Generalized Hypergraph Matching via Iterated Packing and Local Ratio

Ojas Parekh[1] and David Pritchard[2]

[1] Sandia National Laboratories*
[2] Department of Computer Science, Princeton University

Abstract

In k-hypergraph matching, we are given a collection of sets of size at most k, each with an associated weight, and we seek a maximum-weight subcollection whose sets are pairwise disjoint. More generally, in k-hypergraph b-matching, instead of disjointness we require that every element appears in at most b sets of the subcollection. Our main result is a linear programming-based $(k - 1 + \frac{1}{k})$-approximation algorithm for k-hypergraph b-matching. This settles the integrality gap when k is one more than a prime power, since it matches a previously-known lower bound. When the hypergraph is bipartite, we are able to improve the approximation ratio to $k - 1$, which is also best possible relative to the natural LP. These results are obtained using a more careful application of the *iterated packing* method.

Using the bipartite algorithmic integrality gap upper bound, we show that for the family of combinatorial auctions in which anyone can win at most t items, there is a truthful-in-expectation polynomial-time auction that t-approximately maximizes social welfare. We also show that our results directly imply new approximations for a generalization of the recently introduced bounded color matching problem.

We also consider the generalization of b-matching to *demand matching*, where edges have nonuniform demand values. The best known approximation algorithm for this problem has ratio $2k$ on k-hypergraphs. We give a new algorithm, based on local ratio, that obtains the same approximation ratio in a much simpler way.

*MS 1326, Albuquerque NM 87185, USA. 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.

1 Introduction

In a matching problem we want to find the maximum weight subcollection of pairwise disjoint sets within a given collection. Often these problems are studied with respect to the maximum set size k (i.e. on "k-hypergraphs"); matching is polynomial-time solvable for $k = 2$, while it is APX-hard for $k = 3$, even in special cases like *3-dimensional matching* [19].

The b-matching problem generalizes matching: the input specifies a limit b_v for every vertex, and we can select at most b_v sets containing each v; ordinary matching results when b is the all-1 vector. A b-matching instance can allow each set e to be selected multiple times up to some upper *capacity* limit c_e. *Simple* b-matching is the case where all capacities are unit. The *uncapacitated* case is where $c = \vec{\infty}$, i.e. there are no capacity limits.

One of our results considers the generalization of b-matching to *demand matching*, a notion originally introduced for graphs in [27]. For this problem each edge is given a demand value d_e, and we now constrain that for every vertex v, the sum of the d-values of the incident edges should be at most b_v. When d is the all-1 vector we recover the b-matching problem.

Hypergraphic matching problems are often studied via linear programming relaxations. In this paper we use only the naive LP relaxations. The worst-case ratio between the LP optimum and the optimal integral solution is called the *integrality gap*. An *LP-relative α-approximation algorithm* is one that produces (in polynomial time) an integral solution of value at least $1/\alpha$ times the LP's optimal value — this both upper bounds the integrality gap by α and gives an α-approximation algorithm. Many classical approximation algorithms are LP-relative; so the notion is not novel, rather, this terminology helps us be concise.

1.1 Results

Our main result is the following theorem.

Theorem 1. *There is an LP-relative $(k - 1 + \frac{1}{k})$-approximation algorithm for k-hypergraph b-matching, for any capacities.*

In [24] one of the authors announced a weaker result, namely an upper bound of $k - 1 + \frac{1}{k}$ on the integrality gap, but without any algorithm that could find such a good integral solution in polynomial time.

For the special case $b = 1$, Füredi, Kahn and Seymour [15] proved an upper bound of $k-1+\frac{1}{k}$ in 1993, while Chan & Lau [10] recently gave the first poly-time algorithm matching this bound. Their technique does not directly extend to the k-hypergraph b-matching case. The technique that we use to prove Theorem 1 is *iterated packing*, the same technique from [24]. Part of the contribution of the present paper is to simplify the algorithms from [24] and [10]. Our main technical innovation is, in iterated packing, to explicitly specify particular additional solutions as ineligible for packing: not only solutions that would be ineligible for the *original* problem, rather we additionally prohibit solutions exceeding the ceiling of the current fractional solution.

Theorem 1 is tight for infinitely many k: when $k - 1$ is a prime power, as observed in [15], the projective plane $\mathsf{PG}(2, k - 1)$ of order $k - 1$ yields a matching lower bound of $k - 1 + \frac{1}{k}$ on the integrality gap. It is an interesting open question to settle the integrality gap for any other values of k.

We are able to determine the exact integrality gap for another interesting class of hypergraphs. Call a hypergraph *bipartite* ([1]; cf. [25]) if, for some distinguished subset U of vertices, every hyperedge contains exactly one vertex from U.

Theorem 2. *There is an LP-relative* $(k-1)$*-approximation algorithm for* bipartite k-hypergraph b-matching, *for any capacities.*

Chan and Lau [10] proved Theorem 2 in the special case that $b = 1$ and the instance is k-dimensional[1]. Proving Theorem 2 is similar to Theorem 1 plus extending an observation of [10] from k-dimensional hypergraphs to bipartite ones. Like Theorem 1, a matching integrality gap lower bound is known [14, p. 157] when $k-1$ is a prime power: the hypergraphic dual of the affine geometry $\mathsf{AG}(2, k-1)$, i.e. a truncated projective plane, has integrality gap $k-1$.

We obtain the following interesting corollaries from the bipartite case. In the bounded color k-hypergraph b-matching problem we are given an instance of the k-hypergraph b-matching problem along with a partition of the edge set into l color classes, $E = E_1 \cup \cdots \cup E_l$, and a positive integer w_i for $1 \leq i \leq l$. We seek a feasible k-hypergraph b-matching of maximum weight such that at most w_i edges from class E_i is selected for each i.

Corollary 3. *There is an LP-relative k-approximation for bounded color k-hypergraph b-matching.*

Corollary 4. *For combinatorial auctions where each bidder can win at most $(k-1)$ items, there is a randomized polynomial-time mechanism that, in expectation, is both truthful and $(k-1)$-approximately maximizes social welfare.*

We are not aware of any prior results for this extremely natural class of combinatorial auctions, cf. [23, Ch. 12].

The proof of Corollary 4 uses the mechanism of Lavi and Swamy [22], where the distinguished vertices in the bipartite hypergraph correspond to the bidders. For this application, it is crucial that Theorem 2 gives an *LP-relative* approximation in polynomial time.

