SAND2012-3502C

Approximation algorithms for generalized hypergraph matching problems

Ojas Parekh[1] and David Pritchard[2]

[1] Sandia National Laboratories⋆, MS 1316, Albuquerque NM 87185, USA
[2] University of Waterloo

Abstract. In this paper we examine two generalizations of the matching problem on hypergraphs: given a collection of sets of size k, we must select a maximum-profit subcollection subject to certain packing constraints. Our first result is a $k-1+1/k$-approximation for b-matching, where the packing constraints are just degree upper bounds at each vertex; our main technique is a new degree-balanced variant of iterated packing methods. An application of this setting is a truthful mechanism for auctions where each bidder can win a bounded number of items. Our second result is for k-hypergraph demand matching, where edges have demand values and each vertex has an upper bound on the demand of selected incident edges. Previously a 2k-approximation was known via iterated packing, and we show that a much simpler local ratio algorithm gives the same result.

1 Introduction

Matching problems laid the foundation of combinatorial optimization over 40 years ago. Since then matching problems have been generalized in several directions; one such natural and classical direction is the notion of matching in hypergraphs. The matching problem in hypergraphs seeks to find a maximum size or weight collection of hyperedges such that each vertex may have at most one hyperedge incident upon it. In particular our focus is on *k-uniform* instances: those in which each edge has exactly k vertices. This problem is also known as the k-set packing problem and has been extensively studied, from the perspective of both combinatorial optimization and combinatorics (e.g. see Chan and Lau [4]).

Our focus is on generalizations of hypergraph matching. We consider the case where each vertex v has a capacity b_v, and we allow up to b_v hyperedges incident upon v. This generalizes the well-known b-matching problem in graphs. We also consider the further generalization in which each hyperedge is endowed with a demand, which leads to a common generalization of hypergraph matching and the fundamental knapsack problem.

⋆ Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Problem definition. We consider specializations of the *hypergraph demand matching* (HDM) problem. More formally, given a weighted hypergraph $H = (V, E)$ endowed with a demand $d_e \in \mathbb{Z}_+$ for each hyperedge $e \in E$ and a capacity $b_v \in \mathbb{Z}_+$ for $v \in V$, the problem may be defined by the following integer program:

$$\begin{aligned} \text{Maximize} \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{e | v \in e} d_e x_e \leq b_v \qquad \forall v \in V \\ & \qquad\quad x_e \in \{0, 1\} \quad \forall e \in E. \end{aligned}$$

We will use the natural LP relaxation of the above, obtained by replacing the $x_e \in \{0, 1\}$ with $0 \leq x_e \leq 1$. We make the assumption that for each hyperedge e, $d_e \leq b_v$ for all $v \in e$. This so-called *no-clipping* assumption is easy to satisfy by deleting edges that violate it; however, this assumption is necessary in order for the natural LP relaxation to have a bounded integrality gap. We note that the restriction that $x_e \in \{0, 1\}$ is for the sake of exposition and that our results may be extended to apply to multiple copies of edges.

A *k-hypergraph* is one in which $|e| \leq k$ for each $e \in E$, and a *k-uniform hypergraph* has $|e| = k$ for each $e \in E$. The first problem we study is the b-matching problem on k-uniform hypergraphs, which is obtained when $d_e = 1$ for every hyperedge. This is a direct generalization of the well-known b-matching problem in graphs (i.e. $k = 2$). We will also consider the HDM problem on k-hypergraphs, which we call *k-hypergraph demand matching*.

We use mostly standard notation. For a set $S \subseteq E$ and a vector $v \in \mathbb{R}^E$, we use the notation $v(S)$ to refer to $\sum_{e \in S} v_e$. For a singleton set $\{e\}$ we let $v(e) = v(\{e\}) = v_e$. We will use the terms edge and hyperedge interchangeably.

Our technique. We employ a technique called iterative packing which was recently used by Parekh [14] to obtain approximation algorithms for k-hypergraph demand matching problems. Given a fractional solution, iterative packing iteratively constructs an approximate convex decomposition of the fractional solution by greedily packing each edge into a requisite fraction of integer solutions. Pseudo-greedy methods similar to iterative packing have been successfully applied to packing and coloring problems. Chekuri, Mydlarz, and Shepherd [5] used such a technique to obtain a 4-approximation for multicommodity flows on trees. Bar-Yehuda et al. [2] gave both an iterative packing like and primal-dual algorithms for approximating independent sets in t-interval graphs. Feige and Singh [6] applied this type of technique for weighted edge coloring of bipartite graphs.

