Auctions with Budget Constraints ⋆

Nir Andelman and Yishay Mansour

School of Computer Science
Tel-Aviv University
{andelman,mansour}@cs.tau.ac.il

Abstract. In a combinatorial auction k different items are sold to n bidders, where the objective of the seller is to maximize the revenue. The main difficulty to find an optimal allocation is due to the fact that the valuation function of each bidder for bundles of items is not necessarily an additive function over the items. An *auction with budget constraints* is a common special case where bidders generally have additive valuations, yet they have a limit on their maximal valuation. Auctions with budget constraints were analyzed by Lehmann, Lehmann and Nisan [11], as part of a wider class of auctions, where they have shown that maximizing the revenue is NP-hard, and presented a greedy 2-approximation algorithm. In this paper we present exact and approximate algorithms for auctions with budget constraints. We present a randomized algorithm with an approximation ratio of $\frac{e}{e-1} \approx 1.582$, which can be derandomized. We analyze the special case where all bidders have the same budget constraint, and show an algorithm whose approximation ratio is between 1.3837 and 1.3951. We also present an FPTAS for the case of a constant number of bidders.

1 Introduction

Auctions are a popular mechanism for selling and purchasing goods when traditional market mechanisms based on supply and demand are not satisfying, or are not implementable. In a combinatorial auction, a number of items is sold to a group of bidders whose valuation function may not be additive, meaning that a valuation of a bundle of items may express relations between subsets of items.

Mechanisms dealing with combinatorial auctions face several challenges. For instance, if there are k items then each bidder has to submit 2^k bids to fully express her valuation function. This exponential growth would make such an approach infeasible in practice. An alternative typical approach is to assume that bidders have simple preferences that can be expressed compactly.

Another important issue is computational, namely, deciding on the allocation of the items to the bidders. The allocation should maximize an objective function of the auctioneer, which is usually either the auctioneer's revenue or the economic efficiency. Finding an optimal allocation

⋆ Partially supported by a grant of the Israeli Science Foundation

is computationally hard in general, although it is tractable in certain cases [14, 15, 17]. There are various methods to tackle this difficulty, such as finding an approximate allocation rather than the optimal one [11, 6] or developing mechanisms which work well in practice, though do not necessarily have a formal guarantee [6, 10, 14, 16].

Lehmann, Lehmann and Nisan [11] have concentrated on combinatorial auctions where bidders' valuations are known to be subadditive. A very natural subclass of subadditive valuations is decreasing marginal utilities (also known as submodular valuations), where the valuation a bidder gives to an item monotonically decreases as the set of items he already purchased grows. Formally, if $V(\cdot)$ denotes the valuation function of a bidder, then for any two bundles S and T such that $S \subseteq T$ and for any item x such that $x \notin T$, we have $V(S \cup \{x\}) - V(S) \geq V(T \cup \{x\}) - V(T)$. Lehmann et al. [11] presented a greedy algorithm for auctions with decreasing marginal utilities, and proved that it is a 2-approximation.

In this paper we concentrate on auctions with budget constraints, which are a special case of auctions with decreasing marginal utilities. In such auctions bidder i has a budget limit of d_i, and the valuations are additive as long as the limit is not met. Once the limit is reached, the valuation equals the budget limit. Namely, the valuation of bidder i for a bundle A is given by $\min(d_i, \sum_{j \in A} b_{ij})$, where b_{ij} is the bid of bidder i for item j. Art dealers and collectors, for example, are likely to have valuations with budget constraints, since they valuate each item separately, but have a limit on their total expenses.

Bidding in auctions with budget constraints is a rather concise process, since each bidder has to submit only k bids, where k is the number of auctioned items, and also submit the budget limit. Each bidder is charged according to the valuation of the items allocated to her. Our goal is to maximize the revenue of the auctioneer.

We prove that finding an optimal allocation that maximizes the revenue is NP-hard even if there are only two bidders with identical valuations. We show that an exact solution can be found in time $O(\min(n4^k, k^2 4^k + nk))$, where k denotes the number of items and n denotes the number of bidders. If the number of bidders is constant, there is also a pseudo-polynomial algorithm.

