Santa Claus Meets Hypergraph Matchings

Arash Asadpour[1,⋆], Uriel Feige[2], and Amin Saberi[3]

[1] Stanford University, Stanford CA 94305, USA
asadpour@stanford.edu
[2] Weizmann Institute, Rehovot 76100, Israel
uriel.feige@weizmann.ac.il
[3] Stanford University, Stanford CA 94305, USA
saberi@stanford.edu

Abstract. We consider the problem of max-min fair allocation of indivisible goods. Our focus will be on the restricted version of the problem in which there are m items, each of which associated with a non-negative value. There are also n players and each player is only interested in some of the items. The goal is to distribute the items between the players such that the least happy person is as happy as possible, i.e. one wants to maximize the minimum of the sum of the values of the items given to any player. This problem is also known as *the Santa Claus problem* [3]. Feige [9] proves that the integrality gap of a certain configuration LP, described by Bansal and Sviridenko [3], is bounded from below by some (unspecified) constant. This gives an efficient way to estimate the optimum value of the problem within a constant factor. However, the proof in [9] is nonconstructive: it uses the Lovasz local lemma and does not provide a polynomial time algorithm for finding an allocation. In this paper, we take a different approach to this problem, based upon local search techniques for finding perfect matchings in certain classes of hypergraphs. As a result, we prove that the integrality gap of the configuration LP is bounded by $\frac{1}{5}$. Our proof is nonconstructive in the following sense: it does provide a local search algorithm which finds the corresponding allocation, but this algorithm is not known to converge to a local optimum in a polynomial number of steps.

1 Introduction

Resource allocation problems, i.e. allocating limited resources to a number of players while satisfying some given constraints, have been studied extensively in computer science, operations research, economics, and the mathematics literature. Depending on whether the resource is divisible or not one can distinguish two main types of such problems. The divisible case has been considered mostly by combinatorists and measure theorists in the past century under the title of "Cake Cutting" problems [16,5]. On the other hand, the indivisible resource allocation problems have been mostly the focus of algorithmic lines of research. In

⋆ The first and third authors were supported through NSF grants 0546889 and 0729586 and a gift from Google.

A. Goel et al. (Eds.): APPROX and RANDOM 2008, LNCS 5171, pp. 10–20, 2008.

such problems, often the set of resources R consists of m items. There is also a set P of n players. Each player i has a value function $f_i : 2^S \to \mathbb{R}$. For the sake of simplicity we define $v_{ij} = f_i(\{j\})$. The goal is to partition the set of items to subsets $S_1, S_2, \cdots, S_n$ and allocate each part to one of the players such that a certain objective function is optimized.

Depending on the objective functions, various indivisible resource allocation problems can be considered. For example, the problem of *maximizing social welfare* arises when we want to maximize $\sum_i f_i(S_i)$. See [6,8,10,17] for recent progress on this problem.

Minimizing the makespan is another example of indivisible resource allocation problems in which the goal is to minimize $\max_i f_i(S_i)$ and f_i's are linear functions, i.e. $f_i(S_i) = \sum_{j \in S_i} v_{ij}$. Lenstra, Shmoys and Tardos [15] provide a 2-approximation algorithm and also prove that the problem is hard to approximate within a factor of 1.5. Approximation ratios better than 2 are known for some very special cases of this problem [7].

Another interesting trend in indivisible resource allocation is *Max-min fair allocation problems*. Here, we aim to maximize $\min_i f_i(S_i)$ while f_i's are still linear functions. Although very similar at the first glance, this problem has turned out to be fundamentally different from minimizing the makespan and the techniques that are known to be useful there fail to give non-trivial results here. Most notably, the assignment LP used in [15] yields an additive approximation of $\max_{ij} v_{ij}$ [4]. It can be used to find a solution of value at least $\mathrm{OPT} - \max_{ij} v_{ij}$, where OPT is the value of the optimal solution. Unfortunately, it offers no approximation guarantee in the most challenging cases of the problem when $\mathrm{OPT} \leq \max_{ij} v_{ij}$.