Finally, we give a new short proof of the following known theorem:

Theorem 5 ([24])**.** *There is an LP-relative $2k$-approximation for k-hypergraph demand matching.*

Our simpler proof is based on the local ratio method, rather than the iterated packing used in [24]. We rely on a connection in [5, p. 12] between local ratio and iterated packing.

1.2 Related Work

As Tutte observed [28], both in edge-weighted graphs and in the cardinality case, uncapacitated graphic b-matching can be reduced to matching by replacing each vertex by b_v clones. Each edge uv is likewise cloned $b_u b_v$ times. This reduction has two problems: (1) the clones cause an exponential increase in the instance size (from $\lg\|b\|_1$ to $\|b\|_1$); and (2) it does not work in the capacitated case, since we need to prevent too many clones of the same edge from being selected. Cloning applies to hypergraphs, too, but has the same two problems. Algorithmically, we can often avoid (1) by not dealing with the clones explicity. For graphs we can fix problem (2): an edge-trisecting reduction [28] (see also [26, p. 562]) extends cloning to work on capacitated instances. But for hypergraphs, there is no known workaround for problem (2).

As a strawman, let us mention that one can reduce capacitated b-matching in k-hypergraphs to uncapacitated b-matching in $(k+1)$-hypergraphs, by inserting new vertices in each hyperedge

[1] A hypergraph is k-dimensional if for some k-partition of the ground set, every edge intersects every part exactly once.

and by moving each edge's capacity to the b value of its new vertex. One can even then apply cloning. But this is not that useful for us: e.g., we cannot use the previously-known $b = 1$ case of version Theorem 1 to even prove the nonconstructive version of Theorem 1 for general b, since this reduction increases the hyperedge size from k to $k + 1$.

Algorithmically, the *simple* (capacity $c = \mathbf{1}$) case of b-matching is the hardest. The proof is standard, by fixing the integer part of an optimal fractional solution — see Appendix A.

Observation 6. *Given an (LP-relative) α-approximation to simple b-matching in k-hypergraphs, we can obtain the same quality of approximation for general capacities.*

Hypergraph matching. Matching problems in k-uniform hypergraphs are well-studied algorithmically. For any fixed $\varepsilon > 0$ the best known approximation ratios are $\frac{k}{2} + \varepsilon$ for the unweighted version by Hurkens and Schrijver [18] and $\frac{k+1}{2} + \varepsilon$ for the weighted version by Berman [8]. In the case $k = 3$, the algorithmic results of [10] give an ε-improved approximation ratio of 2 for 3-dimensional matching. On the other hand, Hazan, Safra and Schwartz [17] showed that the problem is hard to approximate within a factor of $\Omega(\frac{k}{\log k})$ unless $\mathsf{P} = \mathsf{NP}$, even in the k-dimensional case.

Hypergraph b-matching. For b-matching in k-hypergraphs, Krysta [21] gave a greedy $k+1$-approximation for the simple case, and Young & Koufogiannakis [20] gave a k-approximation for the uncapacitated version. Both of these approximation algorithms give LP-relative guarantees. An improvement in some cases was recently obtained by the *k-exchange system* framework of Feldman et al. [13]. The b-matchings form a *k-exchange system* (this is explicit only for $k = 2$ in [13]). In this way one can obtain a local search-based $(\frac{k+1}{2} + \varepsilon)$-approximation algorithm for weighted k-hypergraph b-matching. However, its running time is exponential in k and it does not give any LP-relative guarantee.

It may be tempting to think that the b-matching problem in hypergraphs is a simple extension of 1-matching in hypergraphs because the theory and algorithms for b-matching in graphs closely relate to those for 1-matchings. As evidenced by the results above, this does not appear to be the case. An approximation algorithm that runs in time polynomial in k with guarantee better than k for k-hypergraph b-matching had been an open problem that we resolve with this work. Our methods are LP-based, whereas local search seems to give the best known results; however, the bounding techinques used in local search for hypergraph 1-matching do not seem to readily extend to the hypergraph b-matching case. For example, Arkin and Hassin [2] give a local search $(k - 1 + \varepsilon)$-algorithm for weighted k-hypergraph 1-matching; however, as a warmup they present a trivial bound of k — even this trivial bound does not easily extend to the k-hypergraph b-matching case.

Other work. Pseudo-greedy methods similar to iterative packing have been successfully applied to several packing and coloring problems, including multicommodity flows on trees [11], independent sets in t-interval graphs [5], and weighted edge coloring of bipartite graphs [12].

As mentioned earlier, a $2k$-approximation for k-hypergraph demand matching is known [24]; a better ratio of 3 is possible when $k = 2$ [24]. These nearly match (exactly match, when $k = 2$) the best known lower bound of $2k - 1$ [3] on the integrality gap of the natural LP relaxation (this construction does not require that $k - 1$ is a prime power). Bansal et al. [3] devised a deterministic $8k$-approximation and a randomized $(ek + o(k))$-approximation for the more general problem of approximating k-column-sparse packing integer programs.

Georgios very recently introduced the bounded color matching problem (defined above in a more general context) and devised a 2-approximation [16]. This result is also based on iterated packing. The author observes that the bounded color matching problem is a special case of 3-hypergraph b-matching. In fact it is mentioned in this paper that a polynomial-time $k-1+1/k$-approximation for k-hypergraph b-matching may be possible. Our work was developed independently of this work, and we observe that our results generalize this work since the special hypergraph b-matching instances obtained are bipartite, and we are able to give a k-approximation.

We will exploit the interplay between LP-relative approximation algorithms and convex decompositions — an equivalence between the two was shown by Carr & Vempala [9]. The Lavi-Swamy [22] mechanism combines techniques from [9] with the VCG mechanism.

We give an overview of iterated packing in the next section. There, we also introduce a structure theorem from [10] and its specialization to bipartite instances, versions of which will be used throughout the paper. Next, to further introduce the iterated packing methodology, we give an iterated packing proof of the same result, although it does not run in polynomial time. This is extended to b-matching in Section 4, which contains our main technical innovations. First an existential proof is given (Algorithm 2) and then finally Algorithm 3 proves Theorems 1 and 2 constructively. Then in Section 7 we present the proof of Theorem 5, which is based on the local ratio method.

2 Iterated Packing Overview

The notion of an approximate convex decomposition is essential to iterated packing, as the latter iteratively builds such a decomposition for a given fractional solution. Here we present a slightly different notion of an approximate convex decomposition than usually considered.

Definition 7. For $\alpha \geq 1$, define *α-convex multipliers* to be any collection of nonnegative reals whose sum is α. Likewise, we say that x is an *α-convex combination* of the points $\{x^i\}_i$ if there are α-convex multipliers $\{\lambda_i\}_i$ so that $x = \sum_i \lambda_i x^i$.

