Approximation algorithms for allocation problems: Improving the factor of $1 - 1/e$

Uriel Feige
Microsoft Research
Redmond, WA 98052
urifeige@microsoft.com

Jan Vondrák
Princeton University *
Princeton, NJ 08540
jvondrak@gmail.com

Abstract

Combinatorial allocation problems require allocating items to players in a way that maximizes the total utility. Two such problems received attention recently, and were addressed using the same linear programming (LP) relaxation. In the Maximum Submodular Welfare (SMW) problem, utility functions of players are submodular, and for this case Dobzinski and Schapira [SODA 2006] showed an approximation ratio of $1 - 1/e$*. In the Generalized Assignment Problem (GAP) utility functions are linear but players also have capacity constraints. GAP admits a* $(1 - 1/e)$*-approximation as well, as shown by Fleischer, Goemans, Mirrokni and Sviridenko [SODA 2006]. In both cases, the approximation ratio was in fact shown for a more general version of the problem, for which improving* $1 - 1/e$ *is NP-hard.*

In this paper, we show how to improve the $1 - 1/e$ *approximation ratio, both for SMW and for GAP. A common theme in both improvements is the use of a new and optimal Fair Contention Resolution technique. However, each of the improvements involves a different rounding procedure for the above mentioned LP.*

In addition, we prove APX-hardness results for SMW (such results were known for GAP). An important feature of our hardness results is that they apply even in very restricted settings, e.g. when every player has nonzero utility only for a constant number of items.

1 Introduction

Allocation problems. Combinatorial allocation problems arise in situations such as combinatorial auctions, where items or goods are to be allocated to competing players (or "bidders") by a central authority, with the goal of maximizing the total utility provided to the players. In the most general setting, there is a finite set I of m items and n players who have possibly different utility functions $w_i : 2^I \to \mathbb{R}$, representing their utility derived from subsets of I. We always assume **monotonicity**: $\forall S \subset T;\ 0 = w_i(\emptyset) \leq w_i(S) \leq w_i(T)$. The goal is to find disjoint sets $S_1, S_2, \ldots, S_n \subseteq I$ to be allocated to the n players such that their total utility $\sum_{i=1}^{n} w_i(S_i)$ is maximized.

In this generality, the problem is NP-hard to approximate within any reasonable factor, such as $m^{1/2-\epsilon}$ even for "single-minded bidders" [10, 11]. Positive results can be achieved only under stronger assumptions on the utility functions. The following properties are commonly considered, in the decreasing order of generality:

1. **Subadditivity.** $w_i(S \cup T) \leq w_i(S) + w_i(T)$.
2. **Fractional subadditivity.** $w_i(S) \leq \sum_k \alpha_k w_i(T_k)$, whenever $0 \leq \alpha_k \leq 1$ and $\sum_{k: j \in T_k} \alpha_k \geq 1$ for each $j \in S$. As proved in [6], this is equivalent to the "XOS property": $w_i(S) = \max_{t \in \mathcal{T}} f_{it}(S)$, where each f_{it} is linear.
3. **Submodularity.** $w_i(S \cup T) + w_i(S \cap T) \leq w_i(S) + w_i(T)$.
4. **Linearity.** $w_i(S) = \sum_{j \in S} w_{ij}$.

Another issue is how the utility functions are represented and accessible to an algorithm. An explicit table of values would require exponential size, while we would like to achieve running time polynomial in n and m. Compact representations are possible in the case of linearity and in other special cases. In general, we assume an *oracle model* to access the utility function. In the weakest model, only a *value oracle* is available, which returns the value of a given set for a given player. Following [3, 4, 6], we use the notion of a *demand oracle*: Given prices p_j to individual items, the oracle returns a set maximizing $w_i(S) - \sum_{j \in S} p_j$.

In this paper, we consider three special types of allocation problems:

*work done while at Microsoft Research

- **Maximum Submodular Welfare (*SMW*).** The allocation problem where utility functions are monotone and submodular. We assume the availability of a demand oracle to query the utility functions.

- **Generalized Assignment Problem (*GAP*).** The allocation problem with linear utility functions, given explicitly by item values w_{ij}. In addition, each player has a *capacity constraint* defining feasible sets: $\sum_{j \in S} s_{ij} \leq 1$. Here, s_{ij} is the size of item j, possibly different for each player i.

- **Maximum Fractionally Subadditive Welfare (*FSAW*).** The allocation problem with fractionally subadditive functions, accessible using a demand oracle.

Note. In the context of *GAP*, "players" are usually referred to as "bins". Throughout this paper, "players" and "bins" refer to the same entities. We use "bins" only when talking specifically about *GAP*.

Capacity constraints can be simulated using a fractionally subadditive function: For each set S_t feasible for player i, define $f_{it}(S) = \sum_{j \in S \cap S_t} w_{ij}$ and $w_i(S) = \max_t f_{it}(S)$. Under this utility function, it does not bring any advantage to allocate an infeasible set S to player i (because we might as well replace it by S_t for some feasible $S_t \subseteq S$). Therefore, the *FSAW* problem with utility functions $w_i(S) = \max_t f_{it}(S)$ is equivalent to *GAP*. It is also known that submodular functions are fractionally subadditive. I.e., both *SMW* and *GAP* are special cases of *FSAW*.

Previous work. Allocation problems have received a lot of attention recently. The Maximum Submodular Welfare problem was considered first by Lehmann, Lehmann and Nisan [9] who presented a $1/2$-approximation algorithm using a *value oracle* only. In this model, Khot et al. proved NP-hardness of approximation for any factor better than $1-1/e$ [8]. The factor of $1-1/e$ has been recently achieved by Dobzinski and Schapira [4], but in the stronger *demand oracle* model. For the more general class of subadditive utility functions, Feige [6] developed a $1/2$-approximation algorithm, which is optimal unless $P = NP$. No APX-hardness result was known for submodular utilities in the demand oracle model.

Regarding the Generalized Assignment Problem, a $1/2$-approximation algorithm was implicit in the work of Shmoys and Tardos [12], as observed by Chekuri and Khanna [2]. The approximation factor has been recently improved to $1 - 1/e$ by Fleischer, Goemans, Mirrokni and Sviridenko [5]. Although the title of [5] might suggest otherwise, it remained an open question whether $1 - 1/e$ is indeed the optimal approximation factor for *GAP*. The authors in [5] proved hardness of $(1 - 1/e + \epsilon)$-approximation for *SAP*, an assignment problem with more general feasibility constraints (in particular, for its special case, the Distributed Caching Problem). For *GAP*, only hardness of $(1 - \epsilon)$-approximation for some small $\epsilon > 0$ was known.