Bansal and Sviridenko [3] studied this problem under the name of *the Santa Claus problem*, where Santa wants to distribute some presents among some kids and his goal is to do this in such a way that the least happy kid is as happy as possible. They considered a certain type of linear programming relaxation of the problem (known as *configuration* LP that we will explain shortly), and showed that it can be used to find a solution with value at least OPT/n. They also showed that the integrality gap of this LP is no better than $O(1/\sqrt{n})$. Asadpour and Saberi [2] showed how to round the solution of configuration LP to get a solution with value at least $\Omega(\mathrm{OPT}/\sqrt{n}(\log n)^3)$.

Our focus here will be on a special case of the Max-min fair allocation problem, known as *restricted assignment problem*, in which each item j has an inherent value v_j and a set of players to which the item can be assigned. In other words, for each such player i, the value of v_{ij} is v_j and for all other players it is 0. Bezakova and Dani [4] showed that this problem is hard to approximate within a factor better than $\frac{1}{2}$. (In fact, this is also the best hardness result known for the general problem.) Bansal and Sviridenko [3] showed that it is possible to round the values of the configuration LP and get a feasible solution with value $\Omega(\mathrm{OPT} \log\log\log n / \log\log n)$. Recently, Feige [9] proved that the optimal value of the configuration LP is in fact within a constant factor of OPT. Although [9] does not give a polynomial time algorithm to find a constant factor

approximation solution, it does provide a constant factor estimation for the optimal value of the problem[1]. This is due to the fact that the configuration LP can be solved (up to arbitrary precision) in polynomial time, and its value is an upper bound on OPT. The main result of this paper can be summarized as the following:

Theorem 1. *In the restricted assignment problem, there is a polynomial time algorithm that estimates the optimal value of max-min allocation problem within a factor of* $\frac{1}{5} - \epsilon$*, where* $\epsilon > 0$ *is an arbitrarily small constant.*

The polynomial time algorithm referred to in the above theorem is simply the configuration LP. The proof of the $\frac{1}{5}$ estimation factor will follow from our proof that the optimal value of the configuration LP is at most 5OPT. There is a small loss of ϵ in the estimation factor because the known polynomial time algorithms [3] solve the configuration LP up to any desired degree of accuracy, but not necessarily exactly.

Our proof of Theorem 1 transforms the problem into a problem of finding a perfect matching in certain hypergraphs. We design a local search algorithm that finds such a perfect matching. It is inspired by the techniques of [11] which will be discussed in Sect.2. This method can be viewed as a generalization of *Hungarian method* [14] to the domain of hypergraphs.

Comparing our results to those in [9], our result has the advantage of providing an explicit bound (of $\frac{1}{5}$) on the integrality gap of the configuration LP. Also, our proof technique suggests an algorithmic approach to round the solution of the configuration LP. While in [9] multiple applications of the Lovasz local lemma are used, here we introduce a local search algorithm and prove that it ends up in a solution with value at least $\frac{\text{OPT}}{5}$. Although we cannot bound the running time within a polynomial, it puts the problem in the complexity class PLS[2] and proposes the open question of whether this local search (or a modified version of it) converges in polynomial time to an appropriate solution.

1.1 The Configuration LP

Fix a real number t and suppose that we want to see if it is possible to do the allocation in such a way that each player i receives a bundle of items S_i with $f_i(S_i) \geq t$. For any bundle S of items, let x_{iS} be the indicator 0/1 variable, representing if the whole bundle S is allocated to person i (in this case x_{iS} will be 1) or not ($x_{iS} = 0$). To provide a bundle with value at least t for every person, we need to solve the following integer program:

[1] We emphasize that all the results related to the hardness of approximation remains valid even for estimating the optimal value of the problem.

[2] The complexity class PLS consists of problems for which, given any input instance there exists a finite set of solutions and an efficient algorithm to compute a cost for each solution, and also a neighboring solution of lower cost provided that one exists. Then the problem is to find a solution, namely a *local optimum*, that has cost less than or equal to all its neighbors. For more information, see [12].

- Every player only accepts bundles with value at least t; $\forall i : x_{iS} = 0$ whenever $f_i(S) < t$.
- Every player receives one bundle; $\forall i : \sum_S x_{iS} = 1$.
- Every item is allocated to at most one player: $\forall j : \sum_{i,S|j\in S} x_{iS} \leq 1$.
- $x_{iS} \in \{0, 1\}$ for every player i and bundle S.