The utility of α-convex combinations is that they provide a convenient way to talk about integrality gaps without rescaling as was done in [9] or [24].

Proposition 8 ([9])**.** *If every feasible LP solution for a packing program can be written as an α-convex combination of integral feasible solutions, then its integrality gap is at most α.*

Proof. We need to show that for any nonnegative weight function w, if x^* is the fractional solution that maximizes $w(x^*)$, then there is an integral solution of weight at least $w(x^*)/\alpha$. A random solution from the α-convex combination representation of x^*, drawing x^i with probability λ_i/α, has expected weight $\sum_i \frac{\lambda_i}{\alpha} w(x^i) = w(x^*)/\alpha$. So one of the x^i has at least this weight. □

(In fact [9] also proves an algorithmic converse, used also by the Lavi-Swamy framework [22] underlying Corollary 4.)

We will use Proposition 8 as follows: we develop a polynomial-time algorithm to write fractional hypergraph b-matchings as ρ-convex combinations of feasible integral b-matchings. Then by Proposition 8, we get the LP-relative ρ-approximation algorithm claimed in Theorems 1 and 2.

In [24] the idea of iterated packing was introduced. Each iteration, called a *packing step*, updates the current α-convex combination to a new one, increasing some terms of the combination on one coordinate.

Definition 9 (Packing step)**.** Let us be given an α-convex combination $x = \sum_i \lambda_i x^i$ where the x^i are feasible integral solutions, an edge e to pack, and a target value $t \in \mathbb{R}_+$. We may think of a *packing step* as packing the edge e into some of the solutions x^i such that each resulting solution is still feasible and that we have packed e into solutions with a total mass of t, i.e. the sum of corresponding λ_i is t.

Let χ_e be the vector in $\mathbb{R}^E$ with coordinate 1 on e and 0 elsewhere. A *packing step* will replace some $0 \leq \lambda'_i \leq \lambda_i$ portion of each x^i with $x^i + \chi_e$, where we allow $\lambda'_i > 0$ only when $x^i + \chi_e$ is feasible. Therefore $\sum_i(\lambda_i - \lambda'_i) \cdot x^i + \sum_i \lambda'_i \cdot (x^i + \chi_e)$, the result of the packing step, expresses $x + t\chi_e$ as an α-convex combination of integer feasible solutions.

For a packing step to actually be feasible, it is clearly both necessary and sufficient that the set $P = \{i \mid x^i + \chi_e \text{ feasible}\}$ of solutions into which e can be packed must satisfy $\lambda(P) \geq t$.

For the sake of polynomial-time implementation of our final algorithm, note we can ensure at most one i has $\lambda'_i \notin \{0, \lambda_i\}$ in the above argument, so that each packing step increases the number of terms by at most one. Alternatively we could use Carathéodory's theorem which guarantees that any α-convex combination can be rewritten as one with at most $d + 1$ terms where d is the number of coordinates.

The basic iterated packing formula starts with a fractional solution x in hand and iteratively constructs an integral solution by starting with an empty hypergraph on V. The edges are processed in some order, and for each edge e, a packing step is performed on e with a target value of x_e. One key fact about iterated packing is that when a target value is larger, packing is easier, hence iterated packing shows how large fractional values facilitate approximation for packing problems much like iterative packing does for covering problems. The basic approach may be refined in several directions. One may start with base integral solution that is non-empty hypergraph. This was explored in [24] to derived an improved approximation for the demand matching problem. Another improvement is to consider a specific ordering of edges.

This key idea driving our algorithm is analyzing an ordering of edges which allows us to obtain a polynomial-time algorithm. Although, as announced in [24], extensions of ideas from [24] may be used to derive an upper bound of $k-1+1/k$ on the integrality gap for the k-hypergraph b-matching problem, the bound is non-constructive and does not give a polynomial-time algorithm. We show that by considering an ordering of edges that was first studied by Chan and Lau [10], we obtain a polynomial-time $k - 1 + 1/k$-approximation. This ordering is based on vertices of small degree in an extreme point solution, which in turns allows one to argue that there is an edge with large fractional value. The lemma below shows that we can find a vertex of sufficiently small degree.

Let $\{A_{v,e}\}_{v,e}$ be the 0-1 incidence matrix for our k-hypergraph: it has rows for vertices and columns for edges, with at most k ones per column. When x^* is an extreme point solution to the matching LP $\{0 \leq x \leq 1 \mid Ax \leq 1\}$, elementary properties of polyhedra show that the incidence matrix of $\{e \mid 0 < x^*_e < 1\}$ has linearly independent columns. This makes the following lemma useful: it was proven by Chan and Lau for the general case, while the bipartite case follows from generalizing their arguments about the k-dimensional case (see Appendix A).

Lemma 10. *If the incidence vectors of $\varnothing \neq E' \subseteq E$ are linearly independent, then some vertex in (V, E') has degree between 1 and k. In the bipartite case, the upper bound can be strengthened to $k - 1$.*

In order to talk about both the general and bipartite cases in a unified way, define

$$\rho := \begin{cases} k-1+\frac{1}{k} & \text{in the general case, and} \\ k-1 & \text{in the bipartite case.} \end{cases}$$

Additionally, define the degree bound

$$\mu := \begin{cases} k & \text{in the general case, and} \\ k-1 & \text{in the bipartite case.} \end{cases}$$

3 Non-Polynomial Time Algorithm for k-Hypergraph Matching

We now give an alternate proof that k-hypergraph matching has integrality gap of at most $k+1-\frac{1}{k}$. The algorithm behind this proof does not run in polynomial time. However, this section also introduces the notation and steps involved in iterated packing, which we will extend in the next section to get our main result.

Lemma 11 ([24]). *In k-hypergraph matching, a packing step to bring x to $x+t\chi_e$, where $x+t\chi_e$ is a feasible fractional solution, is possible if $\alpha \geq k-(k-1)t$.*

Proof. Let Q_v, for each $v \in e$, be the set of solutions i for which $x^i + \chi_e$ is not feasible. We have $\lambda(Q_v) \leq 1-t$ since $x+t\chi_e$ is feasible[2]. We need room (disjoint in the worst case) for all such Q_v, plus an additional t to pack the new edge in solutions that permit it, giving the bound $k(1-t)+t = k-(k-1)t$. □

We can indeed get large coordinates using the following strengthening of Lemma 10 (see Appendix A).