Both $(1 - 1/e)$-approximation algorithms (for *GAP* and *SMW*) use a similar LP-based approach. The factor of $1 - 1/e$ arises because of the same reason: both algorithms use randomized rounding with independent choices for different players. The $(1 - 1/e)$-approximation algorithm in [4] also applies to the more general *FSAW* problem, but only in an even stronger model using an "XOS oracle". In [6], it is shown that the XOS oracle is actually not needed and *FSAW* has a $(1 - 1/e)$-approximation using only demand queries. This is optimal unless $P = NP$ [3]. Still, it was not known whether $1 - 1/e$ is optimal for *GAP* and *SMW*.

Our results. Our main result is that the factor of $1 - 1/e$ is *suboptimal* for both *GAP* and *SMW*. In both cases, we develop randomized approximation algorithms which achieve an approximation factor of $1 - 1/e + \epsilon$ for some absolute constant $\epsilon > 0$. Both algorithms are based on the same LP as used in [4, 6]. The rounding techniques are, however, quite different in the two cases. The intuitive reason why an improvement is possible is that the $(1-1/e)$-approximation algorithm, using independent choices for different players, leaves an expected $1/e$-fraction of all items unclaimed. These items can be possibly allocated to some players, increasing their utilities. However, two issues arise: in the case of submodular utility functions, it is not clear whether additional items always bring additional profit (and indeed, sometimes they do not). In the case of *GAP*, it is not clear whether additional items can be added to any player without violating the capacity constraint. These two obstacles need to be treated differently; we explain the case of *SMW* in Section 3 and the case of *GAP* in Section 4. We prove that in both cases, the LP integrality gap is bounded away from $1 - 1/e$. On the other hand, we present simple examples of integrality gap $5/6$ for *SMW* and $4/5$ for *GAP* (see Section 2.1).

Fair Contention Resolution. The first step in both of our algorithms is a new technique to resolve conflicts between the random choices of different players. We believe that this technique may be of independent interest: Suppose that n players request an item independently with probabilities $p_1, p_2, \ldots, p_n$. This might result in several players requesting the item simultaneously. We show how to resolve contention among the competing players, so that conditioned on any fixed player competing, she obtains the item with the same probability $(1 - \prod_i (1 - p_i)) / \sum_i p_i$. This is optimal since $1 - \prod_i (1 - p_i)$ is the probability that at least one player competes for the item. This technique implies an approximation factor of $1 - (1 - 1/n)^n$ for the *FSAW* problem with n players. For details, see Section 2.2.

Hardness of approximation. We prove that there is $\epsilon > 0$ such that it is NP-hard to approximate the *SMW* problem within a factor of $1-\epsilon$. Our reduction has the property that each player is only interested in a constant number of items. We also show that $(1-\epsilon)$-approximation is NP-hard in the case of n players with the same utility function of polynomial size, and in the case of two players whose utility functions are "separable", meaning that $w(S) = \sum_i w(S \cap C_i)$ where C_i are disjoint classes of constant size. In all these cases, demand queries can be answered efficiently, in contrast to previously known reductions. More details can be found in Section 5.

2 The Configuration LP

The following linear program has been used to develop approximation algorithms for several allocation problems [3, 4, 5, 6].

$$\begin{array}{rl} LP = \max & \sum_i \sum_{S \in \mathcal{I}_i} x_{i,S}\, w_i(S); \\ \forall j; & \sum_i \sum_{S \in \mathcal{I}_i : j \in S} x_{i,S} \leq 1, \\ \forall i; & \sum_{S \in \mathcal{I}_i} x_{i,S} \leq 1, \\ & x_{i,S} \geq 0. \end{array}$$

Here, $\mathcal{I}_i$ denotes the collection of feasible sets for player i. In the case of *SMW*, this is the collection of all sets of items, while in the case of *GAP*, $\mathcal{I}_i$ contains the sets respecting the capacity constraint for player i. The variable $x_{i,S}$ can be interpreted as the extent to which set S is allocated to player i. For an integer solution $x_{i,S} \in \{0, 1\}$, the constraints require each player to choose at most one set, and each item to be allocated to at most one player.

This LP has exponentially many variables but there is an optimal solution with polynomially many nonzero variables. We do not address the issue of solving the LP here. For allocation problems with a demand oracle, the LP can be solved since the demand oracle provides a separation oracle for the dual [3]. For the *GAP* problem, the LP can be solved to an arbitrary precision (see [5] for more details). In this work, we suppose a fractional solution is given to us and we develop randomized rounding techniques to convert this into an integral solution.

2.1 Integrality gaps

We present two simple examples for this LP, showing integrality gap $5/6$ for *SMW* and $4/5$ for *GAP*. The examples use only 2 players and 4 or 3 items, respectively. We can also prove that our gaps are the worst possible for 2 players with a half-integral fractional solution; we do not present the proof here.

Example for SMW, integrality gap $5/6$. Consider 4 items $\{a, b, c, d\}$ and 2 players. Each singleton has value 3 and every set of at least 3 elements has value 6. For pairs of items, define the utility function of player 1 as follows:

$w_1(a,d) = w_1(b,c) = 5$	$w_1(a,c) = 4$	$w_1(b,d) = 4$
$w_1(a,b) = 6$	a	b
$w_1(c,d) = 6$	c	d

Symmetrically, the utility function of player 2 is the same except that $w_2(a,b) = w_2(c,d) = 4$ and $w_2(a,c) = w_2(b,d) = 6$. This can be verified to be a submodular function. The optimal fractional solution is $x_{1,\{a,b\}} = x_{1,\{c,d\}} = x_{2,\{a,c\}} = x_{2,\{b,d\}} = 1/2$ which has value 12, while any integral solution has value at most 10.

Example for GAP, integrality gap $4/5$. Consider 3 items $\{a, b, c\}$ and 2 bins. The following table shows the values and sizes for the two bins:

Item	$size_1$	$value_1$	$size_2$	$value_2$
a	0.5	1	1.0	2
b	0.5	2	0.5	2
c	1.0	2	0.5	1

The optimal fractional solution is $x_{1,\{a,b\}} = x_{1,\{c\}} = 1/2$ and $x_{2,\{a\}} = x_{2,\{b,c\}} = 1/2$ which has value 5. Any integral solution has value at most 4.