Iterative packing can be seen as an extension and unification of the above type of techniques into a single framework. Moreover, akin to the iterative rounding method for covering problems, iterative packing explicitly identifies elements with large fractional values to obtain better approximation ratios. Other aspects of the framework include leveraging a specific ordering the elements and starting with a nontrivial convex decomposition. This combination of ideas allows iterative packing to obtain approximation ratios approaching the integrality gap of the underlying LP formulation. Iterative packing yields a $2k$-approximation for k-hypergraph demand matching [14], nearly matching the best known lower bound of $2k - 1$ [1] on the integrality gap of the natural LP relaxation. For the special case of demand matching (i.e. $k = 2$), iterative packing was able to resolve the integrality gap at 3 [14].

Our results. In the present work we extend iterative packing to give a $(k - 1 + 1/k)$-approximation for the b-matching problem in k-uniform hypergraphs, which settles the integrality gap of the natural LP relaxation; our technique also yields an improved $(k-1)$-approximation on k-partite instances. Our main idea is explicitly specifying excluded solutions when elements are packed; previously only solutions that would be rendered infeasible were excluded. This allows iterative packing to retain a simple greedy flavor while providing more control over resulting approximate convex decomposition.

We also show that the iterative packing $2k$-approximation for k-hypergraph demand matching from [14] can be reinterpreted as a primal-dual algorithm. This significantly simplifies the algorithm and vastly improves its running time by avoiding solving the LP relaxation and maintaining an approximate convex decomposition.

Related work. Matching problems in k-uniform hypergraphs are well-studied. A celebrated result of Füredi, Kahn, and Seymour [8] established the integrality gap of the natural relaxation at $k-1+1/k$ for hypergraph matching in k-uniform hypergraphs, with an improvement to $k-1$ for k-partite k-uniform hypergraphs. On the algorithmic side, the problem is NP-hard for $k \geq 3$, and for any fixed $\varepsilon > 0$ the best known approximation ratios are $(\frac{k}{2} + \varepsilon)$ for the unweighted version by Hurkens and Schrijver [10] and $(\frac{k+1}{2} + \varepsilon)$ for the weighted version by Berman [3]. On the other hand, Hazan, Safra and Schwartz [9] showed that the problem is hard to approximate within a factor of $\Omega(\frac{k}{\log k})$ unless $\mathsf{P} = \mathsf{NP}$.

The above algorithms are based on local search and do not provide a bound on the integrality gap of the natural relaxation. Chan and Lau [4] recently gave a $(k-1+1/k)$-approximation based on the fractional local ratio method, which matches the integrality gap of the natural relaxation and gives an improvement for $k = 3$. They also give linear and semidefinite formulations with an improved gap.

Unfortunately, none of the above results extend to b-matching in k-uniform hypergraphs. For this problem fewer approximation results are known. Most relevant to our work are a greedy $k + 1$-approximation by Krysta [12] and a primal-dual k-approximation by Young and Koufogiannakis [11]. An algorithm with guarantee $(\frac{k+3}{2} + \varepsilon)$ is implicit in the recent work of Feldman et al. [7] and Ward [15] on k-exchange systems; however, its running time has a pseudo-polynomial dependence on b; even so, our result provides a better approximation for $k \leq 4$.

For k-hypergraph demand matching $2k$ is the best known approximation guarantee [14]; however, Bansal et al. [1] devised a deterministic $8k$-approximation and randomized $(ek+o(k))$-approximation for the more general problem of approximating k-column-sparse packing integer programs.

2 Iterative packing for b-Matching on k-uniform hypergraphs

We begin by describing the basic version of iterative packing. Given a feasible fractional solution x for the natural LP relaxation, the iterative packing produces an α-*approximate* convex decomposition of x:

$$\alpha \cdot x = \sum_{i \in I} \lambda_i \chi^i \ , \tag{1}$$

for some $\alpha \in (0,1]$, where each χ^i is a feasible integral solution (and $\sum_i \lambda_i = 1$; $\lambda_i \geq 0$ for all i).

Iterative packing typically starts with the empty hypergraph on V, for which $\chi^1 = 0$, $\lambda_1 = 1$ is a trivial decomposition of the LP solution $x = 0$. Edges are iteratively packed according to an ordering $e_1, e_2, \dots e_m$, where $m = |E|$. At iteration j, the edge $e = e_j$ is greedily packed into any solution χ^i which feasibly accommodates it until e has been packed into an $\alpha \cdot x_e$ fraction of the solutions. We will see that this requires increasing the number of solutions by at most one for each iteration. Thus iterative packing produces a sparse decomposition, namely one with $|I| \leq m+1$. This property will not directly hold for the efined version of iterative packing we will use; however, one may apply elementary linear algebra to retain at most $m+1$ solutions. A procedure to accomplish the latter is related to Carathéodory's Theorem; we leave this as an exercise for the reader.