The configuration LP is the relaxation of the above integer program. The last constraint is replaced by $x_{iS} \geq 0$

If the LP is feasible for some t_0, then it is also feasible for all $t \leq t_0$. Let optLP be the maximum of all such values of t (it can be shown that such maximum exists). Every feasible allocation is a feasible solution of configuration LP. Hence optLP $\geq$ OPT. The value of optLP and a feasible solution to the configuration LP of value optLP can be approximated within any desired degree of accuracy in polynomial time, as shown in [3].

In this paper we show that any fractional solution of configuration LP corresponding to optLP can be rounded (though not necessarily in polynomial time) to an integral solution whose value is within a constant factor of optLP. We provide two versions of our proof. In Section 2 we show how this result can be deduced by combining (in a blackbox manner) a previous intermediate result of Bansel and Sviridenko [3] with a theorem of Haxell [11]. In Section 3 we provide our main result which is basically a local search that finds an integral solution with value at least $\frac{\text{optLP}}{5}$. The proof in Section 3 is inspired by the results of Section 2, but is presented in a self contained way. It circumvents the use of the intermediate result of [3], and extends the proof technique of [11] in certain ways. Any of the two sections 2 and 3 can be read and understood without needing to read the other section.

2 Matchings in Hypergraphs

Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. A *matching* in $\mathcal{H}$ is a set of pairwise disjoint edges. We denote by $\nu(\mathcal{H})$ the maximum size of a matching in $\mathcal{H}$. A matching is called perfect if any vertex appears in exactly one of its edges. Unlike the case for matchings in graphs, the problem of finding a perfect matching in hypergraphs is NP-complete. (A well known special case of this problem is the NP-hard problem of 3-dimensional matching. Note that 3-dimensional matching can also be cast as a special case of finding a perfect matching in a bipartite hypergraph, a problem that we shall describe below.) There are some sufficient conditions known for the existence of perfect matchings in hypergraphs. See for example [1] and [13]. Some of these sufficient conditions are not computable in polynomial time.

Here, we focus on the problem of finding a maximum matching in *bipartite* hypergraphs. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called *bipartite* if the ground set V is the disjoint of sets U and V, and every $E \in \mathcal{E}$ satisfies $|E \cap U| = 1$. A perfect matching in a bipartite hypergraph is defined as a matching that saturates all the vertices in U. A *transversal* for hypergraph $\mathcal{H}$ is a subset $T \subseteq V$ with the property that $E \cap T \neq \emptyset$ for every $E \in \mathcal{E}$. Let $\tau(\mathcal{H})$ denote the minimum size

of a transversal of $\mathcal{H}$. For a subset $C \subseteq U$, we write $\mathcal{E}_C = \{F \subseteq V : \{c\} \cup F \in \mathcal{E} \text{ for some } c \in C\}$, and let $\mathcal{H}_C$ be the hypergraph $(V, \mathcal{E}_C)$. The following theorem is proved by Haxell in [11].

Theorem 2. *(**Haxell [11]**) Let $\mathcal{H} = (U \cup V, \mathcal{E})$ be a bipartite hypergraph such that for every $E \in \mathcal{E}$ we have $|E \cap V| \leq r-1$, and also $\tau(\mathcal{H}_C) > (2r-3)(|C|-1)$ for every $C \subseteq U$. Then $\nu(\mathcal{H}) = |U|$.*

When $r = 2$, $\mathcal{H}$ becomes a graph, and Haxell's theorem reduces to Hall's theorem.

The proof of Theorem 2 as described in [11] is not constructive.

2.1 A Constant Integrality Gap

In this section, we will consider a combinatorial conjecture (which is by now a theorem, by the results of [9]) which is equivalent up to constant factors to the restricted assignment problem, and prove it via Theorem 2. It reveals the intuition behind the relation between the restricted assignment problem and matchings in hypergraphs. Also, it is through this transformation that our local search appears.

Bansal and Sviridenko proved that if the following conjecture is true for some β, then it can be shown that the integrality gap of configuration LP relaxation for the restricted assignment problem is $\Omega(\beta)$.

> **Conjecture (by Bansal and Sviridenko [3]):** There is some universal constant $\beta > 0$ such that the following holds. Let $C_1, \cdots, C_p$ be collections of sets, $C_i = \{S_{i1}, \cdots, S_{il}\}$ for $i = 1, \cdots, p$, where each set S_{ij} is a k-element subset of some ground set and each element appears in at most l sets S_{ij}. Then there is a choice of some set $S_{i,f(i)} \in C_i$ for each $i = 1, \cdots, p$, and a choice $S'_i \subseteq S_{i,f(i)}$ with the property that $|S'_i| \geq \beta k$ and that each element occurs in at most one set in $\{S'_1, \cdots, S'_p\}$.