Our main results are polynomial time approximation algorithms. We present a randomized algorithm which is a $\frac{e}{e-1} \approx 1.582$ approximation, and also derandomize it. We then exhibit improved approximation ratios when all bidders have the same budget constraint (but possibly different valuations), and prove that the approximation ratio is between 1.3837 and 1.3951. We also present a FPTAS for the case of a constant number of bidders. (We remark that the greedy allocation algorithm of [11] for auctions with budget constraints, even if there are only two bidders, has approximation ratio 2.)

An issue that we have not covered in this paper is the effect of the pricing scheme on the bidding strategies of the bidders. Bidders may lie about their valuations if it suits their personal interest, and therefore there is an incentive to search for a pricing mechanism that will induce truthfulness, i.e. reporting the true valuation is a dominant strategy. Designing truthful mechanisms that use approximate allocations is a challenging task, and was successfully accomplished under certain assumptions on the bidders' valuations and the auctioned items [1, 2, 4, 5, 7, 13]. In contrast, we concentrate only on maximizing the auctioneer's revenue given the bids of the bidders. This approach is reasonable if the bidders are indifferent to the allocation mechanism, as long as the pricing is according to their revealed bids.

The rest of this paper is organized as follows: Section 2 formally defines the auction with budget constraints problem, Section 3 presents algorithms for exact solutions and hardness results, and Section 4 presents algorithms for approximate solutions.

2 Model and Notations

An auction with budget constraints consists of n bidders and k items. Let b_{ij} denote the bid of bidder i for item j, which is the maximal price that the bidder is willing to pay for this item, assuming that the budget constraint is not met. Let d_i denote the budget constraint of bidder i.

Let $z_{ij} \in \{0, 1\}$ denote the allocation of the items, where $z_{ij} = 1$ if bidder i receives item j and $z_{ij} = 0$ otherwise. Given an allocation, the price that each bidder is willing to pay is $p_i = \min\{d_i, \sum_{j=1}^{k} z_{ij}b_{ij}\}$. The objective of allocation with budget constraints is maximizing the total payment of all bidders.

This allocation problem can be presented formally as the following Integer Programming (IP) problem:

$$\begin{array}{lll} \max \sum_{i=1}^{n} p_i & \text{s.t.} & /{*}\text{maximizing revenue } {*}/ \\ p_i \leq \sum_{j=1}^{k} z_{ij}b_{ij} & 1 \leq i \leq n & /{*}\text{additive valuations } {*}/ \\ p_i \leq d_i & 1 \leq i \leq n & /{*}\text{budget constraints } {*}/ \\ \sum_{i=1}^{n} z_{ij} \leq 1 & 1 \leq j \leq k & /{*}\text{one copy of each item}{*}/ \\ z_{ij} \in \{0, 1\} & 1 \leq j \leq k, 1 \leq i \leq n & /{*}\text{integral allocation } {*}/ \end{array} \quad (1)$$

Without loss of generality, we assume that the budget constraint is consistent with the bids, i.e. $d_i \geq b_{ij}$, $\forall i, j$, $1 \leq i \leq n$, $1 \leq j \leq k$, and that the budget constraint is effective, i.e. $d_i \leq \sum_{j=1}^{k} b_{ij}$, $\forall i$, $1 \leq i \leq n$. We also assume that all bids and budget constraints are non negative.

3 Exact Solutions

Lehmann, Lehmann and Nisan [11] have analyzed auctions where the bidders have submodular valuations, meaning that the marginal utility that each bidder gains for any item decreases as the set of items already allocated to this bidder increases. They prove that finding an optimal allocation is NP-hard even if there are only two bidders with additive valuations up to budget constraints. The following theorem (the proof is based on a reduction from $PARTITION$ [9]), strengthens this result.

Theorem 1. *Finding the optimal allocation for an auction with budget constraints is NP-hard even for two bidders with identical bids and budget constraints.*

Using dynamic programming [3], an exact solution can be found in a time complexity that is exponential in the number of items. In the i-th stage of the dynamic programming, optimal allocations of any subset of the k items are computed over the first i bidders, by using the optimal allocations from the previous stage. This process yields the following:

Theorem 2. *An exact optimal allocation for an auction with budget constraints can be found in time complexity of* $O(n4^k)$

For $n >> k$, since in an optimal allocation each item is sold only to one of the k highest bidders, the time complexity can be reduced to $O(k^2 4^k + nk)$. When the number of bidders is a constant, then a pseudo-polynomial algorithm based on dynamic programming exists. The details are omitted due to space constraints.