The construction of the decomposition (1) implies that one can find an integral solution with weight at least $\alpha \cdot (w \cdot x^*)$; the resulting approximation guarantee is $\rho \leq \frac{1}{\alpha}$. A nice feature is that the decomposition gives us a weight-oblivious representation of an approximate solution.

Proposition 1. *Iterative packing is a* $\frac{1}{\alpha}$*-approximation.*

Here we briefly outline an argument using iterative packing that yields a $(k-1+1/k)$-approximation for b-matching on k-uniform hypergraphs; this result establishes the integrality gap of the natural linear programming relaxation at $k-1+1/k$ (e.g. Chan and Lau [4] for the corresponding lower bound). An interesting feature of the approach is that it combines a slightly refined iterative packing approach whose edge ordering is obtained from the following lemma, which may be derived from the proof of Lemma 2.3 in Chan and Lau [4].

Lemma 1. *(Chan and Lau [4]) For any extreme point x of the natural relaxation of the k-hypergraph b-matching problem, there exists an ordering of the edges, $e_1, e_2, \dots, e_m$ such that for each i, there exists a vertex $v_i \in e_i$ with:*

$$x_{e_i} \geq \frac{x(\delta(v_i) \cap E_i)}{k},$$

where $E_i = \{e_1, e_2, \ldots, e_i\}$.

This lemma is proven by using linear-algebraic extreme point arguments to show that a vertex of degree at most k exists. Although the lemma is proven for the case of 1-matching in k-uniform hypergraphs by Chan and Lau, their argument readily extends to our context of b-matching. We note that for convenience, the ordering specified above is the opposite of the ordering as specified in their paper. Unfortunately their fractional local ratio algorithm does not extend to hypergraph b-matching; however, we show that iterative packing is able to produce an approximation algorithm matching the worst-case integrality gap. We do require refinements over the standard iterative packing approach in order to obtain our bound.

Algorithm 1 Iterative packing iteration

Require: A collection $\{(\mu_i, \chi^i)\}_{1 \leq i \leq l}$ of multipliers and solutions with
- each $\mu_i > 0$ and $\sum_i \mu_i = 1$,
- each χ^i a feasible integer solution not containing e, and
- a set of indices of *excluded solutions*, X_u for each $u \in e$.

1: ▷ (I_e is the index set of solutions χ^i which contain e)
2: $I_e \leftarrow \emptyset$, $i \leftarrow 1$
3: **while** $\lambda(I_e) < \alpha \cdot x_e$ and $i \leq l$ **do**
4:     $\lambda_i \leftarrow \mu_i$
5:     **if** $i \notin \bigcup_{u \in e} X_u$ **then**
6:         **if** $\lambda_i > \alpha \cdot x_e - \lambda(I_e)$ **then**
7:             $l \leftarrow l + 1$
8:             ▷ (Add a new copy of the solution χ^i)
9:             $\chi^l \leftarrow \chi^i$
10:             ▷ (λ_l and λ_i are set so that $\mu_i = \lambda_l + \lambda_i$ and $\lambda_l, \lambda_i > 0$)
11:             $\lambda_l \leftarrow \mu_i - (\alpha \cdot x_e - \lambda(I_e))$
12:             $\lambda_i \leftarrow \alpha \cdot x_e - \lambda(I_e)$
13:         Pack e into χ^i
14:         $I_e \leftarrow I_e \cup \{i\}$
15:     $i \leftarrow i + 1$

Iterative packing for hypergraph b***-matching.*** Algorithm 1 captures an iteration of iterative packing, in which we seek to insert the hyperedge $e = e_j$ into an existing approximate convex combination of solutions over the edges $e_1, \ldots, e_{j-1}$. The iteration *succeeds* if the algorithm terminates with $\lambda(I_e) = \alpha \cdot x_e$. Note that the algorithm will never terminate with $\lambda(I_e) > \alpha \cdot x_e$, since in order to do so the conditional on Line 6 must be taken, which ensures that $\lambda(I_e) = \alpha \cdot x_e$ and that the algorithm will terminate after the current iteration. If we start with the collection $\{(\lambda_1 = 1, \chi^1 = \emptyset)\}$, and iterative packing suceeds at every iteration, then it terminates with a convex decomposition $\alpha \cdot x = \sum_{i \leq l} \lambda_i \chi^i$.