For every value k, it is not hard to see that the conjecture is true when $\beta = 1/k$. Feige [9] shows that the conjecture is true for some small enough universal constant β, for all values of k. Here, using Theorem 2 we prove that it is true even for $\beta = \frac{1}{5}$. (For every $k \geq 3$, our value of β is the largest number satisfying two constraints. Namely, that $(1-\beta) \geq 2\beta$, which will be needed in the proof of Theorem 3, and that βk is an integer. Hence , $\beta = 1/3$ when k is divisible by 3, but might be as small as $\frac{1}{5}$ for $k = 5$.)

Theorem 3. *Conjecture 2.1 is true for any $\beta \leq \frac{\lfloor k/3 \rfloor}{k}$*

Proof. Consider the following bipartite hypergraph $\mathcal{H} = (U \cup V, \mathcal{E})$. Here, $V = \bigcup_{i,j} S_{i,j}$ and $U = \{a_1, a_2, \cdots, a_p\}$. Also $\mathcal{E} = \{S \cup \{a_i\} : S \subseteq S_{i,j} \text{ for some } j, |S| = \beta k\}$. Note that here $r = \beta k + 1$. By the construction of $\mathcal{H}$, it is enough to prove that $\mathcal{H}$ has a perfect matching (i.e. a matching with size $|U|$). We will do so by showing that $\mathcal{H}$ satisfies the conditions of Theorem 2.

Consider an arbitrary $C \subset U$ and a transversal set T in $\mathcal{H}_C$. Because T is a transversal set in H_C, it must have some intersection with all the edges in H_C.

But edges in H_C correspond to all subsets S of V with βk elements such that for some j and $a_i \in C$ it holds that $S \subseteq S_{i,j}$. It means that for any such i and j, at least $(1-\beta)k$ elements of $S_{i,j}$ should be in T. (In fact, the number of elements of $S_{i,j}$ in T should be at least $(1-\beta)k+1$, but the extra $+1$ term does not appear to have a significant effect on the rest of the proof, so we omit it.)

Now, consider a bipartite graph $G = (V', E)$ such that $V' = U' \cup T$ where $U' = \bigcup_{i \in C}\{a_{i,1}, \cdots, a_{i,l}\}$ and $E = \{\{a_{i,j}, q\} : q \in S_{i,j}\}$. By the above discussion, $\deg(v) \geq (1-\beta)k$, for all $v \in U'$. Hence, $|E| \geq (1-\beta)k|C|l$. Also by the assumption of the conjecture, $\deg(v) \leq l$ for all $v \in V'$. Hence $|E| \leq l|T|$. Therefore,

$$l|T| \geq (1-\beta)k|C|l.$$

Thus, $|T| \geq (1-\beta)k|C| = \frac{1-\beta}{\beta}(r-1)|C|$. Picking any $\beta \leq 1/3$, we have $|T| \geq 2(r-1)|C|$ which means that $\tau(\mathcal{H}_C) > (2r-3)(|C|-1)$ for every $C \subseteq U$. This completes the proof. □

3 A $\frac{1}{5}$-approximate Solution through a Local Search

In this section we prove that the integrality gap of the configuration LP is no worse than $\frac{1}{5}$.

Given a feasible solution $\{x_{iS}\}$ to the configuration LP, we modify it as follows. To simplify notation, scale values of all items so that we can assume that $t = 1$. Recall that $v_{ij} \in \{0, v_j\}$. Call an item j *fat* if $v_j > \frac{1}{5}$ and *thin* it $v_j \leq \frac{1}{5}$. (The value of $\frac{1}{5}$ is taken with hindsight, being the largest value p satisfying $2(p+p) \leq 1-p$, needed later in the proof of Lemma 1.) For every fat item j, change v_j so that $v_j = 1$. Now modify the LP solution so as to make it *minimal*, by restricting players to choose bundles that are minimally satisfying for the player – dropping any item from the set reduces its value below 1. This can be achieved in polynomial time by dropping items from sets whenever possible. We are now left with an LP solution that uses only two types of sets:

- *Fat sets.* These are sets that contain only a single fat item and nothing else.
- *Thin sets.* These are sets that contain only thin items.

