scheme, player $i \in \mathcal{A}$ receives

$$\begin{aligned} \mathbf{E}[w_i((S_i^* \setminus V) \cup (S_i'^* \setminus V \setminus U))] &\geq \sum_j \frac{1 - \prod_{i \in \mathcal{A}} (1 - \tilde{y}_{ij})}{\sum_{i \in \mathcal{A}} \tilde{y}_{ij}} \prod_{i \in \mathcal{B}} (1 - \tilde{y}_{ij}) \tilde{\sigma}_{ij} + \gamma_i^* \\ &\geq 2e^{-1/2}(1 - e^{-1/2}) \left(\sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) \tilde{\sigma}_{ij} + \gamma_i \right) \end{aligned}$$

where $\gamma_i^* = g_i(S_i^* \setminus V \setminus U)$ and $g_i(A) = w_i((S_i \setminus V) \cup A) - w_i(S_i \setminus V)$, similarly to the analysis of Algorithm 4. We needed to employ Lemma 3.5 to estimate $\prod_{i \in \mathcal{B}} (1 - \tilde{y}_{ij}) \geq e^{-1/2}(1 - \sum_{i \in \mathcal{B}} \tilde{y}_{ij}^2)$. The remaining estimates are okay, including $\gamma_i^* \geq 2(1 - e^{-1/2})\gamma_i$; since each element of $S_i' \setminus V$ survives in $S_i'^* \setminus V \setminus U$ with probability $\prod_{i \in \mathcal{A}} (1 - \tilde{y}_{ij}) (1 - \prod_{i \in \mathcal{A}} (1 - \tilde{y}_{ij})) / \sum_{i \in \mathcal{A}} \tilde{y}_{ij} \geq 2e^{-1/2}(1 - e^{-1/2})$ for $\sum_{i \in \mathcal{A}} \tilde{y}_{ij} \leq 1/2$, regardless of granularity.

In the third allocation scheme, player $i \in \mathcal{A}$ receives

$$\mathbf{E}[w_i(Y_i'^*)] \geq 2(1 - e^{-1/2}) \left(\sum_j \tilde{\sigma}_{ij} - \gamma_i \right)$$

which is valid without any change, since the inequality goes the right way.

In the fourth allocation scheme, we have

$$\begin{aligned} \mathbf{E}[w_i(Y_i \setminus \bigcup_{k \in \mathcal{B}} Z'_k)] &\geq \mathbf{E}[w_i(S_i \setminus (V \cap (U' \cup V')))] - \gamma_i \\ &\geq \sum_j \left(1 - \left(1 - \prod_{i \in \mathcal{A}} (1 - \tilde{y}_{ij}) \right) \left(1 - \prod_{i \in \mathcal{A} \cup \mathcal{B}} (1 - \tilde{y}_{ij}) \right) \right) \tilde{\sigma}_{ij} - \gamma_i \\ &\geq \sum_j \left(1 - \left(1 - \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) e^{-1/2} \right) \left(1 - \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) e^{-1} \right) \right) \tilde{\sigma}_{ij} - \gamma_i \\ &\geq \sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) \left(e^{-1/2} + e^{-1} - e^{-3/2} \right) \tilde{\sigma}_{ij} - \gamma_i \end{aligned}$$

using Lemma 3.5 and $\max(\sum_{i \in \mathcal{A}} \tilde{y}_{ij}, \sum_{i \in \mathcal{B}} \tilde{y}_{ij}) \leq 1/2$. Finally, we use Y_i^* instead of Y_i which costs us a factor of $2(1 - e^{-1/2})$:

$$\mathbf{E}[w_i(Y_i^* \setminus \bigcup_{k \in \mathcal{B}} Z'_k)] \geq 2(1 - e^{-1/2}) \left(\sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) \left(e^{-1/2} + e^{-1} - e^{-3/2} \right) \tilde{\sigma}_{ij} - \gamma_i \right).$$

Observe that compared to the proof of Theorem 3.3, some terms get the error factor $(1 - \sum_i \tilde{y}_{ij}^2)$. By taking the appropriate linear combination of the four allocation schemes, player i obtains expected profit at least

$$\frac{1 - e^{-1/2}}{1 + 2e^{1/2}} \left(2e^{1/2} + 3 + e^{-1/2} + e^{-1} - e^{-3/2} \right) \sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) \tilde{\sigma}_{ij} \geq 0.645 \sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2 \right) \tilde{\sigma}_{ij}.$$