For the hyperedge e being packed, the sets X_u define the *excluded solutions* at the vertex u with respect to e. Let $S_u = \{i \mid \chi^i(\delta(u)) = b_u\}$, which is simply the set of solutions that are saturated at u. If we were to set $X_u = S_u$ for $u \in e$, iterative packing would produce a collection of feasible solutions since the above algorithm would only pack e into solutions with $\chi^i(\delta(u)) \leq b_u - 1$ for all $u \in e$. Taking $X_u = S_u$ is the standard approach with iterative packing; however, for the case of k-hypergraph b-matching this yields a poor approximation guarantee. Surprisingly this is true of the standard b-matching problem (i.e. $k = 2$) as well. Our main contribution is giving sets $X_u \supseteq S_u$ which allow us to set $\alpha \geq \frac{1}{k-1+1/k}$, establishing the integrality gap of the standard relaxation.

We define the X_u we will use in the process of proving the following lemma. For the remainder of this section we assume that α is chosen so that iterative packing succeeds. We will bound α and consequently derive our approximation guarantee in the next section.

Lemma 2. *Given a fractional solution x, our iterative packing algorithm produces an approximate convex decomposition into integer solutions χ^i,*

$$\alpha \cdot x = \sum_{i \in I} \lambda_i \chi^i,$$

which satisfies some additional properties. Let $\beta_u = \lceil x(\delta(u)) \rceil$ and $T_u = \{i \in I \mid \chi^i(\delta(u)) = \beta_u\}$ for each vertex $u \in V$. For each $u \in V$, the decomposition satisfies:

(i) $\chi^i(\delta(u)) \leq \beta_u$ (as opposed to $\chi^i(\delta(u)) \leq b_u$) for each solution χ^i, and
(ii) $\lambda(T_u) \leq \alpha \cdot (x(\delta(u)) - (\beta_u - 1))$, if $\beta_u \neq 0$.

Intuitively this type of condition ensures that the number of edges packed at a vertex u does not vary too much across the solutions χ^i; otherwise packing becomes more troublesome in future iterations. This type of condition is necessary to obtain an approximation guarantee matching the natural integrality gap even for the standard b-matching problem.

Proof. Suppose we are given a fractional solution x to k-hypergraph b-matching instance on a graph $G = (V, \{e_1, \ldots, e_m\})$, where the edge ordering is that in which iterative packing is performed. We appeal to induction and hypothesize the existence of an approximate convex decomposition as specified by Lemma 2 of the residual fractional solution $\hat{x}$ on the graph $G - \{e_m\}$:

$$\alpha \cdot \hat{x} = \sum_{i \in J} \hat{\lambda}_i \hat{\chi}^i.$$

Our goal is to show that we may pack $e = e_m$ into an $\alpha \cdot x_e$ fraction of these solutions to obtain a decomposition of x satisfying the conditions of Lemma 2. The base case of $x = \chi^1 = 0$ and $\lambda_1 = 1$ trivially satisfies these conditions for any α. Observe that the functions $x(\delta(\cdot))$ and $\hat{x}(\delta(\cdot))$ are identical on vertices outside of e. For each $u \in e$ we have $x(\delta(u)) = \hat{x}(\delta(u)) + x_e$, and we let $\hat{\beta}_u = \lceil \hat{x}(\delta(u)) \rceil$.

Suppose for some $u \in e$ we have $\beta_u = \hat{\beta}_u$. In order to satisfy the strengthened feasibility condition, (i) we may pack e only into solutions $\hat{\chi}^i$ with $\hat{\chi}^i(\delta(u)) \leq$

$\beta_u - 1$. Letting $\hat{T}_u = \{i \in J \mid \hat{\chi}^i(\delta(u)) = \hat{\beta}_u\}$, we take the excluded solution set, X_u to be precisely $\hat{T}_u$ in this case. If α is chosen so that we are able to pack e, we see that condition (ii) is easily satisfied:

$$\begin{aligned}\lambda(T_u) &\leq \alpha \cdot x_e + \hat{\lambda}(\hat{T}_u) \\ &\leq \alpha \cdot (x_e + \hat{x}(\delta(u)) - (\hat{\beta}_u - 1)) \quad \text{[by (ii) of inductive hypothesis]} \\ &= \alpha \cdot (x(\delta(u)) - (\beta_u - 1)).\end{aligned}$$