We call such a solution to the LP a *minimal solution.*

Construct a bipartite hypergraph based on the modified LP solution. The U side are the players. The V side are the items. For every player i put in hyperedges associated with those sets for which $x_{iS} > 0$ as follows. If $S = \{j\}$ is a fat set, include the hyperedge $\{i, j\}$. If S is a thin set, then for every *minimal* subset $S' \subset S$ of value at least $\frac{1}{5}$ (minimal in the sense that dropping any item from S' reduces its value below $\frac{1}{5}$), put in the hyperedge $\{i, S'\}$. Observe that by minimality, S' has weight at most $\frac{2}{5}$.

Theorem 4. *Given any minimal solution to the configuration LP, the bipartite hypergraph constructed above has a perfect matching (namely, a matching in which all vertices of U are covered).*

We note that Theorem 4 implies that there is an integer solution of value at least $\frac{1}{5}$, since every player can get either a fat set (that contains an item of value more than $\frac{1}{5}$), or a part of a thin set of value at least $\frac{1}{5}$.

Our proof of Theorem 4 is patterned a proof of [11], with some changes. The most significant of these changes is the use of Lemma 1.

For a set W of edges, we use the notation W_U to denote the vertices of U that are covered by W, and W_V to denote the vertices of V that are covered by W.

Proof. The proof is by induction on U. For $|U| = 1$, the theorem is obviously true (since the hypergraph has at least one edge). Hence assume that the theorem is true for $|U| = k$, and prove for $|U| = k + 1$.

Denote the vertices of U by $\{u_0, \ldots u_k\}$. By the inductive hypothesis, there is a matching of size k involving all U vertices except for u_0. (This is true because by removing u_0 from the hypergraph and all its edges, one obtains a hypergraph which corresponds to a minimal solution to an LP with one less player.) Pick an arbitrary such matching M. We present an algorithm that transforms this matching into a new matching of size $k + 1$. The algorithm is in some respects similar to the known algorithm for constructing matchings in bipartite graphs. It constructs an alternating tree in an attempt to find an alternating path. In the graph case, when such a path is found, the matching can be extended. In the hypergraph case, the situation is more complicated, and hence the proof will not provide a polynomial upper bound on the number of steps required until eventually the matching is extended.

In our alternating tree, there will be two types of edges. Edges of type A are edges that we try to add to the matching (A stands for *Add*). Edges of type B will be existing matching edges (hence $B \subset M$) that intersect edges of type A, and hence block us from adding edges of type A to the matching (B stands for *Block*). Every root to leaf path will be an alternating sequence of edges of type A and B.

The A edges will be numbered in the order in which they are added to the alternating tree. Hence their names will be $a_1, a_2, \ldots$, and these names are relative to a currently existing alternating tree (rather than being names that edges keep throughout the execution of the algorithm). For every $i \geq 1$, we associate with edge a_i an integer $m_i \geq 1$ that will correspond to the number of B edges that block a_i. The strict positivity of m_i implies that $|B| \geq |A|$.

Initially one needs to pick the first edge for the alternating tree. Pick an arbitrary edge e such that $e_U = u_0$. Let m_1 denote the number of edges from M that e_V intersects. If $m_1 = 0$, then *terminate*, because the edge e can be added to M, obtaining a perfect matching. If $m_1 > 0$, rename e as a_1, add a_1 to A, and add the m_1 matching edges that intersect a_1 to B.

Let $i \geq 2$ and assume that the alternating tree already contains $i - 1$ edges of type A (named as $a_1, \ldots, a_{i-1}$), and at least $i - 1$ edges of type B. We now pick an edge e such that $e_U \in (A \cup B)_U$ and e_V does not intersect $(A \cup B)_V$. The following lemma shows that such an edge must exist.