2.2 The Fair Rounding Algorithm

We start by giving an alternative algorithm to achieve the factor of $1 - 1/e$ for the *FSAW* problem. This rounding technique combines all the favorable properties of previous approaches: (1) It is efficient and oblivious, i.e. it does not depend on the utility functions. (2) It achieves an approximation factor of $1 - (1 - 1/n)^n$ which is strictly better than $1 - 1/e$ for any fixed number of players. (3) It beats $1 - 1/e$ even more significantly, if the fractional solution is "unbalanced"; this will be useful later. (4) It treats all players equally, with the same approximation guarantee per player.

The first step in a randomized rounding algorithm is to interpret the variables $x_{i,S}$ as probabilities. Since $\sum_S x_{i,S} \leq 1$, this is a valid probability distribution for each player i. We say that a player samples a random set from her distribution, if S is chosen with probability $x_{i,S}$. Observe that $\mathbb{E}[w_i(S)] = \sum_S x_{i,S} w_i(S)$ is exactly the player's share in the LP.

The Fair Rounding Algorithm.

1. **Tentative Choices.** Let each player sample independently a random "tentative set" S_i from her distribution. Each player "competes" for the items in her tentative set.

2. **Fair Contention Resolution.** Consider an item j. Denote by A_j the random set of players competing for item j. Let $p_{ij} = \Pr[i \in A_j] = \sum_{S \in \mathcal{I}_i : j \in S} x_{i,S}$ and $s_j = \sum_{i \in A_j} p_{ij} / \sum_{i=1}^{n} p_{ij}$.

   - If $A_j = \emptyset$, do not allocate the item.
   - If $A_j = \{k\}$, allocate the item to player k.
   - If $|A_j| > 1$, choose randomly one of two contention resolution schemes: (a) with probability $1 - s_j$, (b) with probability s_j.

     (a) Allocate item j to a uniformly random player $k \in A_j$.

     (b) Allocate item j to a random player $k \in A_j$, with probabilities proportional to $\sum_{i \in A_j \setminus \{k\}} p_{ij}$.

Remark. The uniform contention resolution scheme (a) has been previously considered [6] and it is known that it achieves factor $1/2$ for fractionally subadditive functions. In order to improve this, it is necessary to "penalize" players who request the item with high probability, which is the purpose of scheme (b).

Lemma 1. *Let players compete for an item independently with probabilities $p_1, p_2, \ldots, p_n$. Conditioned on player k competing, the Fair Contention Resolution technique allocates it to her with probability*

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

Proof. Condition on A being the set of players competing for the item and assume $|A| > 1$. In scheme (a), each player in A obtains the item with probability $1/|A|$. In (b), we get by normalizing that the probability for player k is $1/(|A| - 1) \cdot \sum_{i \in A \setminus \{k\}} p_i / \sum_{i \in A} p_i$. By averaging these two schemes with weights $1 - s$ and s, where $s = \sum_{i \in A} p_i / \sum_{i=1}^{n} p_i$, we obtain that player k gets the item with probability

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

conditioned on the set of competing players A.

Suppose for now that we allocate the item to each player $k \in A$ with probability $r_{A,k}$, even when $A = \{k\}$. (For the sake of the proof, we interpret the sum over $A \setminus \{k\}$ for $A = \{k\}$ as zero, even though the summand is undefined.) Then the total probability that player k receives the item would be

$$q'_k = p_k \, \mathbb{E}_A[r_{A,k} \mid k \in A]. \qquad (2)$$

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 $\frac{p_k}{\sum_{i=1}^{n} p_i} \Pr[A = \{k\}]$ which makes the total probability that player k obtains the item equal to

$$q_k = q'_k + \frac{p_k}{\sum_{i=1}^{n} p_i} \Pr[A = \{k\}]. \qquad (3)$$

We would like to show that $q_k = \frac{p_k}{\sum p_i} \Pr[A \neq \emptyset]$. This means that q'_k should be equal to $\frac{p_k}{\sum p_i} \Pr[A \setminus \{k\} \neq \emptyset]$. 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 expand (2) as a weighted sum over all possible subsets $A' \subseteq B$:

$$\begin{aligned} q'_k &= p_k \, \sum_{A' \subseteq B} p(A') \, r_{A' \cup \{k\},k} \\ &= \tfrac{p_k}{\sum_{i=1}^{n} p_i} \sum_{A' \subseteq B} p(A') \left(\sum_{i \in A'} \tfrac{p_i}{|A'|} + \sum_{i \in B \setminus A'} \tfrac{p_i}{|A'|+1} \right). \end{aligned}$$

Ideally, we would like to get $q'_k = \frac{p_k}{\sum p_i} \sum_{\emptyset \neq A' \subseteq B} p(A')$ but we have to perform a certain redistribution of terms to achieve this. Observe that

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

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

$$\begin{aligned} q'_k &= \frac{p_k}{\sum_{i=1}^{n} p_i} \Big(\sum_{A' \subseteq B} \sum_{i \in A'} p(A') \frac{p_i}{|A'|} \\ &\quad + \sum_{A' \subseteq B} \sum_{i \in B \setminus A'} p(A' \cup \{i\}) \frac{1 - p_i}{|A' \cup \{i\}|} \Big) \\ &= \frac{p_k}{\sum_{i=1}^{n} p_i} \Big(\sum_{A' \subseteq B} \sum_{i \in A'} p(A') \frac{p_i}{|A'|} \\ &\quad + \sum_{\emptyset \neq A'' \subseteq B} \sum_{i \in A''} p(A'') \frac{1 - p_i}{|A''|} \Big) \end{aligned}$$

and joining the terms where $A' = A'' \neq \emptyset$, we get

$$\begin{aligned} q'_k &= \tfrac{p_k}{\sum_{i=1}^{n} p_i} \sum_{\emptyset \neq A' \subseteq B} p(A') \left(\sum_{i \in A'} \tfrac{p_i}{|A'|} + \sum_{i \in A'} \tfrac{1 - p_i}{|A'|} \right) \\ &= \tfrac{p_k}{\sum_{i=1}^{n} p_i} \sum_{\emptyset \neq A' \subseteq B} p(A') = \tfrac{p_k}{\sum_{i=1}^{n} p_i} \Pr[A \setminus \{k\} \neq \emptyset]. \end{aligned}$$