If $\beta_u \neq \hat{\beta}_u$ we must have $\beta_u = \hat{\beta}_u + 1$, since $0 < x_e \leq 1$. In this case property (i) is trivially satisfied by any successful packing of e since our inductive hypothesis gives us $\hat{\chi}^i(\delta(u)) \leq \beta_u - 1$ for all i. However we must be a bit more careful in order to satisfy (ii). The only way we can produce a solution with $\chi^i(\delta(u)) = \beta_u$ is by packing e into a solution with $\hat{\chi}^i(\delta(u)) = \hat{\beta}_u$. Thus if

$$\min\{\alpha \cdot x_e, \hat{\lambda}(\hat{T}_u)\} \leq \alpha \cdot (x(\delta(u)) - (\beta_u - 1)), \tag{2}$$

any choice of X_u satisfies (ii), and we set $X_u = \emptyset$. Note that this is particularly important for the case when $\hat{\beta}_u = 0$ and $\beta_u = 1$, for which (2)) holds (i.e. $\alpha \cdot x_e \leq \alpha \cdot x(\delta(u))$). The fact that there is constraint on X_u allows us to progress from the base case, whose inductive hypothesis gives us only trivial conditions (i) and (ii).

If (2) does not hold, we have $\hat{\lambda}(\hat{T}_u) > \alpha \cdot (x(\delta(u)) - (\beta_u - 1))$. In this case we select $X_u \subset \hat{T}_u$ such that $\hat{\lambda}(X_u) = \hat{\lambda}(\hat{T}_u) - \alpha \cdot (x(\delta(u)) - (\beta_u - 1))$, ensuring that (ii) is satisfied. Such an X_u may not exist; however, we may remedy this by creating a clone of at most one solution $\hat{\chi}^i$ and distributing the value of $\hat{\lambda}_i$ among the multipliers of the copies as on Lines 7–12 of the algorithm. Since we may have to perform such a cloning operation for each $u \in e$, packing e may create up to $k + 1$ new solutions rather than 1 as with standard iterative packing. □

In the sequel we will require bounds on $\hat{\lambda}(X_u)$, which we state now:

$$\begin{aligned}\hat{\beta}_u = 0,\, \beta_u = 1: \quad \hat{\lambda}(X_u) &= 0 = \alpha \cdot (\hat{x}(\delta(u)) - (\beta_u - 1)), \\ \hat{\beta}_u = \beta_u: \quad \hat{\lambda}(X_u) &= \hat{\lambda}(\hat{T}_u) \\ &\leq \alpha \cdot (\hat{x}(\delta(u)) - (\hat{\beta}_u - 1)) \\ &= \alpha \cdot (\hat{x}(\delta(u)) - (\beta_u - 1)), \text{ and} \\ 0 \neq \hat{\beta}_u \neq \beta_u: \quad \hat{\lambda}(X_u) &= \max\{\hat{\lambda}(\hat{T}_u) - \alpha \cdot (x(\delta(u)) - (\beta_u - 1)), 0\} \\ &\leq \alpha \cdot (\hat{x}(\delta(u)) - (\hat{\beta}_u - 1)) - \alpha \cdot (x(\delta(u)) - (\beta_u - 1)) \\ &= \alpha \cdot (1 - x_e).\end{aligned}$$

Selecting α**.** We are now in a position to derive a universal bound on α such that our iterative packing algorithm will succeed. We must ensure that each edge e is successfully packed satisfying the conditions of Lemma 2. In the proof of this lemma we provided requisite excluded solution sets X_u for each $u \in e$.

Algorithm 1 packs e at a fractional value of $\lambda_i \le \mu_i$ for each $i \notin \bigcup_{u\in e} X_u$. Thus Algorithm 1 succeeds precisely when $\alpha \cdot x_e + \mu(\bigcup_{u\in e} X_u) \le 1$. Since each $\mu_i \ge 0$, a union bound yields $\mu(\bigcup_{u\in e} X_u) \le \sum_{u\in e} \mu(X_u)$, hence we may select α such that

$$\alpha \cdot x_e + \sum_{u\in e} \mu(X_u) \le 1, \tag{3}$$

for the X_u as given in the proof of Lemma 2. The values $\mu(X_u)$ are equivalent to $\hat{\lambda}(X_u)$ in the parlance of the proof of Lemma 2, and we appeal to the previously derived bounds on $\hat{\lambda}(X_u)$. Let us partition the hyperedge e into sets of vertices $e' = \{u \in e \mid \hat{\beta}_u = \beta_u \text{ or } \hat{\beta}_u = 0,\ \beta_u = 1\}$ and $e'' = e - e'$. In particular we have

