How to Sell Hyperedges: The Hypermatching Assignment Problem*

Marek Cygan   Fabrizio Grandoni   Monaldo Mastrolilli†

Abstract

We are given a set of clients with budget constraints and a set of indivisible items. Each client is willing to buy one or more bundles of (at most) k items each (bundles can be seen as hyperedges in a k-hypergraph). If client i gets a bundle e, she pays $b_{i,e}$ and yields a net profit $w_{i,e}$. The *Hypermatching Assignment Problem* (HAP) is to assign a set of pairwise disjoint bundles to clients so as to maximize the total profit while respecting the budgets. This problem has various applications in production planning and budget-constrained auctions and generalizes well-studied problems in combinatorial optimization: for example the weighted (unweighted) k-hypergraph matching problem is the special case of HAP with one client having unbounded budget and general (unit) profits; the Generalized Assignment Problem (GAP) is the special case of HAP with $k = 1$.

Let $\varepsilon > 0$ denote an arbitrarily small constant. In this paper we obtain the following main results:

- We give a randomized $(k + 1 + \epsilon)$ approximation algorithm for HAP, which is based on rounding the 1-round Lasserre strengthening of a novel LP. This is one of a few approximation results based on Lasserre hierarchies and our approach might be of independent interest. We remark that for weighted k-hypergraph matching no LP nor SDP relaxation is known to have integrality gap better than $k - 1 + 1/k$ for general k [Chan and Lau, SODA'10].

- For the relevant special case that one wants to maximize the total revenue (i.e., $b_{i,e} = w_{i,e}$), we present a local search based $(k + O(\sqrt{k}))/2$ approximation algorithm for $k = O(1)$. This almost matches the best known $(k + 1 + \epsilon)/2$ approximation ratio by Berman [SWAT'00] for the (less general) weighted k-hypergraph matching problem.

- For the unweighted k-hypergraph matching problem, we present a $(k + 1 + \epsilon)/3$ approximation in quasipolynomial time. This improves over the $(k+2)/3$ approximation by Halldórsson [SODA'95] (also in quasipolynomial time). In particular this suggests that a $4/3 + \epsilon$ approximation for 3-dimensional matching might exist, whereas the currently best known polynomial-time approximation ratio is $3/2$.

1 Introduction

Consider the following two natural scenarios. We are given a set of clients, each one with a budget, and a set of bundles of items, each one with a price and a profit (which might vary from client to client). Clients wish to buy one or more disjoint bundles of items (they are not interested in buying subsets of bundles however). Our goal is to maximize the total profit, while respecting client budgets. Alternatively, we are given a set of machines and a set of products obtained by assembling together on a machine bundles of indivisible and possibly different components. A component that is used for a product cannot be used for another one. Each product has an assembling time and a profit (both depending on the machine), and the total time spent on each machine cannot exceed a specified amount. The goal is to choose which products we want to produce and assign them to the machines for assembling in order to maximize the total profit.

The mentioned scenarios can be modeled via the following *Hypermatching Assignment Problem* (HAP). We are given a k-hypergraph $G = (V, E)$ and a set of m clients (or machines) $I = \{1, \dots, m\}$. Note that k and m are not required be constant[1]. Each client i has a *budget* $B_i \geq 0$. For each client i and hyperedge e, there is a cost (or price) $b_{i,e} \geq 0$ and a weight (or profit) $w_{i,e} \geq 0$. The goal is to compute a set M of disjoint

*Partially supported by the ERC Starting Grant NEWNET 279352, by the Swiss NSF project 200020-122110/1 "Approximation Algorithms for Machine Scheduling Through Theory and Experiments III", and by Hasler Foundation Grant 11099.