Lemma 12. *Any nonzero extreme point solution x to the k-hypergraph matching polytope has some fractional coordinate at least $1/\mu$.*

Using this, we obtain an iterated packing algorithm for the k-hypergraph matching problem, which is displayed as Algorithm 1. Note that this algorithm is presented as a recursive top-down variant of iterated packing, while the basic version in the previous section was presented as a bottom-up algorithm for ease of exposition. Another more crucial deviation of this algorithm from the basic iterated packing formula is that since our analysis requires an extreme point, we must express each non-extreme solutions as convex combinations of extreme points, and we use that:

$$\text{A convex combination of } \alpha\text{-convex combinations is an } \alpha\text{-convex combination.} \qquad (1)$$

In fact this is the reason the algorithm is not guaranteed to run in polynomial time; however, the algorithm does terminate since the number of nonzero coordinates of x decreases in each recursive call.

Proposition 13. *Given any LP solution x, Algorithm 1 returns an expression of x as a ρ-convex combination of integral solutions.*

[2] In detail, the solutions x^i for $i \in Q_v$ have degree 1 at v, so by the definition of a convex combination $(Ax)_v = \lambda(Q_v)$, but $(Ax)_v \leq 1-t$ since, by feasibility, $1 \geq A(x+t\chi_e)_v = (Ax)_v + t$.

Algorithm 1 HM$^*(V, E, x)$ // write x as ρ-convex comb. of 0-1 solutions

```
If $x = \mathbf{0}$ return the trivial $\rho$-convex combination $\lambda_1 = \rho, x_1 = \mathbf{0}$.
If $x$ is not an extreme point solution to $\{x \in \mathbb{R}_+^E \mid Ax \leq 1\}$,
    Write $x$ as a convex combination of extreme point solutions.
    Recurse on each extreme point and return their result combined via (1).
Pick $e$ so that $x_e$ is maximized and let $x'$ be $x$ except with $x'_e$ set to zero.
Recurse: $(x^i, \lambda_i)_i :=$ HM$^*(V, E, x')$.
Packing step: pack $x_e$ of $e$ into $(x^i, \lambda_i)_i$ and return the result.
```

Proof. This follows from Lemmas 11 and 12, since if $\mu = k$ we have $k - (k-1)/\mu = k - 1 + \frac{1}{k} = \rho$, and if $\mu = k - 1$ we have $k - (k-1)/\mu = k - 1 = \rho$. □

This completes the non-polynomial time iterated packing proof that the integrality gap for matching is at most ρ. Next, we extend it to b-matching.

4 Iterated Packing and k-Hypergraph b-Matching

In this section, which contains the main new iterated packing technique, we build on the ideas from the previous section. We begin with a non-constructive iterated packing algorithm to show that the integrality gap for k-hypergraph b-matching is at most ρ. Then, we move to a constructive version via iterated packing that runs in polynomial time.

By Observation 6, we assume unit capacities (simple b-matching). We will use the following statement, whose proof is analogous to Lemma 12.

Lemma 14. *Any nonzero extreme point solution x to the k-hypergraph b-matching polytope has some fractional coordinate at least $1/\mu$.*

The naive adaptation of iterated packing (Algorithm 1) to b-matching would involve writing the input as a convex combination of extreme point solutions to $\{x \in [0, 1]^E \mid Ax \leq b\}$, working with α-convex combinations of integer 0-1 solutions to $Ax \leq b$. However, this approach is unworkable. When we try to mimic Lemma 11, as b gets larger, we cannot bound $\lambda(Q_v)$ by anything less than 1, giving an approximation ratio of k or worse.

To fix this problem, we will enforce two additional conditions. One of these conditions, the main driver of the new proof, is that the *strengthened* degree bound $Ax^i \leq \lceil Ax \rceil$ must hold in *every* level of the recursion (rather than the unworkable requirement that solutions merely respect the final target degrees). The second condition is that the λ-mass of solutions meeting this strengthened bound with equality cannot be more than $\langle (Ax)_v \rangle$ (here $\langle \cdot \rangle$ denotes the fractional part), except in the degenerate case that $(Ax)_v$ is integral. Intuitively (i) balances the number of edges packed at a vertex across the solutions x^i, avoiding the trouble that the naive approach would encounter in future iterations, while (ii) helps achieve (i) inductively. A *modified packing step* is a packing step that, given a solution (x, λ) satisfying both of these properties, produces another (x', λ') satisfying both of these properties. Then the definition of the resulting algorithm, Algorithm 2, is as follows.

We will prove by induction that the algorithm succeeds in finding packings meeting both conditions.

Algorithm 2 HbM$^*(V, E, x)$ // write x as ρ-convex comb. of special 0-1 solutions

```
If $x = \mathbf{0}$ return the trivial $\rho$-convex combination $\lambda_1 = \rho, x_1 = \mathbf{0}$.    // as before
If $x$ is not an extreme point solution to $\{y \in [0,1]^E \mid Ay \leq \lceil Ax \rceil\}$,
    Write $x$ as a convex combination of extreme point solutions.    // as before
    Recurse on each extreme point; return their combination via (1).    // as before
Pick $e$ so that $x_e$ is maximized; let $x'$ be $x$ with $x'_e$ set to zero.    // as before
Recurse: $(x^i, \lambda_i)_i :=$ H$b$M$^*(V, E, x')$.    // as before
Modified packing step: pack $x_e$ of $e$ into $(x^i, \lambda_i)_i$ and return the result.
```

Lemma 15. *For any $0 \leq x \leq 1$, HbM*(V, E, x) returns an expression of x as a ρ-convex combination of 0-1 x^i that satisfies (i) $Ax^i \leq \lceil Ax \rceil$ for each i, and (ii) for every v such that $(Ax)_v$ is non-integral, $\lambda(\{i \mid (Ax^i)_v = \lceil Ax_v \rceil\}) \leq \langle (Ax)_v \rangle$.*

For the proof, it is helpful to realize that we use $(Ax)_v$ interchangeably as $x(\delta(v))$, and that it represents the "degree" of x at v.

Proof. The base case and the non-extreme case are easy; while the extreme points decomposing a non-extreme solution may have smaller values for $\lceil Ax \rceil$, this does not hurt us. So we only need to deal with the case that x is extreme and nonzero, where e is chosen with $x_e \geq 1/\mu$.

To prove that the modified packing step can always be carried out while satisfying (i) and (ii), we again bound a set of unpackable solutions. Specifically, our goal will be to define sets Q_v for each $v \in e$ such that any packing step that avoids adding e to any of the solutions $\bigcup_{v \in e} Q_v$ will satisfy (i) and (ii) for x, and such that the sets Q_v are λ-small enough that e always has room to be added.