$$\begin{aligned}
\alpha \cdot x_e + \sum_{u\in e} \mu(X_u) &\le \alpha \cdot x_e + \alpha \cdot \sum_{u\in e'} (\hat{x}(\delta(u)) - (\beta_u - 1)) + \alpha \cdot \sum_{u\in e''} (1 - x_e) \\
&= \alpha \cdot \left(x_e + \sum_{u\in e'} (x(\delta(u)) - x_e - (\beta_u - 1)) + \sum_{u\in e''} (1 - x_e) \right) \\
&= \alpha \cdot \left((1 - |e|) \cdot x_e + |e''| + \sum_{u\in e'} (x(\delta(u)) - (\beta_u - 1)) \right).
\end{aligned}$$

Thus we may satisfy (3) by selecting α so the last quantity above is 1. Recall that $\rho \le 1/\alpha$ (Proposition 1), where ρ is our approximation guarantee, hence

$$\rho \le (1 - k) \cdot x_e + |e''| + \sum_{u\in e'} (x(\delta(u)) - (\beta_u - 1)), \tag{4}$$

since $|e| = k$.

Approximation Guarantee.

Theorem 1. *Iterative packing on an extreme point of the natural LP relaxation in the order provided by Lemma 1 with X_u as derived in the proof of Lemma 2 is a $k - 1 + 1/k$ approximation algorithm.*

Proof. Lemma 1 gives us the existence of a $v \in e$ such that $x_e \ge x(\delta(v))/k$, which in conjunction with (4) yields:

$$\rho \le (1/k - 1) \cdot x(\delta(v)) + |e''| + \sum_{u\in e'} (x(\delta(u)) - (\beta_u - 1)), \tag{5}$$

since $k \ge 1$. If $v \in e'$ then we see that

$$\begin{aligned}
\rho &\le 1/k \cdot x(\delta(v)) - (\beta_v - 1) + |e''| + \sum_{u\in e'\setminus\{v\}} (x(\delta(u)) - (\beta_u - 1)) \\
&\le 1/k \cdot x(\delta(v)) - (\beta_v - 1) + k - 1 \\
&\le 1/k \cdot x(\delta(v)) - 1/k \cdot (\beta_v - 1) + k - 1 \quad [\text{since } \beta_v \ge 1] \\
&\le k - 1 + 1/k,
\end{aligned}$$

which follow from $k = |e'| + |e''|$ and $x(\delta(u)) - (\beta_u - 1) \leq 1$ for all $u \in V$. On the other hand, if $v \in e''$ then (5) implies:

$$\rho \leq (1/k - 1) \cdot x(\delta(v)) + k < k - 1 + 1/k;$$

to see the latter inequality note that $k \geq 1$ and $x(\delta(v)) > 1$, since $v \in e''$ implies $\beta_v = \hat{\beta}_v + 1 \geq 2$. ☐

Improvement for k-partite instances. If our k-uniform hypergraph is k-partite, then we may appeal to a strengthening of Lemma 1 from Chan and Lau which sharpens the lemma to produce

$$x_{e_i} \geq \frac{x(\delta(v_i) \cap E_i)}{k - 1},$$

instead of $x_{e_i} \geq x(\delta(v_i) \cap E_i)/k$. This directly implies the existence of a $v \in e$ with $x_e \geq x(\delta(v))/(k-1)$ above, and making the appropriate substitutions above yields $\rho \leq k - 1$.

3 Application: Allocations

Consider the following general auction setting: you have a set of n bidders and a set of m items, with the only restriction being that each bidder can win at most t items, where t is a fixed constant. Observe that for this simple setting, we can even explicitly specify each bid in polynomial space, since each bidder has only $\binom{m}{t} + \binom{m}{t-1} + \cdots$ outcomes. What kind of truthful, approximately-efficient mechanisms exist for this setting?

We will take advantage of the Lavi-Swamy framework [13], which is a fractional version of the well-known Vickrey-Clarke-Groves (VCG) mechanism. We cannot directly use VCG in this setting, because one of the steps in VCG is to compute the allocation which maximizes the total utility of all players, and this problem is NP-complete in our setting for $t \geq 2$, by a reduction from 3-dimensional matching. The main result of Lavi and Swamy is that once we have an *LP-relative α-approximation algorithm* with respect to the natural LP, we can get a truthful-in-expectation mechanism, which also maximizes the expected overall utility within a factor of α. Minimizing this factor means we are coming closer to a VCG-like mechanism, whereas allocating everyone the empty set is truthful but is a bad approximation.