†IDSIA, University of Lugano, Switzerland, {marek,fabrizio,monaldo}@idsia.ch

[1] We remark that assuming $m = O(1)$ would simplify the problem considerably.

hyperedges (hypermatching) of G, and to partition M into m subsets $M_1, \ldots, M_m$ so that the total profit $\sum_{i=1}^m w_i(M_i)$, $w_i(M_i) := \sum_{e \in M_i} w_{i,e}$, is maximized and the budget constraint $b_i(M_i) := \sum_{e \in M_i} b_{i,e} \leq B_i$ is satisfied for all clients i.

W.l.o.g., we also assume that all hyperedges have arity exactly k (i.e. G is k-uniform): this can be achieved by adding dummy nodes. For shortness, we will sometimes drop the suffix hyper in hypergraph, hyperedge, etc.

Previous Work. To the best of our knowledge, HAP was not studied before (even not for $k = 2$). However, it generalizes in a natural way two well-studied problems. When there is only one client (i.e., $m = 1$) with an unbounded budget, then HAP is the k-hypermatching problem. When $k = 1$ then HAP is the Generalized Assignment Problem (GAP).

The hypergraph matching problem, also known as the set packing problem, is a fundamental problem in combinatorial optimization with various applications. Håstad [19] proved that set packing cannot be approximated to within $O(N^{1-\varepsilon})$ unless $NP \subseteq ZPP$ (where N is the number of sets)[2]. The hypergraph matching problem in k-uniform hypergraphs is also known as the k-set packing problem and it is a generalization of some classical combinatorial optimization problems, e.g. the k-dimensional matching problem, and the maximum independent set problem in bounded degree graphs. All the best known approximation algorithms for the hypergraph matching problem in k-uniform hypergraphs are based on local search methods [2, 4, 7, 17, 21]: For the unweighted problem, Hurkens and Schrijver [21] gave a $\left(\frac{k}{2} + \varepsilon\right)$-approximation algorithm. For the weighted case, the currently best known algorithm is due to Berman [2] who gave a $\left(\frac{k+1+\varepsilon}{2}\right)$-approximation algorithm. Interestingly, a better result is possible in quasi-polynomial time, as proved by Halldórsson [17] who presented a $(k+2)/3$ approximation algorithm for the unweighted hypergraph matching problem in k-uniform hypergraphs.[3] On the other hand, Hazan, Safra and Schwartz [20] proved that it is hard to approximate within a factor of $O\left(\frac{k}{\log k}\right)$. Recently, Chan and Lau [6] analyzed different linear and semidefinite programming relaxations for the hypergraph matching problem, and studied their connections to the local search method. In their paper they show that the standard linear programming relaxation of the problem has integrality gap exactly $k - 1 + 1/k$ for k-uniform hypergraphs. For the unweighted problem, they show that there is a polynomial size semidefinite program with integrality gap at most $\frac{k+1}{2}$ (the proof is obtained in an indirect way, i.e. not by a rounding algorithm). The latter result is not known to extend to the weighted case. In [6] it is also observed that no example with integrality gap larger than $\Omega\left(\frac{k}{\log k}\right)$ (as implied by the hardness result in [20]) is known.

Regarding GAP, the problem is known [5] to be NP-hard to approximate to any factor better than 11/10. On the positive side, a 2-approximation algorithm is implicit in the work of Shmoys and Tardos [30], as observed by Chekuri and Khanna [8]. Their method is based on rounding a standard LP relaxation (with integrality gap 2). The approximation factor has been improved to $\frac{e-1}{e}$ by Fleischer, Goemans, Mirrokni and Sviridenko [14]. Their approach is based on rounding a provably stronger LP, called *configuration LP*, with an exponential number of variables. The configuration LP is a relaxation of the problem where, for each client i, there is a variable for each one of the (exponentially many) possible feasible assignments to that client (configuration). This LP can be solved up to an arbitrary precision, since the separation problem of the dual is a knapsack problem. Any (fractional) solution of the configuration LP can be seen as a probability distribution over the set of all feasible assignments. By independently sampling with this distribution, they obtain a *tentative assignment* for each client which satisfies the corresponding budget constraint. There might be conflicts between clients, namely the same item might be assigned to different clients. These conflicts are resolved via a randomized procedure. Recently, Feige and Vondrák [13] improved the approximation factor for GAP to $\frac{e-1}{e} - \delta$ for a proper (small) constant $\delta > 0$. The algorithm is based on the same configuration LP as in [14], but the rounding technique is different.