Lemma 1. *Let $H(U, V, E)$ be the hypergraph associated with a minimal solution to the configuration LP. Then given any alternating tree as described above, there always is an edge e such that $e_U \in (A \cup B)_U$ and e_V does not intersect $(A \cup B)_V$.*

Proof. Let ℓ denote the number of vertices of U in the alternating tree. Each hyperedge corresponds in a natural way either to a fat set or to (part of) a thin set. Let A_f (A_t, respectively) denote the collection of A edges in the alternating tree that correspond to fat sets (thin sets, respectively), and similarly for B_f and B_t with respect to B edges in the alternating tree. Observe that in an alternating tree necessarily $|A_f| + |A_t| = |A| < \ell$ and $|B_f| + |B_t| = |B| < \ell$. Moreover, $|A_f| = |B_f| = |(A_f \cup B_f)_V|$, because every fat edge of A contains exactly one vertex in V, this vertex is contained only in fat edges, and hence this fat edge is intersected by exactly one fat edge in B.

Consider now the restriction of the minimal solution to the LP to the set of players P represented by the ℓ vertices of $(A \cup B)_U$. Let S_f be the collection of fat sets and S_t be the collection of thin sets. Let $\alpha = \sum_{i \in P,\ S \in S_f} x_{iS}$ denote the total weight assigned by this restricted solution to fat sets, and let $\beta = \ell - \alpha = \sum_{i \in P,\ S \in S_t} x_{iS}$ denote the total weight assigned by this restricted solution to thin sets. If $\alpha > |(A_f \cup B_f)_V|$ then it must be the case that some fat set has positive weight in the restricted solution but is not part of the alternating tree. In this case, this fat set can contribute a hyperedge to the alternating tree. Hence it remains to deal with the case that $\alpha \leq |A_f|$. In this case, $2\beta \geq |A_t| + |B_t| + 2$. The hyperedges in the alternating tree that correspond to thin sets each take up value at most $\frac{2}{5}$. Hence even after removing all items appearing in the alternating tree, the sum of weights multiplied by respective remaining value of thin sets in the LP is

$$\sum_{i \in P,\ S \in S_t} x_{iS} \sum_{j \in S \setminus (A_t \cup B_t)} v_{ij} > \frac{\beta}{5}$$

This means than at least one thin set must have retained a value of at least $\frac{1}{5}$. Hence, this thin set can contribute a hyperedge to the alternating tree. □

Pick an arbitrary hyperedge e satisfying Lemma 1 and let m_i denote the number of edges of M that e intersects. If $m_i > 0$, we call this an *extension* (the alternating tree grew larger), rename e as a_i, add a_i to A, and add the m_i matching edges that intersect a_i to B.

We now describe what to do when $m_i = 0$. If $e_U = u_0$, add edge e to the matching M, and *terminate*. If $e_U \neq u_0$, then let e' be the unique edge in B for which $e_U = e'_U$. Let a_j (here necessarily we will have $j < i$) be the unique edge in A that intersects e'. In the matching M, replace the matching edge e' by the matching edge e. Note that this still gives a valid matching of size k, because by construction, e does not intersect any edge of M except for sharing its U side vertex with e', which is removed from M. Update m_j by decreasing it by 1. If the new value of m_j is still positive, this step ends. However, if $m_j = 0$, then the above procedure is repeated with j replacing i (in particular, a_j will also become part of the matching M). Because $j < i$, the number of times the procedure can

be repeated is finite, and hence eventually the step must end. We call such a step a *contraction* (the alternating tree becomes smaller).

This completes the description of the algorithm. Observe that the algorithm terminates only when we extend the matching M by one more edge. Hence it remains to show that the algorithm must terminate.

To see this, consider the evolution of vector $m_1, m_2, \ldots, m_j$. For simplicity of the argument, append at the end of each such vector a sufficiently large number ($|M|+1$ would suffice). We call the resulting vector the *signature* of the alternating tree. We claim that the signatures of any two alternating trees are distinct. This is because ordering the signatures by the time in which they were generated sorts them in decreasing lexicographic order. For extension steps, this follows from the fact that we appended $|M|+1$ at the end of the respective vector. For contraction steps, this follows from the fact that m_j decreases.

Since $\sum_i m_i \leq |M|$ and $m_i > 0$ (whenever m_i is defined), the number of possible signatures is $2^{|M|}$ (there is a one to one correspondence between these vectors and choices of after which items to place delimiters in a sequence of $|M|$ items), and hence the algorithm cannot have infinite executions. □

The proof of Theorem 4 implicitly provides a local search algorithm to find an integral solution with value $\frac{1}{5}$. Its basic objects are the alternating trees. A basic step is that of adding an edge to the tree, resulting in either an *extension* step or a *contraction* step. The measure of progress of the local search is via the lexicographic value of the corresponding signature. Given a matching with $|M| < n$ edges (an allocation to M players), it will be extended after at most $2^{|M|}$ steps. Hence starting with the empty matching it takes at most $\sum_{|M|=0}^{n-1} 2^{|M|} < 2^n$ local search steps uptil a perfect matching is found. This corresponds to allocating disjoint bundles of value at least optLP/5 to all players. Noting that optLP is at least as large as the optimal solution, the following theorem is established.