So indeed, we obtain from (3) that player k receives the item with probability

$$q_k = \tfrac{p_k}{\sum_{i=1}^{n} p_i} \Pr[A \neq \emptyset] = \tfrac{p_k}{\sum_{i=1}^{n} p_i} \left(1 - \prod_{i=1}^{n} (1 - p_i)\right).$$

∎

In the case of two players who compete for an item with probabilities $p_1 + p_2 = 1$, this algorithm resolves a possible contention by allocating the item with *reversed probabilities*: p_2 for player 1 and p_1 for player 2. This achieves an approximation factor of $3/4$ for two players, even for fractionally subadditive utilities, as observed already in [6]. For n players, this technique guarantees an approximation factor of $1-(1-1/n)^n > 1-1/e$. This follows just like in [6] from the fact that each player receives every requested item with conditional probability $\rho \geq 1-(1-1/n)^n$. This matches exactly the known integrality gap for the *FSAW* problem - for fractionally subadditive utilities, this technique cannot be improved. However, our main goal is to obtain an absolute constant larger than $1-1/e$, independent of n, for the *SMW* and *GAP* problems.

2.3 Unbalanced fractional solutions

Observe that the Fair Contention Resolution technique actually gives a value of ρ better than $1-(1-1/n)^n$, if some players take the item with a large probability $p_{ij} = \sum_{S \in \mathcal{I}_i : j \in S} x_{i,S}$. Let's fix an $\epsilon_1 > 0$ and call an item j ϵ_1-*unbalanced* if $p_{ij} > \epsilon_1$ for some i, or if $\sum_i p_{ij} < 1-\epsilon_1$. An elementary estimate gives that in both cases, each player receives item j with conditional probability

$$\rho_j = \frac{1-\prod_{i=1}^{n}(1-p_{ij})}{\sum_{i=1}^{n} p_{ij}} \geq 1-\left(1-\frac{1}{2}\epsilon_1^2\right)\frac{1}{e}.$$

If a substantial fraction of the LP value comes from unbalanced items, then we gain compared to $1-1/e$. However, we have to define carefully what we mean by the "contribution of an item" in the case of submodular utility functions.

Definition 2. *Fix an ordering of the items* $[m] = \{1,2,\ldots,m\}$ *and denote* $[j] = \{1,2,\ldots,j\}$*. Define the expected contribution of item* j *to player* i *as*

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

where S_i *is a random set sampled from player* i*'s distribution.*

Note that $\sum_j \sigma_{ij} = \mathbb{E}[w_i(S_i)]$ and $\sum_{i,j} \sigma_{ij} = LP$. This way of partitioning the LP value is useful because of the following lemma.

Lemma 3. *Let* S_i *be a random set sampled from player* i*'s distribution, and let* X *be a random set such that conditioned on any* S_i *and for any item* j*,* $\Pr[j \in X \mid S_i] \geq \rho_j$*. Assume* w_i *monotone and submodular. Then taking the expectation over both* S_i *and* X*,*

$$\mathbb{E}[w_i(S_i \cap X)] \geq \sum_{j=1}^{m} \rho_j \sigma_{ij}.$$

Proof. Using the property of decreasing marginal values, we obtain

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

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

$$\mathbb{E}_X[w_i(S_i \cap X) \mid S_i] \geq \sum_{j=1}^{m} \rho_j (w_i(S_i \cap [j]) - w_i(S_i \cap [j-1]))$$

and therefore $\mathbb{E}_{S_i,X}[w_i(S_i \cap X)] \geq \sum_{j=1}^{m} \rho_j \sigma_{ij}$. □

Note. This applies to *GAP* even more easily, since items have individual values w_{ij} and we have $\sigma_{ij} = p_{ij} w_{ij}$.

Now we can argue that we gain if ϵ_1-unbalanced items contribute a significant fraction of the LP value. As a consequence, we can assume that there are no ϵ_1-unbalanced items; we call such a fractional solution ϵ_1-balanced.

Lemma 4. *Suppose that for any* ϵ_1*-balanced fractional solution, there is a rounding technique (for either SMW or GAP) which achieves an approximation factor of* $1-1/e+\epsilon$*,* $0 < \epsilon < 1/(2e)$*. Then there is a rounding technique for any fractional solution which achieves factor at least* $1-1/e+\epsilon\epsilon_1^2/(2e)$*.*

Proof. Let U denote the set of ϵ_1-unbalanced items. First suppose that $\sum_i \sum_{j \in U} \sigma_{ij} \geq \epsilon \cdot LP$ and apply the Fair Rounding Algorithm. For each unbalanced item $j \in U$, each player competing for it receives it with conditional probability at least $1-1/e+\epsilon_1^2/(2e)$. By Lemma 3, the expected total profit from this rounding is at least

$$\begin{aligned} \sum_i \sum_j \rho_j \sigma_{ij} &\geq \left(1-\frac{1}{e}\right)\sum_i \sum_j \sigma_{ij} + \frac{\epsilon_1^2}{2e}\sum_i \sum_{j \in U} \sigma_{ij} \\ &\geq \left(1-\frac{1}{e}+\frac{\epsilon\epsilon_1^2}{2e}\right) \cdot LP. \end{aligned}$$

Otherwise, we have $\sum_i \sum_{j \in U} \sigma_{ij} < \epsilon \cdot LP$. Then let's remove the unbalanced items. In the worst case this incurs a factor of $(1-\epsilon)$ on the value of the fractional solution. By assumption, there is a technique (to be shown later) which we can apply to the balanced solution, that achieves approximation factor at least $(1-1/e+\epsilon)(1-\epsilon) = 1-1/e+\epsilon/e-\epsilon^2$ which is at least $1-1/e+\epsilon/(2e)$ for $\epsilon < 1/(2e)$. □

The arguments so far were equally valid for *SMW* and *GAP*. We can assume now that the fractional solution is ϵ_1-balanced. Applying the Fair Rounding Algorithm (or in fact any rounding procedure where every player samples independently a random set and contention is resolved in some way), we see that item j remains unclaimed with probability $\prod_{i=1}^{n}(1-p_{ij}) \simeq 1/e$. On the average, $1/e$ of all items are available to be allocated in a second stage, which gives some hope that an improvement over $1-1/e$ should be possible. Somewhat surprisingly, there exist instances where the remaining items do not bring any additional profit (for *SMW*) or they do not fit (for *GAP*), regardless of how we assign them or how conflicts were resolved in the first stage. Therefore we must redesign even the first stage of the algorithm in order to achieve some improvement. This is the point where the two solutions diverge.