4 Approximate Solutions

In this section we present our main results, which include approximation algorithms for the case that the number of bidders is constant and for the general case. We also present improved bounds when all bidders have the same budget constraint, yet possibly different valuations.

4.1 Constant Number of Bidders

If the number of bidders is constant then there exists a fully polynomial time approximation scheme (FPTAS), meaning that the approximate allocation is at least $1 - \epsilon$ (for any $\epsilon > 0$) times the optimal allocation, and the running time is polynomial in the number of items k and in $\frac{1}{\epsilon}$. (The algorithm is an adaptation of the FPTAS for the scheduling problem with unrelated machines of Horowitz and Sahni [8].)
The algorithm uses sets of tuples $(v_1, a_1, v_2, a_2, \ldots, v_n, a_n, t)$ to construct the approximation. Each tuple represents an allocation of a subsets of items to the users. Let v_i be the benefit of bidder i from the items allocated to her, and let a_i be a bit-vector indicating which items were allocated to this bidder. Let t be the total benefit of the partial allocation. The set S_j contains tuples representing allocations of the first j items.
Algorithm Dynamic Programming Allocation (DPA)

1. Let $\gamma = \max\{d_i\}$.
2. Divide the segment $[0, n\gamma]$ into $\frac{nk}{\epsilon}$ equal intervals of length $\frac{\gamma\epsilon}{k}$ each.
3. Initialize $S_0 = (0, 0, \ldots, 0)$.
4. For each item j,
   (a) Construct S_j from S_{j-1}, by replacing each tuple s with n tuples $s_1, \ldots, s_n$, where tuple s_i represents the same allocation as s, with item j allocated to bidder i.
   (b) For any tuple $s = (v_1, a_1, \ldots, v_n, a_n, t)$, if there is a tuple $s' = (v'_1, a'_1, \ldots, v'_n, a'_n, t')$ such that v_i, and v'_i are in the same interval for every i, and $t' \leq t$, remove s' from S_j.
5. Return the allocation represented by the tuple from S_k with the largest total benefit.

Analyzing Algorithm DPA yields the following (proof omitted):

Theorem 3. *Algorithm* DPA *is a* FPTAS *for the auction with budget constraints problem with a constant number of bidders.*

Since DPA requires space for representing $O\left(\left(\frac{k}{\epsilon}\right)^n\right)$ tuples, its space complexity is polynomial in $\frac{1}{\epsilon}$. We note that there exists an approximation algorithm with space complexity polynomial in $\log \frac{1}{\epsilon}$, which is the space required for representing the value ϵ. The approximation, however, is a *PTAS*, but not a *FPTAS*, i.e. the running time is not polynomial in $\frac{1}{\epsilon}$. The description of this algorithm is omitted due to space considerations.

4.2 General Number of Bidders

In this section we analyze an algorithm with a provable approximation ratio of 1.582 for an arbitrary number of bidders.
In order to find an approximation to this allocation problem, we use the following Linear Programming (LP), which solves a relaxed version of the original Integer Programming (1):

$$\begin{array}{ll} \max \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} b_{ij} \text{ s.t.} & \\ \sum_{j=1}^{k} x_{ij} b_{ij} \le d_i & 1 \le i \le n \\ \sum_{i=1}^{n} x_{ij} \le 1 & 1 \le j \le k \\ x_{ij} \ge 0 & 1 \le j \le k, 1 \le i \le n \end{array} \tag{2}$$

The relaxation replaces the original Boolean variables z_{ij} by variables x_{ij}, indicating a fractional assignment of items. The equations are simplified by removing the p_i variables, which become redundant in the LP. As is the case with any relaxation method, our main task is to round the fractional assignments to an integral solution.

Algorithm Random Rounding (RR) is an approximation algorithm for the auction with budget constraints problem with a variable number of bidders.

Algorithm RR
1. Find an optimal fractional allocation using LP.
2. For each item j, assign it to bidder i with probability x_{ij}.

Obviously, Algorithm RR is randomized, outputs a feasible allocation, and terminates in polynomial time complexity. The following theorem proves the approximation ratio of the algorithm.

Theorem 4. *The expected approximation ratio of Algorithm* RR *is at most* $\frac{e}{e-1} \approx 1.582$.

Proof. Let Z_i be a random variable indicating the revenue from bidder i after the allocation. The expected total revenue is $\sum_i E(Z_i)$. Therefore, it is sufficient to prove that for any bidder, the expected ratio between the benefit of the fractional allocation and the final integer allocation is at most $\frac{e}{e-1}$.