Theorem 5. *After 2^n local moves, our algorithm finds a feasible integral $\frac{1}{5}$-approximate allocation.*

4 Open Directions

Characterizing the best possible approximation ratio for the max-min allocation problem is still open, both for the restricted assignment version and for the general version of the problem. We list here some research questions that are suggested by our work.

1. *Integrality gap.* We showed that the integrality gap of the configuration LP for the restricted assignment problem is no worse than 1/5. It was previously known to be no better than 1/2 (in particular, this follows from the NP-hardness result of [4]). Narrow the gap between these two bounds.
2. *Complexity of local search.* Our proof is based on a local search procedure. Can a locally optimal solution with respect to this local search be found in polynomial time? Is finding such a solution PLS-complete? These questions

apply also to similar local search procedures that find a perfect matching in hypergraphs satisfying the conditions of Theorem 2.

3. *Approximation algorithms.* Provide an approximation algorithm (that actually finds an allocation) with a constant approximation ratio for the restricted assignment problem.
4. *Hypergraph matchings.* Can the proof techniques used in our paper be used also for other problems? For example, can our approach be employed to prove that the integrality gap of configuration LP for general max-min fair allocation problem is $\Theta(\frac{1}{\sqrt{n}})$ (saving a $\log^3 n$ factor compared to [2])?

Acknowledgements

Part of this work was performed at Microsoft Research, Redmond, Washington.

References

1. Aharoni, R., Haxell, P.: Hall's theorem for hypergraphs. Journal of Graph Theory 35, 83–88 (2000)
2. Asadpour, A., Saberi, A.: An Approximation Algorithm for Max-Min Fair Allocation of Indivisible Goods. In: Proceedings of the ACM Symposium on Theory of Computing (STOC) (2007)
3. Bansal, N., Sviridenko, M.: The Santa Claus problem. In: Proceedings of the ACM Symposium on Theory of Computing (STOC) (2006)
4. Bezakova, I., Dani, V.: Allocating indivisible goods. SIGecom Exchanges (2005)
5. Brams, S.J., Taylor, A.D.: Fair division: from Cake Cutting to Dispute Resolution. Cambridge University Press, Cambridge (1996)
6. Dobzinski, S., Schapira, M.: An improved approximation algorithm for combinatorial auctions with submodular bidders. In: Proceedings of Symposium on Discrete Algorithms (SODA) (2006)
7. Ebenlendr, T., Krcal, M., Sgall, J.: Graph Balancing: A Special Case of Scheduling Unrelated Parallel Machines. In: Proceedings of Symposium on Discrete Algorithms (SODA) (2008)
8. Feige, U.: On maximizing welfare when utility functions are subadditive. In: Proceedings of the ACM Symposium on Theory of Computing (STOC) (2006)
9. Feige, U.: On allocations that maximize fairness. In: Proceedings of Symposium on Discrete Algorithms (SODA) (2008)
10. Feige, U., Vondrak, J.: Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In: Proceedings of Foundations of Computer Science (FOCS) (2006)
11. Haxell, P.E.: A Condition for Matchability in Hypergraphs. Graphs and Combinatorics 11, 245–248 (1995)
12. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? Journal of Computer and System Sciences 37, 79–100 (1988)
13. Kessler, O.: Matchings in Hypergraphs. D.Sc. Thesis, Technion (1989)
14. Kuhn, H.W.: The Hungarian Method for the assignment problem. Naval Research Logistic Quarterly 2, 83–97 (1955)

15. Lenstra, J.K., Shmoys, D.B., Tardos, E.: Approximation algorithms for scheduling unrelated parallel machines. Mathematical Programming, Series A (1993)
16. Steinhaus, H.: The problem of fair division. Econometrica (1948)
17. Vondrak, J.: Optimal approximation for the Submodular Welfare Problem in the value oracle model. In: Proceedings of the ACM Symposium on Theory of Computing (STOC) (2008)