3 Maximum Submodular Welfare

Our general idea is that each player should sample multiple sets independently from their distribution. However, contention must be resolved carefully among these sets, so that we can provably increase the total welfare. Our final rounding technique is quite complicated; it is instructive to describe it first on the example of two players.

3.1 Two players with a balanced fractional solution

Assume that for any item j, $\sum_{S:j\in S} x_{1,S} = \sum_{T:j\in T} x_{2,T} = 1/2$. This is the worst case for the Fair Rounding Algorithm, in which case the approximation factor is exactly $3/4$. We show that in fact we can improve upon $3/4$ in this special case.

Algorithm for two players with submodular utilities and a balanced fractional solution.

- Player 1 samples independently random sets S, S'.
- Player 2 samples independently random sets T, T'.
- Let $Y = (S\cap T)\cup(S'\cap\overline{T})$, $Z' = (T\cap S')\cup(T'\cap\overline{S'})$.

We assign items randomly as follows:

Probability	1/3	1/3	1/6	1/6
Player 1	S	$\overline{T}$	Y	$Y\setminus Z'$
Player 2	$\overline{S}$	T	$Z'\setminus Y$	Z'

Theorem 5. *The algorithm for two balanced players gives expected profit at least* $37/48\cdot LP$.

Proof. We only sketch the important arguments. Note that each of the random sets S, T, S', T', Y, Z' contains every item with probability $1/2$. For player 1, let $\alpha = \mathbb{E}[w_1(S)]$ be his share in the LP, while $\mathbb{E}[w_1(\overline{T})]$ is what's left after player 2 chooses her set T first. Since every item remains in $\overline{T}$ with prob. $1/2$, by monotonicity and Lemma 3 with $X=\overline{T}$ and $\rho_j = 1/2$ we get $\mathbb{E}[w_1(\overline{T})] \geq \mathbb{E}[w_1(S\cap\overline{T})] \geq \alpha/2$. If the inequalities were tight, player 1 would get only $1/2$ of his share, following player 2's choice. Randomizing the ordering of the two players, this would yield a factor of $3/4$. However, at this moment the sets Y and Z' come into play. For Y, the condition of submodularity implies that

$$\begin{aligned}\mathbb{E}[w_1(Y)] + \mathbb{E}[w_1(\overline{T})] &\geq \mathbb{E}[w_1(Y\cup\overline{T})] + \mathbb{E}[w_1(Y\cap\overline{T})]\\ &\geq \mathbb{E}[w_1(S)] + \mathbb{E}[w_1(S'\cap\overline{T})] \geq \tfrac{3}{2}\alpha\end{aligned}$$

using linearity of expectation and Lemma 3. I.e., if $\mathbb{E}[w_1(\overline{T})]$ is close to $\alpha/2$, then $\mathbb{E}[w_1(Y)] \simeq \alpha$. The same holds for Z' by the same reasoning for player 2. Thus in this case, the sets Y, Z' retain the complete value of the LP. Moreover, they are not independent like S and T. The events $j\in Y, j\in Z'$ are *negatively correlated*: While $\Pr[j\in Y] = \Pr[j\in Z'] = 1/2$,

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

rather than $1/4$ which is the probability of appearance in $S\cap T$. Thus we lose only $3/16$ of the expected value by resolving contention between Y and Z'. A detailed computation yields that the total expected profit from this rounding technique is at least $37/48\cdot LP$. ☐

Similarly to Lemma 4, this implies an improvement over $3/4$ for two players in the general case. Let's remark that by a more careful analysis, we are able to combine the balanced and unbalanced case to achieve an approximation factor of $13/17$ for two players.

3.2 n players with a balanced fractional solution

We adapt the ideas from the two-player case to achieve a constant improvement over $1-1/e$ for any number of players. Our approach is to divide the players randomly into two groups $\mathcal{A}, \mathcal{B}$ and treat them as two "superplayers". Let's assume again that the fractional solution is ϵ_1-balanced. Due to a concentration result on sums of independent variables, we have for most items j

$$\sum_{i\in\mathcal{A}} p_{ij} \simeq \sum_{i\in\mathcal{B}} p_{ij} \simeq \frac{1}{2} \pm O(\sqrt{\epsilon_1}).$$

For a collection of sets $\{S_i : i\in\mathcal{A}\}$ sampled by players in one group, we use Lemma 1 to resolve contention. To distribute items between the two groups, we use ideas inspired by the two-player case.

Algorithm for n balanced submodular 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 contention among S_i to obtain disjoint sets S_i^*. Similarly, resolve contention among S'_i to obtain disjoint sets $S_i'^*$. Let the players in $\mathcal{B}$ resolve contention among T_i to obtain disjoint sets T_i^* and among T'_i to obtain disjoint sets $T_i'^*$.
- Let $Y_i^* = (S_i^* \cap V) \cup (S_i'^* \cap \overline{V})$, $Z_i^* = (T_i^* \cap U) \cup (T_i'^* \cap \overline{U})$, $Y_i'^* = (S_i^* \cap V') \cup (S_i'^* \cap \overline{V'})$ and $Z_i'^* = (T_i^* \cap U') \cup (T_i'^* \cap \overline{U'})$.

We assign the items randomly according the following table, with $p = 1/(1+2e^{1/2})$:

Prob.	Player $i \in \mathcal{A}$	Player $i \in \mathcal{B}$
$e^{1/2}p$	S_i^*	$(T_i^* \cup (T_i'^* \setminus V)) \setminus U$
$e^{1/2}p$	$(S_i^* \cup (S_i'^* \setminus U)) \setminus V$	T_i^*
$p/2$	$Y_i'^*$	$Z_i^* \setminus \bigcup_{j\in\mathcal{A}} Y_j'^*$
$p/2$	$Y_i^* \setminus \bigcup_{j\in\mathcal{B}} Z_j'^*$	$Z_i'^*$

Theorem 6. *For n players with an ϵ_1-balanced fractional solution, the algorithm above yields expected profit at least* $(1 - 1/e + 1/100 - O(\sqrt{\epsilon_1})) \cdot LP$.

We omit the proof from this extended abstract. Due to Lemma 4, this implies a $(1 - 1/e + \epsilon)$-approximation for any fractional solution.