We analyze separately the expected revenue from each bidder i. Without loss of generality, when considering bidder i we normalize the bids and the budget constraint such that $d_i = 1$, in order to simplify the calculations. Let $B_i = \sum_j x_{ij} b_{ij}$ indicate the revenue from bidder i generated by the fractional assignment. We prove that $E(Z_i) \geq \frac{e-1}{e} B_i$. Without loss of generality, we assume that the indices of the items assigned (fully or fractionally) to bidder i are $1, \ldots, r$.

Let X_{ij} be a random variable, which indicates whether item j is allocated to bidder i. The random variable X_{ij} is 1 with probability x_{ij} and 0 otherwise. The expected revenue from bidder i is therefore $Z_i = \min(1, \sum_{j=1}^{r} X_{ij} b_{ij})$.

Suppose we replace b_{i1} and x_{i1} with $\hat{b}_{i1} = 1$ and $\hat{x}_{i1} = b_{i1} x_{i1}$. The size of the fractional item assigned by the Linear Programming is $\hat{b}_{i1} \hat{x}_{i1} = b_{i1} x_{i1}$, meaning that the benefit of bidder i in the fractional assignment remains unchanged. We observe the effect of replacing X_{i1} with the corresponding $\hat{X}_{i1}$ on Z_i, when the remaining variables are kept constant. We denote $Z_{i1} = \min(1, \sum_{j=2}^{r} X_{ij} b_{ij})$ and observe that $Z_i - Z_{i1}$ is a random variable that denotes the marginal contribution of X_{i1} to the total revenue. We examine how replacing X_{i1} with $\hat{X}_{i1}$ effects the possible values of $Z_i - Z_{i1}$.

1. If $0 \leq Z_{i1} \leq 1 - b_{i1}$: The marginal contribution of X_{i1} is either 0 or b_{i1}, so the expected contribution is $b_{i1} x_{i1}$. The contribution of $\hat{X}_{i1}$ is either 0 or $1 - Z_{i1} \leq 1$, so the expected marginal contribution is at most $\hat{x}_{i1} = b_{i1} x_{i1}$
2. If $1 - b_{i1} < Z_{i1} \leq 1$: The marginal contribution of both variables is either 0 or $1 - Z_{i1}$. The expected marginal contribution of X_{i1} is $(1 - Z_{i1}) x_{i1}$. The expected marginal contribution of $\hat{X}_{i1}$ is $(1 - Z_{i1}) x_{i1} b_{i1} \leq (1 - Z_{i1}) x_{i1}$.

In both cases, by replacing X_{i1} with $\hat{X}_{i1}$ we can only decrease $E(Z_i)$, without changing B_i. Similarly, for each $2 \leq j \leq r$ we replace b_{ij} with $\hat{b}_{ij} = 1$, x_{ij} with $\hat{x}_{ij} = b_{ij} x_{ij}$ and X_{ij} with $\hat{X}_{ij}$. Since each replacement does not increase $E(Z_i)$, we have $E(\hat{Z}_i) = E(\min(1, \sum_{j=1}^{r} \hat{X}_{ij})) \leq E(\min(1, \sum_{j=1}^{r} X_{ij}))$.

Since for each j, $\hat{X}_{ij}$ is either 0 or 1, then $\hat{Z}_i$ is also either 0 or 1. Therefore:

$$E(\hat{Z}_i) = P(\hat{Z}_i = 1) = 1 - P(\hat{Z}_i = 0) = 1 - \prod_{j=1}^{r} (1 - \hat{x}_{ij}) \tag{3}$$

The expectation is minimized when $\prod_{j=1}^{r} (1 - \hat{x}_{ij})$ is maximized. Under the constraint $B_i = \sum_{j=1}^{r} b_{ij} x_{ij} = \sum_{j=1}^{r} \hat{x}_{ij}$, the maximum is when all $\hat{x}_{ij}$ are equal to B_i / r. Therefore:

$$E(\hat{Z}_i) \geq 1 - \left(1 - \frac{B_i}{r}\right)^r \geq 1 - e^{-B_i} \geq B_i (1 - e^{-1}) \tag{4}$$

The last inequality follows since $1 - e^{-x} \geq x(1 - e^{-1})$ for $x \in [0, 1]$. Therefore, we have:

$$\frac{B_i}{E(Z_i)} \leq \frac{B_i}{E(\hat{Z}_i)} \leq \frac{e}{e-1} \approx 1.582 \tag{5}$$

Since the approximation holds for the expected revenue of each bidder separately, it also holds for the expected total revenue. □

The following theorem claims that in the worst case, RR has an approximation ratio of $\frac{e}{e-1}$.