For each $v \in e$, there are three cases, the main distinction being whether $\lceil (Ax')_v \rceil = \lceil (Ax)_v \rceil$. Note that these terms are either equal, or differ by one.

- Case (I), $(Ax')_v = 0$. This packing is trivial, set $Q_v = \varnothing$.

- Case (II), $\lceil (Ax)_v \rceil = \lceil (Ax')_v \rceil \neq 0$. Proving (ii) is vacuous when $(Ax)_v$ is integral, and otherwise it follows easily by induction since $\langle (Ax)_v \rangle = \langle (Ax')_v \rangle + x_e$ and at most x_e of λ-mass of solutions will have its degree increased at v. To show (i) is satisfied inductively, just like in Section 3, define Q_v to be the set of i with $(Ax^i)_v = \lceil (Ax)_v \rceil$; e can be added to any other x^i without violating the degree constraint. The terms $(Ax)_v$ and $(Ax')_v$ differ by x_e and have the same integer ceiling, so by induction on (ii), we have the bound $\lambda(Q_v) \leq \langle x'(\delta(v)) \rangle \leq 1 - x_e$ showing that Q_v is not too big. This bound will be used later.

- Case (III), $\lceil (Ax)_v \rceil = 1 + \lceil (Ax')_v \rceil$ and $(Ax')_v \neq 0$. Then satisfying (i) at v is easy (since all x^i have degree at most $\lceil (Ax')_v \rceil$ at v) but we must design Q_v so that (ii) is satisfied after the packing step.

  If $(Ax)_v$ is integral any packing works (we can take $Q_v = \varnothing$), so assume the opposite. Moreover, when $(Ax')_v$ is integral, by (i) all solutions x^i have degree less than $\lceil (Ax)_v \rceil$ at each $v \in e$, and since we are only packing $x_e = \langle (Ax)_v \rangle$ amount of e, (ii) is also satisfied by any possible packing.

  Hence, assume both $(Ax')_v$ and $(Ax)_v$ are non-integral. If we pack e arbitrarily, the total weight of new solutions with degree $\lceil (Ax)_v \rceil$ at v could be too large to satisfy (ii). Therefore, we will define Q_v to exclude some subset of the solutions $Q'_v := \{i \mid (Ax^i)_v = \lceil (Ax')_v \rceil\}$ that could rise to have this degree. We have $\lambda(Q'_v) \leq \langle (Ax')_v \rangle$ from (ii) inductively. We now define Q_v to be some subset of Q'_v with $\lambda(Q_v) = 1 - x_e$. This is not possible if $\lambda(Q'_v) < 1 - x_e$ but in this case we just define $Q_v := Q'_v$. Also, even if no subset of Q'_v has λ-value *exactly* $1 - x_e$ we can split[3] a term of the ρ-convex decomposition to achieve this. The point of this Q_v is that, using (ii) inductively, the post-packing total λ-value of the solutions with degree $\lceil (Ax)_v \rceil$ at v will be at most $\lambda(Q'_v \setminus Q_v) \leq \langle (Ax')_v \rangle - (1 - x_e) = \langle (Ax)_v \rangle$; the latter equality holds since $(Ax)_v = (Ax')_v + x_e$ and by the hypotheses of this case. So these Q_v allow us to inductively satisfy (i) and (ii), on top of which $\lambda(Q_v) \leq 1 - x_e$.

In all cases, $\lambda(Q_v) \leq 1 - x_e$. Analogous to Lemma 11 there is enough room to complete the packing step so long as $\rho \geq x_e + \lambda(\bigcup_{v \in e} Q_v)$. By a union bound this would be implied by $\rho \geq x_e + k(1 - x_e)$. This gives the same analysis as before (Proposition 13) in terms of our bounds on x_e and ρ, so the modified packing step succeeds and we are done. □

4.1 Polynomial-Time Iterated Packing for k-Hypergraph b-Matching

Finally, we give our main algorithm. It uses modified packing steps and always maintains a ρ-convex combination satisfying the conditions of Lemma 15. As usual, the core algorithm HbM (Algorithm 3) operates on solutions where the incidence matrix A is of full column rank.

```
Algorithm 3 H$b$M$(V, E, x)$                    // write $x$ as $\rho$-convex comb. of 0-1 solutions
Require: $A$ has its columns linearly independent
  If $x = \mathbf{0}$ return the trivial $\rho$-convex combination $\lambda_1 = \rho, x_1 = \mathbf{0}$.
  Pick a vertex $\widehat{v}$ with minimum nonzero degree.
  Pick $e \in \delta(\widehat{v})$ such that $x_e$ is maximized.
  Recurse: $(x^i, \lambda_i)_i := $ H$b$M$(V, E \setminus \{e\}, x|_{E \setminus \{e\}})$.
  Extend each $x^i$ back to $\mathbb{R}^E$ by setting the $e$-coordinates to 0.
  Modified packing step: pack $x_e$ of $e$ into $(x^i, \lambda_i)_i$ and return the result.
```

Lemma 16. *If* $0 < x < 1$ *and the columns of the incidence matrix A are linearly independent, HbM expresses x as a ρ-convex combination of* 0-1 *solutions satisfying the same properties as Lemma 15.*

Proof. The proof is very similar to proof of Lemma 15 (except we have linear independence instead of extremeness) and we therefore re-use its notation and some of the observations therein. Our goal is to show that each modified packing step succeeds. Write Q for $\bigcup_{v \in e} Q_v$. For the modified packing step to succeed we need $\lambda(Q) + x_e \leq \rho$ as before. We will use that $\lambda(Q_v) \leq (1 - x_e)$ for each v, which holds as in Lemma 15.

The first case we will handle is $|e| < k$. In this case, $\lambda(Q) + x_e \leq |e|(1 - x_e) + x_e \leq (k-1)(1 - x_e) + x_e \leq k - 1 \leq \rho$, as needed. So we assume $|e| = k$.

Since Lemma 10 applies to our setting, the degree of $\widehat{v}$ is at most μ. The next case we will handle is $x_e \geq 1/\mu$. In this case, $\lambda(Q) + x_e \leq k(1 - x_e) + x_e = k - (k-1)x_e \leq k - (k-1)/\mu = \rho$ (like the proof of Proposition 13). So we may assume $x_e < 1/\mu$.