4 Generalized Assignment Problem

Now we turn to the Generalized Assignment Problem (*GAP*). Recall that in *GAP*, each item j has an explicit value w_{ij} for bin i. The added complication is that only sets satisfying the capacity constraint $\sum_{j\in S} s_{ij} \le 1$ can be allocated to bin i. As a result, our approach for the *SMW* problem does not work here, since we cannot pack two sets S_i, S'_i into the same bin. Instead, we proceed as follows. Recall that we have to deal only with the case of ϵ_1-balanced fractional solutions, for some very small constant $\epsilon_1 > 0$. Also, we will show that we may further assume that item values are *uniform*, and there are no *precious sets* (these terms will be defined shortly). Thereafter, we will be able to prove that it is possible to improve the Fair Rounding Algorithm by reallocating certain items and packing some additional items.

Non-uniform item values. First, we treat the case where item values vary significantly between different bins. We know that the Fair Rounding Algorithm yields a factor of at least $1 - 1/e$. However, since now the utility functions are linear, there is a simpler way to resolve conflicts - just allocate the item to the bin that gives it the maximum value w_{ij}. This is the technique used in [5]; let's call it the *Greedy Rounding Algorithm.* We show that if the item values are significantly non-uniform then Greedy Rounding gains significantly compared to $1 - 1/e$. The following lemma follows easily by comparing the two rounding methods.

Lemma 7. *Fix $\epsilon_2 > 0$. For each item j, let $W_j = \sum_i p_{ij} w_{ij}$ and $B_j = \{i : w_{ij} < (1-\epsilon_2)W_j\}$. Call the value of item j non-uniform if $\sum_{i\in B_j} p_{ij} > \epsilon_2$. For any non-uniform item, the Greedy Rounding Algorithm retrieves expected value at least* $(1 - 1/e + (\epsilon_2/e)^2)W_j$.

We interpret the set B_j as "bad bins" for item j. The meaning of this lemma is that if many bins are bad for item j, then we gain by placing it in the optimal bin rather than a bad bin. We choose $\epsilon_2 > 0$ such that $\epsilon_1 = e^{-2}\epsilon_2^3$. If at least an ϵ_2-fraction of the LP value comes from non-uniform items, we gain $e^{-2}\epsilon_2^3 LP = \epsilon_1 LP$. If the contribution of non-uniform items is smaller than ϵ_2, we remove each item from all its bad bins, and we remove all non-uniform value items completely. We redefine the value of item j to $(1 - \epsilon_2)W_j$. This decreases the value of the LP by at most $3\epsilon_2 \cdot LP$ and after this procedure, each item has the same value for each bin where it is used (and the new value is not higher than the original value).

Precious sets. Now we can assume that the value of each item is independent of the bin where it is placed, $w_j = w_{ij}\ \forall i$. Next, we consider the distribution of values among different sets for a given bin. The average value assigned to bin i is $V_i = \sum_S x_{i,S} w(S) = \sum_j p_{ij} w_j$. We call a set S *precious* for bin i if $w(S) > 10V_i$. We prove that if many sets are precious, we can gain by taking these sets with higher probability. We set $\epsilon_3 = 400\epsilon_2$. The following lemma can be proved by increasing the sampling probability of each precious set by a factor of 2 and decreasing the sampling probabilities of non-precious sets.

Lemma 8. *Assume that precious sets contribute value at least $\epsilon_3 \cdot LP$. Then there is an algorithm which achieves expected value at least* $(1 - 1/e + \epsilon_3/100) \cdot LP$.

Again, we either gain due to this lemma, or we remove all precious sets from the LP. If most of a bin's value is composed of precious sets, we remove the bin from the solution completely. For other bins, the expected value V_i might go down by a factor of 2, but overall we lose at most $2\epsilon_3 \cdot LP$ and any set possibly allocated to bin i has value at most $20V_i$.

Migrants. Now the item values are uniform and there are no precious sets. Our final goal is to pack some additional items into the gaps created by contention among bins. We process the bins in the order of $V_1 \leq V_2 \leq V_3 \leq \ldots$ and resolve contention in favor of the first bin that claims an item in this order. We call this the *Sequential Allocation Algorithm*. This would still achieve an approximation factor of $1 - 1/e$. However, we prove that this algorithm can be improved, since some items are gone due to contention with preceding bins and some additional items can be packed in the available space. We refer to items useful for this purpose as "migrants".

Definition 9. *Fix an $\epsilon_4 > 0$. A* migrant *for bin k is an item j such that $\sum_{i=1}^{k-1} p_{ij} \geq 1 - \epsilon_4$.*

We choose ϵ_4 such that $\epsilon_3 << \epsilon_4^3$ but ϵ_4 is still very small. Therefore, at the moment it is considered for bin k, a migrant is available with probability roughly $1/e$. This imposes constraints on the other ϵ's and consequently our final improvement ($\simeq \epsilon_1^3$). We can set for example $\epsilon_4 = 0.001$ which implies $\epsilon_1 \simeq 10^{-60}$ and then our final improvement is on the order of 10^{-180}.

Note that being migrant is only defined with respect to a certain bin. Every item is migrant for some bins, since $\sum_{i=1}^{n} p_{ij} > 1 - \epsilon_1$ (balanced fractional solution). Therefore, we can assume that an $\Omega(\epsilon_4)$-fraction of the LP value comes from migrants. Consequently, at least an $\Omega(\epsilon_4)$-fraction of bins (by LP contributions) have an $\Omega(\epsilon_4)$-fraction of their value in migrants. Let's call these bins "flexible". We will prove that each of them can gain a constant fraction of its LP share in addition to what it would get under the Sequential Allocation Algorithm.