This is for a fixed balanced solution with marginal values $\tilde{\sigma}_{ij}$. For a random partition $(\mathcal{A}, \mathcal{B})$ and the associated balanced solution, we have $\mathbf{E}[\tilde{\sigma}_{ij}] \geq \sigma_{ij}/(1 + \sqrt{\sum_{i=1}^n y_{ij}^2})$. Also, observe that by the balancing procedure, granularity can only decrease: $\sum_{i=1}^n \tilde{y}_{ij}^2 \leq \sum_{i=1}^n y_{ij}^2$, so player i gets at least

$$0.645 \sum_j \left(1 - \sum_{i=1}^n \tilde{y}_{ij}^2\right) \tilde{\sigma}_{ij} \geq 0.645 \sum_j \frac{1 - \sum_{i=1}^n y_{ij}^2}{1 + \sqrt{\sum_{i=1}^n y_{ij}^2}} \sigma_{ij}.$$

□

This procedure works well for fractional solutions of low granularity. On the other hand, if the granularity is not low, then Algorithm 4 performs better than $1 - 1/e$. A combination of the two is our final algorithm.

Algorithm 6. $((1 - 1/e + 0.00007)$ -approximation for general n players)

1. With probability 0.99, run Algorithm 4.
2. With probability 0.01, run Algorithm 5'.

Theorem 3.7. *For any fractional solution, Algorithm 6 gives each player expected value at least $1 - 1/e + 0.00007 \doteq 0.63219$ of their share in the fractional solution.*

Proof. Consider a fractional solution with σ_{ij} and y_{ij} defined as before. By Lemma 3.1 and 3.5, the first allocation scheme gives each player a set S_i^* of expected value

$$\mathbf{E}[w_i(S_i^*)] \geq \sum_j \left(1 - \prod_{i=1}^n (1 - y_{ij})\right) \sigma_{ij} \geq \sum_j \left(1 - \frac{1}{e} \left(1 - \frac{1}{2} \sum_{i=1}^n y_{ij}^2\right)\right) \sigma_{ij}.$$

By Lemma 3.6, the balancing procedure followed by Algorithm 4 yields an allocation where player i has expected profit at least

$$0.645 \sum_j \left(\frac{1 - \sum_{i=1}^n y_{ij}^2}{1 + \sqrt{\sum_{i=1}^n y_{ij}^2}}\right) \sigma_{ij}.$$

A numerical analysis shows that

$$0.99 \left(1 - \frac{1}{e} \left(1 - \frac{1}{2} \sum_{i=1}^n y_{ij}^2\right)\right) + 0.01 \cdot 0.645 \left(\frac{1 - \sum_{i=1}^n y_{ij}^2}{1 + \sqrt{\sum_{i=1}^n y_{ij}^2}}\right) \geq 1 - \frac{1}{e} + 0.00007$$

for any $\sum_i y_{ij}^2 \in [0, 1]$ and therefore the overall expected profit of player i is at least

$$\sum_j \left(1 - \frac{1}{e} + 0.00007\right) \sigma_{ij} = \left(1 - \frac{1}{e} + 0.00007\right) \sum_S x_{i,S} w_i(S).$$

□

4 Hardness results

In this section we show that there is some constant $\rho < 1$ such that it is NP-hard to approximate the maximum 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 [8] that the the maximum submodular welfare problem is hard to approximate within a ratio better than $1 - 1/e$. In [8] 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 [8]). 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 type 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 maximum 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 maximum 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 maximum submodular welfare problem can be solved in polynomial time.

An interesting feature of our constructions (as well as of the results of [8]) 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 had 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 [5] for max k -cover and the hardness of approximation result of [2] for maximum welfare with XOS utility functions. Section 4.6 presents a new hardness of approximation result for maximum XOS welfare, but this time with constant size utility functions. 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 [2], within ratio of $1/2$ for subadditive utility function, see for example [6], 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 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 [6]), then one can use the following *simple XOS extension*. The value of a set T is $\max_{S}[|T \cap S|]$, 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 [2]) 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 all but t items have 0 value. In this case only t items participate in the extension. The remaining items have 0 value also in the extension.
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 have marginal value, because by then the value of a set (after the extension) is 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 [7].

Lemma 4.2. *There is some $\epsilon > 0$ 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 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 $\epsilon m/3$ (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 $3m - \epsilon m/6$, showing that maximum submodular welfare cannot be approximated within a factor better than $\rho = 1 - \epsilon/18$.