[3] Splitting means to replace the term (x^i, λ^i) with two terms $(x^i, p), (x^i, \lambda^i - p)$ with distributed λ-mass on the same integer solution x^i.

Likewise, by the definition of μ, we may assume $x(\delta(\widehat{v})) < 1$, since otherwise we fall in to the previous case by our choice of e.

Since $x(\delta(\widehat{v})) < 1$, we can get an exact expression for $Q_{\widehat{v}}$ more specific than that given in the proof of Lemma 15. All solutions x^i in the ρ-convex combination have degree 0 or 1 at v, and the latter are the ones in $Q_{\widehat{v}}$ (blocking e at $\widehat{v}$), and so $\lambda(Q_{\widehat{v}}) = (Ax')_{\widehat{v}} = (Ax)_{\widehat{v}} - x_e = x(\delta(\widehat{v})) - x_e$. This complements the upper bounds $\lambda(Q_v) \leq 1 - x_e$ that hold for all other $v \in e$ with $v \neq \widehat{v}$. This lets us bound the amount of room needed for the modified packing step:

$$\begin{aligned} x_e + \lambda(Q) &\leq x_e + x(\delta(\widehat{v})) - x_e + (k-1)(1-x_e) \\ &\leq \mu x_e + (k-1)(1-x_e) = k - 1 + (\mu - k + 1)x_e \\ &\leq k - 1 + (\mu - k + 1)/\mu = k - (k-1)/\mu = \rho \end{aligned}$$

where the middle inequality used $x(\delta(\widehat{v})) \leq \mu x_e$ and the last used $x_e < \frac{1}{\mu}$. □

To complete the proofs of Theorems 1 and 2, we yet again use the approach of starting with an extreme point solution and fixing its integer part (like Observation 6), recursing only on the residual b-matching problem, which has linearly independent rows and $0 < x < 1$.

5 Application: Bounded Color k-Hypergraph b-Matching

We observe that improved approximations for the bounded color k-hypergraph b-matching problem, which is defined above Corollary 3, follow directly from our results. The specialization of this problem for the case of 1-matchings in graphs was very recently introduced by Georgios [16], who gave a 2-approximation (note that Gerogios had considered only 1-matchings and not b-matchings). This independent result also leverages a variant of iterated packing. We give a k-approximation for the general case of bounded color k-hypergraph b-matching.

Georgios observed that the problem was a special case of 3-hypergraph b-matching: for each color class E_i, add a new vertex c_i with capacity w_i. Now replace each edge $\{u, v\}$ with a hyperedge $\{c_i, u, v\}$. This precisely models the bounded color 1-matching problem. An analogous reduction shows that bounded color k-hypergraph b-matching is a special case of standard $k+1$-hypergraph b-matching. To obtain our approximation, we simply observe that these special instances are bipartite, as the set U consisting of all the c_i vertices intersects every hyperedge exactly once. This gives us a k-approximation since the instance under consideration is a $k+1$-hypergraph.

6 Application: Allocations

We will take advantage of the Lavi-Swamy framework [22], 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 ρ-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 ρ. Minimizing this factor means we are coming closer to a VCG-like mechanism, whereas allocating everyone the empty set is truthful but 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. Altogether the fractional allocation LP is:

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

We assume the input to the mechanism is an explicit list from each bidder, consisting of their valuation for each set upon which they wish to put a positive bid. The number of variables and constraints in the LP is polynomial in the number of such bids. Although for constant k, any reasonable bid language or oracle can be used, since the number of sets of size $< k$ is polynomial and we can convert everything to an explicit list.

Definition 17. An *ρ-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 $\sum_i v^i_{T(i)}/\rho$ 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 18 (Lavi-Swamy [22]). *Given a polynomial-time LP-relative ρ-approximation algorithm for an allocation problem, we can obtain a polynomial-time ρ-approximate truthful-in-expectation mechanism.*

However, the allocation problem here is precisely bipartite k-hypergraph matching: for each bidder and each set of items they could win, create a set out of them all together, and this set has size at most $1 + k - 1 = k$; and each such hyperedge contains exactly one bidder, so the hypergraph is indeed bipartite. So our bipartite extension of the Chan-Lau theorem (Section 2) applies and we are done. The LP-relative property is essential; the non-LP relative local search approach from [13] cannot be used with [22].

7 Local Ratio and k-Hypergraph Demand Matching

We recommend [4, 6, 7] for background on the local ratio method, including its relationship with the primal-dual method. The heart of the local ratio approach is the following lemma:

Lemma 19 (Local ratio lemma). *Let x_{OPT} be the (unknown) optimal integral solution. If $w_i \cdot x_{LR} \geq w_i \cdot x_{OPT}$ for all i, and $w = \sum_i w_i$, then $w \cdot x_{LR} \geq w \cdot x_{OPT}$, i.e. x_{LR} is α-approximately optimal.*

Compared with fractional local ratio, we do not start by solving an LP, which is faster. But, we cannot use x^* to guide the algorithm — we have to ensure an oblivious approximation guarantee that holds against the unknown optimal solution.

In this section we reinterpret the $2k$-approximation for k-hypergraph demand matching from [24] as a local ratio algorithm. Compared with [24], the new algorithm will be both simpler and faster (as we solve no LPs). The inspiration for this simplified algorithm is a connection between local ratio algorithm and iterative packing elucidated by Bar-Yehuda et al. [5, p. 12].

As before, let A be the incidence matrix, and let $A[d]$ be the same matrix but with the column for each e having its entries multiplied by d_e. Then an ILP formulation for the hypergraph demand matching problem is to find an integral x maximizing wx subject to $A[d]x \leq b$ and $c \geq x \geq 0$.

We will assume that $d_e \leq b_v$ whenever $v \in e$. This is without loss of generality for the purposes of approximation, while for bounding the integrality gap this *no-clipping assumption* is needed to even get a constant upper bound (even if $k = 1$, a.k.a. knapsack). Additionally, we treat only the main case that c is unit, with the general case and the proof details in Appendix B; the same basic ideas were used in [24] but arranged differently.

The crux in our case is to show that for every instance, there is a hyperedge e and a weight function satisfying that any feasible solution is either $2k$-approximately optimal or has room for e to be added. With this (Lemma 20) and using the local ratio lemma, we can show that Algorithm 4 is a $2k$-approximation algorithm.