Theorem 5. *The expected approximation ratio of Algorithm* RR *is at least* $\frac{e}{e-1} \approx 1.582$.

Proof. The lower bound of $\frac{e}{e-1}$ for the approximation ratio is achieved in the following setting: $n+1$ bidders, $A_0, A_1, \ldots, A_n$, compete on $2n$ items, $I_1, I_2, \ldots, I_{2n}$. Bidder A_0 has a budget constraint $d_0 = n$, bids $b_{1i} = n$ for items $I_1, \ldots I_n$, and bids $b_{1j} = 0$ for items $I_{n+1}, \ldots I_{2n}$. The other bidders all have a budget constraint of $d_i = n^{-1}$. For $1 \leq i \leq n$, bidder A_i bids $b_{ij} = n^{-1}$ for item I_i, $b_{ij} = n^{-2}$ for item I_{n+i}, and $b_{ij} = 0$ for the other items.
The Linear Programming finds a unique fractional assignment which satisfies all budget constraints: for each $1 \leq i \leq n$, bidder A_i receives item I_{n+i} (fully) and a $\frac{n-1}{n}$ fraction of item I_i. The remaining fraction of $\frac{1}{n}$ of item I_i is given to bidder A_0. The revenue from the fractional assignment is $n+1$.
Algorithm RR achieves an expected revenue of $n(1-(1-\frac{1}{n})^n) + \frac{n^2-n+1}{n^2}$ which is approximately $n(1-e^{-1})+1$ for sufficiently large n. □

From Theorems 4 and 5 we have the following corollary.

Corollary 1. *the approximation ratio of Algorithm* RR *is exactly* $\frac{e}{e-1}$.

4.3 Derandomized Rounding

A natural derandomization of Algorithm RR which maintains the $\frac{e}{e-1}$ approximation ratio would be to sequentially assign each item such that the expected revenue maintains above the expectation. Although calculating exactly the expected revenue may be computationally hard, this difficulty can be resolved by replacing the exact expected revenue with lower bounds, which are derived by techniques similar to those used in the proof of Theorem 4.
This section discusses alternative deterministic algorithms for rounding the fractional assignments. Among all possibilities to round the fractions in a solution to a LP instance, we refer to the rounding with the highest total revenue as an *optimal rounding*. Obviously, an algorithm that returns an optimal rounding has an approximation ratio of at most any other rounding algorithm, including RR.
A convenient method to observe the output of the LP is by constructing a bipartite graph $G = (N, K, E)$, where the nodes N and K correspond to the bidders and items, respectively, and the edges E indicate that a bidder was assigned an item (or a fraction of an item).
The original allocation problem can be divided into several subproblems, where each subproblem consists only bidders and items from the same component in G. Solving the Linear Programming separately for each

subset of bidders and items should return the same allocation. Therefore, we may concentrate on each component of G separately.

The LP includes $nk + n + k$ constraints using nk variables. In the solution of the LP, at least nk of the constraints are satisfied with equality, meaning that at most $n + k$ of the x_{ij} variables are non-zero. Each non zero variable x_{ij} matches one edge in G, and therefore G contains at most $n+k$ edges. On the other hand, since G is connected and has $n+k$ nodes, it must have at least $n+k-1$ edges. Therefore, G is either a tree, or has exactly one cycle.

The following lemma claims that if G has a cycle, then the optimal solution to the LP can be modified, such that one edge will be deleted from G, and therefore G will be a tree while maintaining the optimality of the solution (proof omitted).

Lemma 1. *There is a node in the polyhedron of the LP which maximizes the objective function and induces a graph without cycles, and can be found in polynomial running time.*

By applying Lemma 1, the fixed graph is a tree. Therefore, out of at most $n + k$ constraints in the LP that are not satisfied with equality, exactly $n+k-1$ of them are of type $x_{ij} \geq 0$. This means that at most one of the non-trivial constraints is not strict: Either there is at most one item which is not fully distributed, yet all bidders reach their budget constraint, or there is at most one bidder that doesn't reach its budget constraint, yet all items are fully distributed. We have the following observation:

Observation 1 *For each component in G, at most one bidder has not met its budget constraint.*

Algorithm Semi Optimal Rounding (SOR), returns an allocation that has a total revenue of at least $1 - \epsilon$ times the optimal rounding, for any $\epsilon > 0$. The algorithm applies a recursive 'divide and conquer' process on the tree graph representing the fractional allocations to choose a nearly optimal rounding.