For each set S which could be assigned to bin k, let's define $M_k(S)$ to be the migrants for bin k in S. We will drop the index when it's clear which bin we are referring to. We can assume that the value of these items is either $w(M_k(S)) = \Omega(\epsilon_4)V_k$ or $M_k(S) = \emptyset$. (For sets with migrants of value less than $\Omega(\epsilon_4)V_k$, we set $M_k(S) = \emptyset$; this decreases the contribution of migrants only by a constant factor.) Let's call the sets with nonempty $M_k(S)$ "flexible" for bin k. We have

$$\sum_{S \in \mathcal{F}_k} x_{k,S} w(M_k(S)) = \Omega(\epsilon_4) V_k$$

where $\mathcal{F}_k$ is the collection of flexible sets for bin k. Suppose that the probability of sampling a flexible set for bin k is

$$\sum_{S \in \mathcal{F}_k} x_{k,S} = r_k.$$

Note that since migrants should contribute at least an $\Omega(\epsilon_4)$-fraction of the bin's value V_k, this probability must be $r_k = \Omega(\epsilon_4)$. Here, we use the fact that there are no sets of value significantly exceeding V_k. The probability of sampling each individual set is at most $\epsilon_1 << \epsilon_4$ which allows us to assume that we can split the collection of flexible sets into three (roughly) equal parts in terms of probability. Let's select a portion of the flexible sets $S \in \mathcal{F}_k$ with the largest sizes of $M(S)$ and denote them by $\mathcal{A}_k$. For each set $S \in \mathcal{A}_k$, order the remaining flexible sets $S' \in \mathcal{F}_k \setminus \mathcal{A}_k$ by the size of $M(S') \setminus M(S)$, and partition them into two parts $\mathcal{B}_k(S), \mathcal{C}_k(S)$, so that for any $S' \in \mathcal{B}_k(S), S'' \in \mathcal{C}_k(S)$, we have $size_k(M(S') \setminus M(S)) \geq size_k(M(S'') \setminus M(S))$. We make these parts roughly equal so that

$$\sum_{S \in \mathcal{A}_k} x_{k,S} \geq \sum_{S' \in \mathcal{B}_k(S)} x_{k,S'} \geq \sum_{S'' \in \mathcal{C}_k(S)} x_{k,S''} \geq \frac{r_k}{4}.$$

The purpose of this partitioning is that we will be able to switch migrants between sets: any $M(S')$ for $S' \in \mathcal{B}_k(S)$ fits into the space occupied by $M(S), S \in \mathcal{A}_k$, and any $M(S'') \setminus M(S)$ for $S'' \in \mathcal{C}_k(S)$ fits into the space of $M(S') \setminus M(S)$ for $S' \in \mathcal{B}_k(S)$. This allows a scheme of switching migrants that achieves an approximation factor strictly better than $1 - 1/e$.

The Improved Sequential Allocation Algorithm.

- Order the bins by $V_1 \leq V_2 \leq V_3 \leq \ldots$ For bin k, let U_k denote the items allocated to previous bins.
- Let bin k sample a random set S_k from its distribution. If k is a flexible bin, sample two more sets S'_k, S''_k.
- If $S_k \in \mathcal{A}_k$ and $S'_k \in \mathcal{B}_k(S_k)$, pack $(S_k \cup M(S'_k)) \setminus U_k$ if possible into bin k.
- If $S'_k \in \mathcal{A}_k$, $S_k \in \mathcal{B}_k(S'_k)$, $S''_k \in \mathcal{C}_k(S'_k)$ and $(S'_k \cup M(S_k)) \setminus U_k$ fits in bin k (as in the previous case but with the roles of S_k and S'_k exchanged), then pack $(S_k \setminus (M(S_k) \setminus M(S'_k)) \cup (M(S''_k) \setminus M(S'_k))) \setminus U_k$ into bin k. This set is guaranteed to fit because $size_k(M(S''_k) \setminus M(S'_k)) \leq size_k(M(S_k) \setminus M(S'_k))$ due to the properties of $\mathcal{B}_k(S'_k)$ and $\mathcal{C}_k(S'_k)$.
- In all other cases, allocate $S_k \setminus U_k$ to bin k and go to the next bin.

Theorem 10. *For any balanced fractional solution where item values are uniform and there are no precious sets, the Improved Sequential Allocation Algorithm achieves expected value at least $(1 - 1/e + \Omega(\epsilon_4^5)) \cdot LP$.*

We only sketch the proof here. Consider a flexible bin k. For each pair of sets $S_k \in \mathcal{A}_k$, $S'_k \in \mathcal{B}_k(S)$, we calculate the expected size of $(M(S_k) \cup M(S'_k)) \setminus U_k$ by averaging over the previously allocated items U_k. Since the size of $M(S'_k)$ is at most the size of $M(S_k)$, and each migrant appears in U_k with probability at least $1 - e^{-(1-\epsilon_4)}$, we have

$$\mathbb{E}_{U_k}[size_k((M(S_k) \cup M(S'_k)) \setminus U_k)]$$

$$\leq e^{-(1-\epsilon_4)} size_k(M(S_k) \cup M(S'_k)) \leq \frac{4}{5} size_k(M(S_k)).$$

By Markov's inequality, we get

$$\Pr_{U_k}[size_k((S_k \cup M(S'_k)) \setminus U_k) \leq 1] \geq \frac{1}{5}.$$

Thus for each fixed pair (S_k, S'_k) like this, the algorithm succeeds in packing $(S_k \cup M(S'_k)) \setminus U_k$ with constant probability. We call U_k "favorable" for the pair (S_k, S'_k) if this occurs. Now consider what happens when the same pair of sets is selected by the bin in the opposite order, i.e. the roles of S_k and S'_k are switched. Now, $S'_k \in \mathcal{A}_k$ and $S_k \in \mathcal{B}_k(S'_k)$. Suppose U_k is favorable for (S'_k, S_k) and in the previous case, we managed to pack $(M(S_k) \setminus M(S'_k)) \setminus U_k$ in addition to what the standard algorithm would have packed. Now when the algorithm samples (S_k, S'_k), we can afford to give up this set and take only $(S_k \setminus (M(S_k) \setminus M(S'_k))) \setminus U_k$. By averaging the two cases, this would bring us back to a random allocation equivalent to the standard algorithm. In addition, however, our improved algorithm allocates $(M(S''_k) \setminus M(S'_k)) \setminus U_k$. This happens whenever $S'_k \in \mathcal{A}_k$, $S_k \in \mathcal{B}_k(S'_k)$, $S''_k \in \mathcal{C}_k(S'_k)$ and U_k is favorable for the pair (S'_k, S_k). Since the set $S''_k \in \mathcal{C}_k(S'_k)$ is sampled independently of everything else, we will gain a definite improvement if there is at least a constant fraction of $\mathcal{C}_k(S'_k)$ still available for bin k. For this purpose, we use the following concentration result.