First we define the natural LP relaxation for the allocation problem. Let x^i_S be a fractional indicator variable indicating whether player i will win exactly the set S of items. Then the LP requires that each player wins one set of items, and that each item is allocated at most once, fractionally. Write v^i_S as the valuation of player i for set S. Then the fractional allocation LP is:

$$\max \sum_{i,S} x^i_S v^i_S : 0 \leq x \leq 1; \forall i \in [n] : \sum_S x^i_S = 1; \forall s \in [m] : \sum_i \sum_{S : s \in S} x^i_S \leq 1. \quad (\mathcal{A})$$

Note that in our sample application x^i_S and v^i_S are defined only for sets S of size at most t, and so the LP has polynomial size.

Definition 1. *An* LP-relative α-approximation algorithm *for the allocation problem is one that, for all nonnegative v, outputs an integer feasible solution to $(\mathcal{A})$, such that its value is at least $1/\alpha$ times the LP optimum of $(\mathcal{A})$.*

Often this is equivalently described as an α-approximation algorithm which also bounds the integrality gap of $(\mathcal{A})$ by α.

Definition 2. *An* α-approximate truthful-in-expectation mechanism *for the allocation problem is a randomized algorithm of the following form. It takes the values v as inputs; its outputs are a valid allocation of items to players together with prices p_i charged to each player i. It has the following two properties. First, where $S(i)$ denotes the set of items allocated to player i, we have $\sum_i v^i_{S(i)}$ is at least $\sum_i v^i_{T(i)}/\alpha$ for every valid allocation T. Second, for every fixed v^{-i}, a player who gives insincere valuations $\widehat{v}^i$ as their input, resulting in random variables $\widehat{p}, \widehat{S}$ compared to the original ones p, S, does not increase their expected net utility:*

$$E[v^i_{\widehat{S}(i)} - \widehat{p}_i] \leq E[v^i_{S(i)} - p_i].$$

Moreover, $0 \leq E[p_i] \leq E[v^i_{S(i)}]$ for all i.

Theorem 2 (Lavi-Swamy [13]). *Given a polynomial-time LP-relative α-approximation algorithm for an allocation problem, we can obtain a polynomial-time α-approximate truthful-in-expectation mechanism.*

We note that the LP-relative property is essential. Our allocation problem is a set packing problem with sets of size at most $t+1$, and there is a $\frac{t}{2}+2+\epsilon$-approximation for this problem [15], but we cannot use it with the Lavi-Swamy theorem because it is not LP-relative. Our previous results, and the work of Chan and Lau [4], give an LP-relative $(k-1+1/k)$-approximation. We will show that this can be improved to a $(k-1)$-approximation.

Chan and Lau observe that for a general k-uniform hypergraph, any extreme point solution of the matching LP has a value with degree at most k in the support; and in Lemma 2.3 they show that for the special case of k-dimensional hypergraphs, the k can be tightened to k-1. However, their proof immediately implies that a more general result holds: it is enough that there exists a set W_1 of vertices so that every hyperedge intersects W_1 exactly once. This coincides with the setting needed for our application, where W_1 is the set of bidders and all other vertices correspond to items that are up for auction.

4 A primal-dual approach for k-hypergraph demand matching

In this section we reinterpret the $2k$-approximation for k-hypergraph demand matching from [14] as a simple primal-dual algorithm. Although the analysis of our algorithm bears resemblance to the original, it does not require solving the LP relaxation or maintaining an approximate convex decomposition. The inspiration for our algorithm is a similar connection between a primal-dual algorithm and an iterative packing like algorithm from Bar-Yehuda et al. [2].

Algorithm 2 ALG(V, E, d, b, w)

1: For each $e \in E$ with $w_e \leq 0$, remove e from E.
2: Pick $e \in E$ such that d_e is minimum.
3: Define a new weight function $w_1 \in \mathbb{R}^E$ by

$$w_1(f) = \begin{cases} 1, & \text{if } f = e \\ \sum_{v \in e \cap f} d_f / \max\{b_v - d_e,\, d_e\}, & \text{otherwise.} \end{cases}$$

and multiply it by the scalar $w(e)$ to get the weight function $w(e) \cdot w_1$.
4: Recurse: $\mathcal{F}' := \text{ALG}(V, E - e, d, b, w - w(e) \cdot w_1)$.
5: Return $\{e\} \cup \mathcal{F}'$ if it's feasible, else return $\mathcal{F}'$.