Algorithm SOR(G, ϵ)

1. Find an optimal fractional solution using LP.
2. Construct a bipartite graph G representing the LP allocation.
3. For each component G_i in G:
   (a) If G_i contains a cycle, convert G_i to a tree by modifying the LP solution.
   (b) Apply ROUND(G_i, ϵ)

We use the following notation in process ROUND: for a tree T and a node v, let $T_{v,i}$ denote the i-th subtree rooted at v. Let $T^+_{v,i}$ denote the tree containing the subtree $T_{v,i}$, the node v and the edge connecting v to $T_{v,i}$. When node v denotes a bidder, let u_i denote the i-th item node shared by v and bidders in $T_{v,i}$. Let $T^-_{v,i}$ denote the same tree as $T_{v,i}$, with item u_i replaced with a dummy item u^-_i, which has zero valuations from all bidders.

Process ROUND(T, ϵ)
1. If T Includes only one bidder, allocate all the items to this bidder. If there are no bidders in T, return a null assignment.
2. Otherwise, find vertex $v \in T$, which is a center of T.
3. If v represents an item, for each subtree $T_{v,i}$ recursively compute ROUND($T_{v,i}$, ϵ) and ROUND($T^+_{v,i}$, ϵ). Allocate v to a bidder such that the total revenue is maximized (explanation follows).
4. If v represents a bidder, for each subtree $T_{v,i}$ recursively compute ROUND($T_{v,i}$, $\frac{\epsilon}{2}$) and ROUND($T^-_{v,i}$, $\frac{\epsilon}{2}$). Find a combination of the partial allocations, whose revenue is at least $1 - \frac{\epsilon}{2}$ times an optimal combination (explanation follows).

When the central node v represents an item, combining the partial allocations of the subtrees is a simple process, since only one subtree may receive item v. Formally, we enumerate on v's neighbors to calculate $\max_j \Big(\text{ROUND}(T^+_{v,j}, \epsilon) + \sum_{i \neq j} \text{ROUND}(T_{v,i}, \epsilon)\Big)$.
However, if v represents a bidder, the number of combinations is exponential in the degree of v, and this is why an approximation is preferred over an exact solution. The approximation process is as follows:
Let r be the degree of node v. Let b_{vi} be the bid of bidder v for the item shared with bidders in the i-th subtree. Let $C_i = \text{ROUND}(T_{v,i}, \frac{\epsilon}{2})$ be the approximated revenue of rounding the i-th subtree when v does not get the i-th item it shares. Let $c_i = \text{ROUND}(T^-_{v,i}, \frac{\epsilon}{2})$ be the approximated revenue of rounding the i-th subtree when v gets the i-th item (while the bidders in $T^-_{v,i}$ share a dummy item with no value). We construct the following allocation problem with 2 bidders: The items are a subset of the original items, reduced to those allocated (partially or fully) to v, denoted as $I_1, I_2, \ldots, I_r$ plus r special items $I'_1, I'_2, \ldots, I'_r$. Bidder 1 has the same budget constraint as the bidder represented by v, and the same bids on $I_1, \ldots I_r$. Bidder 1 bids 0 on the special items $I'_1, \ldots, I'_r$. Bidder 2 has an unbounded budget constraint, and bids c_i for each special item I'_i, and $\max\{0, C_i - c_i\}$ for the original items I_i.
The new allocation problem is a reduction of the original rounding problem. Any rounding possibility matches an allocation with the same benefit. Therefore, if we approximate the reduced allocation solution, we get an approximation to the original rounding problem. Since there are only two bidders in the reduced problem, we use Algorithm DPA, of Theorem 3, to approximate an optimal allocation.
The following holds for SOR (proof omitted):

Theorem 6. *Algorithm* SOR *has an approximation ratio of at most* $\frac{e}{e-1} + \epsilon$ *and a polynomial running time for any* $\epsilon > 0$.