Lemma 20. *Let e be the hyperedge so that d_e is minimal. Define a weight function $\widehat{w}$ on all hyperedges by $\widehat{w}_e = 1$, and for all other f,*

$$\widehat{w}_f := \sum_{v \in e \cap f} \frac{d_f}{\max\{b_v - d_e, d_e\}}. \tag{2}$$

Then (i) every feasible solution (whether or not it contains e) has value at most $2k$ under $\widehat{w}$, (ii) $\widehat{w}_e \geq 1$, and (iii) any feasible subset of $E \setminus \{e\}$ to which e cannot be added has weight at least 1 under $\widehat{w}$.

Algorithm 4 HDM(V, E, d, b, w) // for hypergraph demand matching

```
Pick $e \in E$ such that $d_e$ is minimum, or return $\varnothing$ if $E = \varnothing$.
Define a new weight function $\widehat{w} \in \mathbb{R}^E$ via $\widehat{w}_e = 1$ and (2) for $f \neq e$.
Let $w_e\widehat{w}$ be its scalar multiple by $w_e$, and $w' := w - w_e\widehat{w}$.     // note $w'_e = 0$
Define $E' := \{e \in E \mid w'_e > 0\}$.                                               // note $e \notin E'$
Recurse: $\mathcal{F}' :=$ HDM$(V, E', d, b, w'|_{E'})$.
If $\mathcal{F}' \cup \{e\}$ is feasible define $\mathcal{F} := \mathcal{F}' \cup \{e\}$, else define $\mathcal{F} := \mathcal{F}'$.
Return $\mathcal{F}$.
```

References

[1] R. Aharoni and O. Kessler. On a possible extension of hall's theorem to bipartite hypergraphs. *Discrete Mathematics*, 84(3):309 – 313, 1990.

[2] E. M. Arkin and R. Hassin. On local search for weighted k-set packing. *Math. Oper. Res.*, 23(3):640–648, Mar. 1998.

[3] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. On k-column sparse packing programs. In *Proc. 14th IPCO*, pages 369–382, 2010.

[4] R. Bar-Yehuda, K. Bendel, A. Freund, and D. Rawitz. Local ratio: A unified framework for approximation algorithms. *ACM Comp. Surv.*, 36(4):422–463, Dec. 2004.

[5] R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006.

[6] R. Bar-Yehuda and D. Rawitz. On the equivalence between the primal-dual schema and the local ratio technique. *SIAM J. Disc. Math.*, 19(3):762–797, 2005.

[7] R. Bar-Yehuda and D. Rawitz. A tale of two methods. *Theoretical Computer Science*, pages 196–217, 2006.

[8] P. Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. In *Proc. 7th SWAT*, pages 214–219, 2000.

[9] R. D. Carr and S. Vempala. Randomized metarounding. *Random Struct. Algorithms*, 20(3):343–352, 2002.

[10] Y. Chan and L. Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Math. Program.*, pages 1–26, 2011.

[11] 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.

[12] U. Feige and M. Singh. Edge coloring and decompositions of weighted graphs. In *Proc. ESA*, pages 405–416, 2008.

[13] M. Feldman, J. Naor, R. Schwartz, and J. Ward. Improved approximations for k-exchange systems. In *Proc. 19th ESA*, pages 784–798, 2011.

[14] Z. Füredi. Maximum degree and fractional matchings in uniform hypergraphs. *Combinatorica*, 1:155–162, 1981.

[15] Z. Füredi, J. Kahn, and P. D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.

[16] S. Georgios. Approximation algorithms for bounded color matchings via convex decompositions. In *39th International Symposium on Mathematical Foundations of Computer Science (to appear)*, 2014.

[17] E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating k-set packing. *Computational Complexity*, 15(1):20–39, 2006.

[18] 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.

[19] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, 1991.

[20] C. Koufogiannakis and N. E. Young. Distributed fractional packing and maximum weighted b-matching via tail-recursive duality. In *Proc. 23rd DISC*, pages 221–238, 2009.

[21] P. Krysta. Greedy approximation via duality for packing, combinatorial auctions and routing. In *Proc. 30th MFCS*, pages 615–627, 2005.

[22] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proc. 46th FOCS*, pages 595–604, 2005.

[23] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[24] O. Parekh. Iterative packing for demand and hypergraph matching. In *Proc. 15th IPCO*, pages 349–361, 2011.

[25] Y. Person and M. Schacht. Almost all hypergraphs without fano planes are bipartite. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 217–226, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[26] A. Schrijver. *Combinatorial optimization*. Springer, New York, 2003.

[27] F. B. Shepherd and A. Vetta. The demand-matching problem. *Mathematics of Operations Research*, 32(3):563–578, August 2007.

[28] W. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954.

A Proofs for Sections 1 and 2

Proof of Observation 6. Given an arbitrary k-hypergraph b-matching instance $\max\{wx \mid x \in \mathbb{Z}, 0 \leq x \leq c, Ax \leq b\}$, let $\overline{x}$ be an optimal solution to the LP relaxation. We fix the integer part of $\lfloor \overline{x} \rfloor$ and work with a residual problem. Let $c'_e = 1$ if $\overline{x}_e$ is non-integral and $c'_e = 0$ otherwise, and let $b' := b - A\lfloor \overline{x} \rfloor$; then the residual problem is $\max\{wx \mid x \in \mathbb{Z}, 0 \leq x \leq c', Ax \leq b'\}$, which is simple (it has unit capacities as well as some zero-capacity edges that can be deleted). The LP optimum of the residual problem is $w\lfloor \overline{x} \rfloor$ less than the LP optimum of the original problem. A short calculation then shows that an (LP-relative) α-approximation for the residual problem, added to $\lfloor \overline{x} \rfloor$, is an approximation of the same quality for the original problem. □

Proof of Lemma 10. The first part is a counting argument. The incidence matrix retains its rank if we delete the all-zero rows, leaving only those rows corresponding to the set V' of vertices with nonzero degree. The number of such vertices must satisfy $|V'| \geq |E'|$ or else rank $|E'|$ could not be achieved. Since each column has at most k unit entries, there are at most $k|E'|$ unit entries in the whole matrix. So averaging, some row has at most $k|E'|/|V'| \leq k$ nonzeroes, and this gives the desired vertex.