The primal-dual algorithm. We use the following *local ratio lemma* (we skip the proof): if some feasible $\mathcal{F}$ is an ρ-approximately optimal solution for weight function w_1, and also for weight function w_2, then it is ρ-approximately optimal for any nonnegative linear combination of w_1 and w_2. The analysis of Algorithm 2 is comprised of the following two lemmas.

Lemma 3. *For each w_1 constructed by the algorithm, the final result ALG of the top-level call is $2k$-approximately optimal.*

Proof. We will show that $w_1(ALG) \geq 1$ and $w_1(OPT) \leq 2k$.

Claim. $w_1(ALG) \geq 1$.

Proof. On the one hand, if we return $\{e\} \cup \mathcal{F}'$ this is obvious as $w_1(e) = 1$, and w_1 is non-negative. On the other hand, if $\{e\} \cup \mathcal{F}'$ is not feasible, it means that the constraint of some vertex v is preventing the inclusion of e. Thus $d_e + \sum_{f \in \mathcal{F}' : v \in f} d_f > b_v$, so $\sum_{f \in \mathcal{F}' : v \in f} d_f > b_v - d_e$. Furthermore, since e was chosen to have minimal demand, each edge in $\{f \in \mathcal{F}' \mid v \in f\}$ has $d_f \geq d_e$. Consequently $\sum_{f \in \mathcal{F}' : v \in f} d_f \geq d_e$. Now look at the definition of w_1 — we defined it precisely so that these two conditions together imply $w_1(\mathcal{F}') \geq 1$. □

Claim. Every feasible $\mathcal{F}_0$ has $w_1(\mathcal{F}_0) \leq 2k$.

Proof. Note that $\max\{b_v - d_e,\, d_e\} \geq b_v/2$. Consequently, for $f \neq e$ we have $w_1(f) \leq \sum_{v \in e \cap f} \frac{2d_f}{b_v}$. Moreover, for each v the sum $\sum_{e \in \mathcal{F}_0 | v \in e} d_f$ is at most b_v. Reversing the order of summation and using $|e| \leq k$ gives the claim. □

This proves the lemma. □

Lemma 4. *Let w' be the original weight function minus all the $w(e) \cdot w_1$'s produced by the algorithm. Then the final result ALG of the top-level call is 2k-approximately optimal for w'.*

Proof. We have $w' \leq 0$. Moreover, for each e picked in Line 2 of the algorithm, $w'_e = 0$, and ALG returns a subset of these. □

Combining the lemmas and using the "local ratio lemma," proves the $2k$-approximation. We may extend the analysis to show that the algorithm gives a bound on the integrality gap of the natural relaxation.

References

1. N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. On -column sparse packing programs. In F. Eisenbrand and F. B. Shepherd, editors, *IPCO*, volume 6080 of *Lecture Notes in Computer Science*, pages 369–382. Springer, 2010.
2. R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006.
3. P. Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. In M. M. Halldórsson, editor, *SWAT*, volume 1851 of *Lecture Notes in Computer Science*, pages 214–219. Springer, 2000.
4. Y. Chan and L. Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Mathematical Programming*, pages 1–26, 2011.
5. C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms*, 3(3), 2007.
6. U. Feige and M. Singh. Edge coloring and decompositions of weighted graphs. In D. Halperin and K. Mehlhorn, editors, *ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 405–416. Springer, 2008.
7. M. Feldman, J. Naor, R. Schwartz, and J. Ward. Improved approximations for k-exchange systems - (extended abstract). In *Proc. 19th ESA*, pages 784–798, 2011.
8. Z. Füredi, J. Kahn, and P. D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.
9. E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating -set packing. *Computational Complexity*, 15(1):20–39, 2006.
10. C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discrete Math.*, 2(1):68–72, 1989.
11. C. Koufogiannakis and N. E. Young. Distributed fractional packing and maximum weighted b-matching via tail-recursive duality. In *DISC*, pages 221–238, 2009.
12. P. Krysta. Greedy approximation via duality for packing, combinatorial auctions and routing. In *MFCS*, pages 615–627, 2005.
13. R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 595–604, Washington, DC, USA, 2005. IEEE Computer Society.
14. O. Parekh. Iterative packing for demand and hypergraph matching. In *Proc. 15th IPCO*, pages 349–361. 2011.
15. J. Ward. A $(k + 3)/2$-approximation algorithm for monotone submodular k-set packing and general k-exchange systems. In *Proc. 29th STACS*, pages 42–53, 2012.