By applying both SOR and the sequential rounding discussed at the beginning of this section we can get rid of the additional factor of ϵ and guarantee an approximation ratio of $\frac{e}{e-1}$. This bound is not tight, as the lower bound of $\frac{4}{3}$ for rounding algorithms (presented in Section 4.5) holds also for this algorithm.

4.4 Bidders with Identical Budget Constraints

In this section we show improved approximation bounds for RR in the case where all bidders have the same budget constraint. The approximation ratio of $\frac{e}{e-1}$ is due to a bound of $1-(1-1/r)^r$ on the expected valuation of each bidder, where r is the number of items owned or shared by a bidder. Actually, by considering items fully assigned to the same bidder as one large item, we achieve a tighter bound of $1-(1-\frac{1}{a+1})^{a+1}$, where a is the number of partial assignments of items to a bidder. When a goes to infinity, the bound goes to $\frac{e-1}{e}$, however not all bidders will have infinitely many fractional items. If all bidders have identical budget constraints, we can use this property to derive a tighter analysis for RR. Assuming the bipartite graph G, constructed from the solution to LP has only one component, let G' be the subgraph of G where nodes corresponding to items that are allocated only to one bidder are removed. Therefore, G' represents only the items shared among several bidders. Let N_i denote the node in G' corresponding to bidder i and let K_j denote the node in G' corresponding to item j. We define the following sets:

Definition 1. *Let R_a be the bidders corresponding to the set of nodes in G' such that $\{N_i|Deg(N_i)=a\}$, and let S_a be the items corresponding to the set of nodes in G' such that $\{K_j|Deg(K_j)=a\}$.*

The sets R_a and S_a have the following property, which is based on the fact that G' is a tree:

Lemma 2. $|R_1| = 2 + \sum_{a\geq 3}(a-2)|R_a| + \sum_{a\geq 3}(a-2)|S_a|$

According to Lemma 2, the number of bidders who have only one fraction of an item is fairly large: There are two of these bidders to begin with. Each bidder that has more than two fractions of items enforces another bidder in R_1 for each extra fraction. Also, Each item shared between three or more bidders adds another bidder to R_1 for each share over the second. We can use this property to prove the following:

Theorem 7. *When all bidders have the same budget constraint, the approximation ratio of algorithm RR is at most $\frac{27}{19} \approx 1.421$.*

Proof. Without loss of generality assume that the identical budget constraint is 1, and that G' is a tree. For each bidder from R_a $(a>2)$ we match $a-2$ bidders from R_1. By Lemma 2 this is possible, leaving at least 2 bidders from R_1 unmatched.

We first assume all bidders reach their budget constraint. Bidders from R_2 are unmatched, and have an expected valuation of at least $1-(1-1/3)^3=\frac{19}{27}$ each. Bidders from set R_a have an expected valuation of at least $1-(1-1/(a+1))^{(a+1)}$, which is less than $\frac{19}{27}$ for $a\geq 3$, but they are matched with $a-2$ bidders from R_1 who have an expected valuation of at least $1-(1-1/2)^2=3/4$, each. On average, the expected valuation is larger than $\frac{19}{27}$, for any $a\geq 3$.

If not all bidders reach their budget constraint, by Observation 1 only one bidder u has not met the constraint. If u belongs to R_a, then the expected valuation of u is still at least $1-(1-1/(a+1))^{(a+1)}$ times the

valuation achieved by the fractional assignment. If bidder u participates in a match, the ratio between the total expected valuation of the bidders in the match and the fractional valuation will remain above $\frac{19}{27}$ as long as u does not belong to R_1. If $u \in R_1$ it is possible to replace u with an unmatched bidder from R_1, since by Lemma 2 at least 2 bidders that are in R_1 are unmatched.
For each group of matched bidders, the ratio between the total expected valuation to the fractional valuation is at least $\frac{19}{27}$. Unmatched bidders remain only in R_1 or R_2 and therefore also have an expected valuation of at least $\frac{19}{27}$ times the fractional valuation. Therefore, the approximation ratio is at most $\frac{27}{19} \approx 1.421$. □

Theorem 7 implies that bidders from R_2 are the bottleneck of the analysis, as they are not matched with bidders from R_1. If this bottleneck can be resolved, the approximation ratio could drop to 1.3951, which is induced by bidders from R_3, who are matched with bidders from R_1, therefore their average expectation is $\frac{1}{2}(\frac{3}{4} + \frac{175}{256}) \approx 0.7168 = (1.3951)^{-1}$. By using the sets S_a the following theorem claims that this improvement is indeed achievable.