Lemma 11. *Let A be a finite set with a weight function $f : A \to \mathbb{R}_+$, $f(A) = 1$; $\alpha, \lambda > 0$ and $0 < \epsilon < 1/2$. Let $S_1, S_2, \ldots, S_q$ be independent random subsets of A where $\Pr[j \in S_i] = p_{ij}$ and*

- $\forall i; \forall j \in A; p_{ij} \leq \epsilon$.
- $\forall j \in A; \sum_{i=1}^{q} p_{ij} \leq 1$.
- *We always have $f(S_i) \leq \alpha$.*

Then

$$\Pr\left[f\left(\bigcup_{i=1}^{q} S_i\right) > 1 - e^{-(1+\epsilon)} + \lambda\right] < \frac{\alpha + O(\epsilon)}{\lambda^2}.$$

This lemma can be proved using the second moment and Chebyshev's inequality. We apply it to a function f which represents the contribution of migrants in $\mathcal{C}_k(S'_k)$ to bin k, normalized to sum up to 1. The sets $S_1, S_2, \ldots$ are the sets of items allocated to preceding bins. The crucial point here is that any set S_i allocated to bin $i < k$ has value at most $20V_i \leq 20V_k$; since $V_k = \sum_j p_{kj} w_{kj}$ and $p_{kj} \leq \epsilon_1$, the loss of LP value for bin k due to items in S_i can be at most $20\epsilon_1 V_k$. This is very small even compared to the value of migrants in $\mathcal{C}_k(S'_k)$ which is $\Omega(\epsilon_4 V_k)$. Therefore, Lemma 11 implies that a constant fraction of the value in migrants is still available for bin k, say with probability $9/10$ (with respect to U_k). Even if this event is negatively correlated with the event that U_k is favorable for (S'_k, S_k), which happens with probability at least $1/5$, there is probability at least $1/10$ that both events occur. Thus each flexible bin k gains a constant fraction of its value V_k.

5 Hardness of approximation for *SMW*

In this section we show that there is some constant $\epsilon > 0$ such that it is NP-hard to approximate the Maximum Submodular Welfare problem within a ratio better than $1 - \epsilon$, even using a demand oracle. Previously it was shown in [8] that the maximum submodular welfare problem is hard to approximate within a ratio better than $1 - 1/e$; however, there 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. In particular, it is NP-hard for players to answer demand queries. We remark that a powerful oracle model can bypass previous hardness results; e.g., there is a polynomial time incentive compatible mechanism based on "fair division queries" that extracts the maximum welfare, provided only that all players have the same utility function (submodular or not). Our new hardness results are the following.

Theorem 12. *There is some constant $\epsilon > 0$ such that it is NP-hard to approximate the Maximum Submodular Welfare problem within a ratio better than $1 - \epsilon$ in the following cases.*

1. *When all players have constant size utility functions. In particular, each player gets nonzero utility only from 15 specific items.*
2. *When all players have the same utility function. Moreover, the utility function is constant for sets of size more than 7. The value of ϵ here is $1/(12 \cdot 23)$.*
3. *When there are only two players. Moreover, their utility functions are separable; i.e., the items can be partitioned into disjoint classes C_j of constant size such that $w_i(S) = \sum_j w_i(S \cap C_j)$.*

In all these cases, demand queries can be answered efficiently. The proofs are based on reductions from Max 3-coloring-5 and Max k-cover. We present only the proof of the first hardness result. We believe that the use of constant size utility functions is the most natural way to rule out a possible approximation result even in a very strong oracle model. 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.

Proof. We use the following NP-hardness result [7]: *There is an $\epsilon > 0$ such that given a 5-regular graph, it is NP-hard to distinguish between the case where its vertices can be legally 3-colored, and the case where every 3-coloring makes an ϵ-fraction of edges illegally colored.*

Given a 5-regular graph with n vertices and m edges (hence $2m = 5n$) we reduce it to the following *SMW* instance. 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. The utility function of the player p_v who is associated with vertex v is nonzero on 15 items, the items associated with the edges incident with v. There are three special sets of size 5: the monochromatic subsets of these 15 items. Each special set has value 5, while any other set of size 5 has value 4.5. Any set of size $b < 5$ has value b, and any set of size $b > 5$ has value 5. These utility functions are submodular.

On positive instances, we can legally 3-color the graph. Then each vertex player gets the five items associated with her chosen color. Each edge player can get the unallocated item associated with her edge. 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 it can contribute more). Likewise, 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 define a vertex coloring derived from the colors of these items. For vertex players whose items are all of the same color, assign the same color to the vertex. Such players are maximally happy. Assume that k vertex players are "unhappy" in the sense that their 5 items are not monochromatic. For such vertices, choose a majority color among the 5 items and assign it to the vertex. For each unhappy vertex player, there can be up to 3 edges with an illegal coloring. For all other edges, the coloring of the two endpoints is derived from two items of different colors, therefore their coloring is legal. The number of illegally colored edges is at least ϵm and at most $3k$, i.e. $3k \geq \epsilon m$. Each unhappy player loses value $1/2$, hence the maximum welfare is at most $3m - \epsilon m/6$, showing that maximum submodular welfare cannot be approximated within a factor better than $1 - \epsilon/18$. $\square$

References

[1] Steven J. Brams, Alan D. Taylor. Fair Division: From cake-cutting to dispute resolution. Cambridge University Press, 1996.

[2] Chandra Chekuri and Sanjeev Khanna. A PTAS for the multiple knapsack problem. Proceedings of SODA 2000: 213–222.

[3] Shahar Dobzinski, Noam Nisan and Michael Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. Proceedings of STOC 2005: 610–618.

[4] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. Proceedings of SODA 2006: 1064–1073.

[5] Lisa Fleischer, Michel X. Goemans, Vahab Mirrokni and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. Proceedings of SODA 2006: 611–620.

[6] Uriel Feige. On maximizing welfare when utility functions are subadditive. Proceedings of STOC 2006: 41–50.

[7] Uriel Feige, Magnus M. Halldorsson, Guy Kortsarz and Aravind Srinivasan. Approximating the domatic number. SIAM J. Comput. 32(1): 172–195 (2002).

[8] Subhash Khot, Richard Lipton, Evangelos Markakis and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. Proceedings of WINE 2005.

[9] Benny Lehmann, Daniel J. Lehmann and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In ACM Conference on Electronic Commerce 2001: 18–28.

[10] Daniel J. Lehmann, Liadan O'Callaghan and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. In ACM Conference on Electronic Commerce 1999.

[11] Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auctions. Proceedings of IJCAI 1999.

[12] David Shmoys and Eva Tardos. An approximation algorithm for the generalized assignment problem. Math. Programming 62(3), 461-474 (1993).