For bipartite hypergraphs, examine the situation in which equality holds. This can only happen if $|E'| = |V'|$ and the matrix has exactly k ones per row and per column. Let U be the subset of vertices so that every hyperedge intersects U exactly once. So, each hyperedge intersects the complement of U exactly $k-1$ times. Therefore, the vector in $\mathbb{R}^V$ with $(-k-1)$ entries in U and unit entries elsewhere is orthogonal to all rows, contradicting that the adjacency matrix has full rank. □

Proof of Proposition **??**. **(i)** Note that $\widehat{w}(x)$ is the sum of the x values of all edges that intersect e (including itself). We group them by the point of intersection and isolate the v term, giving

$$\widehat{w}(x) \leq x_e + (x(\delta(v)) - x_e) + \sum_{u \in e, u \neq v} (x(\delta(u)) - x_e)) \leq x(\delta(v)) + (|e| - 1)(1 - x_e).$$

Applying $x_e \geq x(\delta(v))/\mu$ to the above inequality,

$$\widehat{w}(x) \leq |e| - 1 + x(\delta(v)) \frac{\mu - |e| + 1}{\mu}. \tag{3}$$

Now we use Lemma 10. In the bipartite case $\mu = k - 1$, (3) is at most $|e| - 1 \leq k - 1 = \rho$ as needed. In the general case where $\mu = k$, note that the right-hand side of (3) has nonnegative coefficient in $|e|$ (as $x(\delta(v)) \leq 1 \leq \mu$) so it suffices to consider $|e| = k$. Then (3) is at most $k - 1 + x(\delta(v))/\mu \leq k - 1 + \frac{1}{k} = \rho$ as needed.

(ii) This follows from (i). For the induction to work, we use the fact that a subset of full-rank columns is again full-rank. Observe that $\widehat{w}(F) \geq 1$ and so $\widehat{w}(F) \geq \widehat{w}(x)/\rho$. Using induction, $w'(F) \geq w'(x)/\rho$. Combining these facts using the local ratio lemma, we have the desired result. □

Proof of Lemma 12. If $x_e = 1$ for any coordinate then we are done, so suppose otherwise. We know from elementary linear algebra that there is a set V'' of vertices and a set E'' of edges so that x is the unique solution to $x_e = 0, \forall e \notin E''; x(\delta(v)) = 1, \forall v \in V''$. Then the same counting argument as in Lemma 10 (resp. and the same linear independence in the bipartite case) ensures that some $v \in V''$ is incident on at most k (resp. $k-1$) edges. Since it has $x(\delta(v)) = 1$ the $e \in \delta(v)$ maximizing x_e satisfies the lemma. $\square$

B Proofs for Section 7

Proof of Lemma 20. **(i)** Suppose some $X \subseteq E$ has $e \in X$. Then for every $v \in e$, the total demand of $X \setminus \{e\}$ at v must be at most $b_v - d_e$. Hence

$$\begin{aligned}\widehat{w}(X) &\le 1 + \sum_{f \in X \setminus \{e\}} \sum_{v \in e \cap f} \frac{d_f}{b_v - d_e} = 1 + \sum_{v \in e} \frac{\sum_{f \in X \setminus \{e\}, v \in f} d_f}{b_v - d_e} \\ &\le 1 + \sum_{v \in e} 1 \le |e| + 1 \le k + 1.\end{aligned}$$

On the other hand if $e \notin X$, then we use that $\max\{b_v - d_e, d_e\} \ge b_v/2$ and the feasibility of X (specifically $A[d]X \le b$) to obtain

$$\widehat{w}(X) \le \sum_{f \in X} \sum_{v \in e \cap f} \frac{d_f}{b_v/2} = 2 \sum_{v \in e} \frac{\sum_{f \in X, v \in f} d_f}{b_v} \le 2|e| \le 2k.$$

(ii) Holds by definition.

(iii) If some $X \not\ni e$ is feasible and $X \cup \{e\}$ is not feasible, let v be some vertex for which feasibility is exceeded when we add e. Thus $\sum_{f \in X, v \in f} d_f \ge b_v - d_e$. Also, $d_f \ge d_e$ for all $f \in X$ by our choice of e as having minimal demand, and X must have at least one edge incident on v by the no-clipping assumption, so

$$\widehat{w}(X) \ge \sum_{f \in X, v \in f} \frac{d_f}{\max\{b_v - d_e, d_e\}} \ge \frac{\max\{b_v - d_e, d_e\}}{\max\{b_v - d_e, d_e\}} = 1. \qquad \square$$

Then with standard local ratio arguments, we get:

Theorem 21. *Algorithm 4 is a $2k$-approximation algorithm for k-hypergraph demand matching.*

Proof. We prove the result by induction on $|E|$; the base case is trivial. For the recursive step, we know by induction that $\mathcal{F}'$ satisfies $w'(\mathcal{F}') \ge w'(X)/2k$ for any subfamily X of E'. Since w' is nonpositive on $E \setminus E'$ and $w'_e = 0$, we have that $\mathcal{F}$ is a $2k$-approximately optimal subset of E under w'.

We claim that $\mathcal{F}$ is also a $2k$-approximately optimal subset of E under $\widehat{w}$. First, if $F = F' \cup \{e\}$ then $\widehat{w}(F) \ge 1$ by Lemma 20(ii), and if $F = F'$ then $\widehat{w}(F) \ge 1$ by Lemma 20(iii). Second, the optimal subset of E under $\widehat{w}$ has weight at most $2k$, by Lemma 20(i).

Finally we use the "local ratio lemma:" as $\mathcal{F}$ is a $2k$-approximately optimal subset of E under both w' and $\widehat{w}$, so it is under their conic combination w. $\square$

Theorem 5 describes an LP-relative guarantee. To show this it is enough to prove a stronger version of Lemma 20(i) where we argue that all feasible *fractional* solutions x have $\widehat{w}(x) \le 2k$ — the details are the same as in [24, p. 356] and so we omit them.

Finally, the above explanations assumed that we have $c = 1$. Unfortunately, dealing with the general case is not as simple as using Observation 6, since the residual instance might not satisfy the no-clipping assumption. Nonetheless, we can modify the algorithm to one that runs for arbitrary c in polynomial time via the following changes. This will complete the proof of Theorem 5 in its entirety. First, we assume WOLOG that $c_e \leq \min_{v \in e} \lfloor b_v / d_e \rfloor$ for each e. When $c_e \geq 2$ we alter the definition of $\widehat{w}$ so that for $f \neq e$,

$$\widehat{w}_f := \sum_{v \in e \cap f} \frac{d_f}{\min\{(b_v - d_e)/c_e, (b_v - (c_e - 1)d_e)/2\}}.$$

The algorithm defines F by greedily adding as many copies of e as possible (up to c_e) to F'. Like Lemma 20 we can show that this choice of $\widehat{w}$ ensures both $\widehat{w}(F) \geq c_e$ and that every fractionally feasible solution x has $\widehat{w}(x) \leq 2kc_e$. The ideas used to fill in the details of this proof are the same ideas as are used in the $c = 1$ case.