We remark that rather than use the simple submodular extension, one may use a different submodular extension that gives a value somewhat better than $1 - \epsilon/18$ 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 [4]. Recall that a *matching* in a hypergraph is a collection of hyperedges that do not intersect.

Lemma 4.3. *For every $\epsilon > 0$, 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 $(1 - \epsilon)n$ 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 maximum submodular welfare as follows. The vertices of the graph are the items. There are $(1 - \epsilon)n/6$ 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 $(1 - \epsilon)n$, 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 $6\epsilon n$), and $(1/23 - 2\epsilon)n$ 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 cover the maximum number of items. A straightforward greedy algorithm (or alternatively, use of a linear programming relaxation) achieves an approximation ratio of $1 - 1/e \simeq 0.632$ for this problem. In [5] 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 $1 - 1/e + \epsilon$ fraction of the items. Hence it is NP-hard to approximate the max k -cover problem within a ratio better than $1 - 1/e + \epsilon$.

Let us now recall how the hardness of max k -cover is used in [2] 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 $1 - 1/e + \epsilon$ fraction of the items, which for the simple XOS extension means that the utility per item is only $1 - 1/e + \epsilon$ 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 $1 - 1/e + \epsilon$ 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 [5]. Unfortunately, there is no explicit theorem in [5] that lists all these properties. We do not wish to reproduce the rather lengthy proof of [5] in detail here. Hence we shall only list the properties that we use, together with some clearly marked "hints" regarding why the construction of [5] has these properties.

Fix an arbitrarily small value of $\epsilon > 0$. The polynomial time reduction from max-3SAT-5 to max k -cover described in [5] 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 [5] 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 [5] 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 [5], and the partition system described in the proof of Theorem 5.3 in [5].) 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 $(1 - 1/e + \epsilon)$ of the points.

4.6 From max k -cover to maximum welfare

We now 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 $G_p = \cup_{S \in g_p} S$ 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 [2], this reduction gives a hardness of approximation within a ratio of $1 - 1/e + \epsilon$, but the advantage of our reduction is that each of the utility functions is of constant size.

For the case of submodular utility functions, we can instead take 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 $(1 - 1/e + \epsilon)$. (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 $s - (1/e - \epsilon)/2$ per player, and hence it

is NP-hard to approximate maximum submodular welfare within ratio better than $1 - (1/e - \epsilon)/2s$. Observe that this ratio is bounded away from 1 (but not by much) because s is some constant.

We remark that using additional properties of the reduction of [5] (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 $1/2 + \epsilon$ 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.

4.7 Two players

In this section we prove 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 [5]. Namely, rather than use a k prover proof system (which was perhaps the central aspect in which [5] improves over the previous reduction of [10]), we shall limit the number of provers to two (as in the proof in [10]). This has the effect that the hardness of approximation ratio for max k -cover becomes $3/4 + \epsilon$ rather than $1 - 1/e + \epsilon$, 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 maximum 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 $1/4$.

References

- [1] Steven J. Brams, Alan D. Taylor. Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press, 1996.
- [2] Shahar Dobzinski, Noam Nisan, Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. STOC 2005: 610–618.
- [3] Shahar Dobzinski, Michael Schapira. An Improved approximation algorithm for combinatorial auctions with submodular bidders. To appear in the proceedings of SODA 2006.
- [4] Elad Hazan, Shmuel Safra, Oded Schwartz. On the Complexity of Approximating k -Dimensional Matching. RANDOM-APPROX 2003: 83–97.
- [5] Uriel Feige. A Threshold of $\ln n$ for Approximating Set Cover. J. ACM 45(4): 634–652 (1998).
- [6] Uriel Feige. On maximizing welfare when utility functions are subadditive. To appear in STOC 2006.

- [7] Uriel Feige, Magnus M. Halldorsson, Guy Kortsarz, Aravind Srinivasan. Approximating the Domatic Number. *SIAM J. Comput.* 32(1): 172–195 (2002).
- [8] Subhash Khot, Richard Lipton, Evangelos Markakis, Aranyak Mehta. Inapproximability Results for Combinatorial Auctions with Submodular Utility Functions. To appear in WINE 2005.
- [9] Benny Lehmann, Daniel J. Lehmann, Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *ACM Conference on Electronic Commerce 2001*: 18–28.
- [10] Carsten Lund, Mihalis Yannakakis. On the Hardness of Approximating Minimization Problems. *J. ACM* 41(5): 960–981 (1994).
- [11] Noam Nisan, Ilya Segal. The communication requirements of efficient allocations and supporting prices. To appear in *Journal of Economic Theory*.