Theorem 8. *When all bidders have the same budget constraint, the approximation ratio of algorithm RR is at most* 1.3951*.*

The following lower bound nearly matches the upper bound:

Theorem 9. *The approximation ratio of Algorithm* RR *with identical budget constraints is at least* 1.3837

4.5 Fractional Versus Integral Allocations

In this section we derive a general lower bound for algorithms that are based on solving the LP and rounding the fractional assignments. The following theorem proves a lower bound for any algorithm that uses the relaxed LP.

Theorem 10. *LP has an integrality ratio of at least $\frac{4}{3}$. Also, the optimal solution of the IP can be $\frac{4}{3}$ times any solution that is based on rounding nonzero fractional allocations of the LP.*

Proof. The integrality ratio if $\frac{4}{3}$ is achieved in the following case: Observe the following auction with 2 bidders, A and B, and 3 items x, y and z. Bidder A bids 1 for x, 0 for y and 2 for z and has a budget constraint of 2. Bidder B bids 0 for x, 1 for y and 2 for z and has a budget constraint of 2. Optimally, A get x, B gets y and either bidder gets z, and the revenue is 3. However, the LP produces an optimal fractional assignment, which divides z between both bidders, and achieves a revenue of 4.
The ratio of $\frac{4}{3}$ between the optimal integral solution and any other solution that is based on rounding the LP solution is achieved in a similar auction, but now both A and B bid 1 for x and y and bid 2 for z (the budget constraints remain 2). Optimally, A gets both x and y while B gets z, and the revenue is 4. However, the LP might produce a fractional assignment, such as x for A, y for B, and z divided between both bidders. The revenue is also 4, but any rounding technique will either grant z to A or to B, either way the revenue is 3. □

References

1. A. Archer, C. H. Papadimitriou, K. Talwar and E. Tardos, An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proc. 14th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2003, 205–214.
2. Y. Bartal, R. Gonen and N. Nisan, Incentive compatible multi unit combinatorial auctions. In *Proc. 9th Conf. on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2003
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, Second Edition. MIT Press, 2001.
4. K. Deshmukh, A. V. Goldberg, J. D. Hartline and A. R. Karlin, Truthful and competitive double auctions. In *Proc. 10th Ann. European Symp. on Algorithms (ESA)*, 2002, 361–373.
5. A. Fiat, A. V. Goldberg, J. D. Hartline and A. R. Karlin, Competitive generalized auctions. In *Proc. 34th Ann. ACM Symp. on Theory of Computing (STOC)*, 2002, 7-81.
6. Y. Fujishima, K. Leyton-Brown and Y. Shoham, Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1999, 548–553.
7. A. V. Goldberg and J. D. Hartline, Competitiveness via consensus. In *Proc. 14th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)* 2003, 215–222.
8. E. Horowitz and S. Sahni, Exact and approximate algorithms for scheduling nonidentocal processors. In *J. of the Association for Computing Machinery 23* 1976, 317–327.
9. R. M. Karp, Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenium Press, New York, 1972, 85–104.
10. K. Leyton-Brown, Y. Shoham and M. Tennenholtz, Bidding clubs in first-price auctions. In *AAAI/IAAI*, 2002, 373–378.
11. B. Lehmann, D. Lehmann and N. Nisan, Combinatorial auctions with decreasing marginal utilities. In *Proc 3rd ACM Conf. on Electronic Commerce (EC)*, 2001, 18–28.
12. J. K. Lenstra, D. B. Shmoys and E. Tardos, Approximation algorithms for scheduling unrelated parallel machines. In *Mathematical Programming 46*, 1990, 259–271.
13. A. Mu'alem and N. Nisan, Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI/IAAI* 2002, 379–384.
14. N. Nisan, Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC)*, 2000, 1–12.
15. M. Rothkopf, A. Pekec and R. Harstad, Computationally manageable combinatorial auctions. In *Management Science 44(8)*, 1998, 1131–1147.
16. T. Sandholm, An algorithm for optimal winner determination in combinatorial auctions. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1999, 542–547.
17. M. Tennenholtz, Tractable combinatorial auctions and b-matching. In *Artificial Intelligence 140(1/2)*, 2002, 231–243.