Our Results. In this paper we obtain the following main results:

(1) General case. We present a randomized $(k+1+\varepsilon)$-approximation algorithm for HAP (see Section 2). A natural approach for the addressed problem would be to extend the techniques in [13, 14] for GAP, by considering the configuration LP for HAP. However, for GAP the configuration LP can be solved in polynomial-time up to an arbitrary precision, but for HAP we cannot hope for the same since this would imply a PTAS for the HAP with a single client, which is a generalization of the k-uniform hypergraph matching problem.

In this paper we take an alternative approach, which might be of independent interest. We design a novel *partial configuration* LP, of intermediate complexity w.r.t.

[2]Throughout this paper $\varepsilon > 0$ denotes an arbitrarily small constant.

[3]The quasi-polynomial time $(k+1+\varepsilon)/3$-approximation algorithm claimed in [17] is incorrect [18].

the standard LP and configuration LP for HAP: our LP contains a polynomial number of variables (but exponential in $1/\varepsilon$) which model the poly$(1/\varepsilon)$ most profitable edges assigned to each machine in the optimum solution[4]. We strengthen this LP by 1-round of the Lasserre's lift-and-project method and solve it in polynomial-time (within an arbitrarily small additive error) [15]. Then we exploit a non-trivial randomized rounding procedure to obtain an independent tentative assignment for each client of expected weight arbitrary close to the weight of the fractional solution of the Lasserre relaxation. Here we crucially exploit the *Decomposition Lemma* in [22] to turn fractional assignments of the most profitable edges into integral ones. The computed tentative assignments respect budget constraints. However, there might be conflicts (due to nodes assigned multiple times) among clients and even within the same client (differently from [14]). Intuitively, our sampling procedure produces *relaxed configurations*, where conflicts among the least profitable edges are allowed. Nonetheless, we are still able to solve such conflicts with a simple randomized procedure. Note that the resulting approximation factor is w.r.t. the (almost) optimal solution to the Lasserre relaxation. We remark that for weighted k-hypermatching problem no LP nor SDP relaxation is known [6] to have integrality gap better than $k - 1 + 1/k$ for general k. We believe that our combination of a partial configuration LP with Lasserre hierarchies might turn out to be useful in other contexts as well, for example where configuration LPs cannot be approximated well enough.

(2) Uniform case. We consider the *uniform* version of HAP, where $b_{i,e} = w_{i,e}$ for all i and e (see Section 3). This models the relevant special case where one wants to maximize the total revenue (i.e. sum of the prices) rather than the total profit. For example, this might be the case if profits are (roughly) proportional to prices. Alternatively, fixed costs of the seller might largely prevail on bundle costs.

For this problem we present a local search based $(k + O(\sqrt{k}))/2$ approximation algorithm for $k = O(1)$. This almost matches the best known $(k + 1 + \varepsilon)/2$ approximation ratio by Berman [2] for the (less general) weighted k-hypermatching problem (also based on local search and for $k = O(1)$). The main extra difficulty that we need to address w.r.t. to the hypermatching case is that improving the total profit with local changes is not sufficient for us, we need also to satisfy budget constraints on each machine at any time. To this aim, we introduce dummy clients which inherit some fraction of the budget of the associated (original) client, and carefully move edges from dummy clients to original ones and back.

(3) Unweighted hypermatching. Finally, we consider the standard unweighted k-hypermatching problem (see Section 4). We present a $(k + 1 + \varepsilon)/3$ approximation running in quasi-polynomial time. This improves over the $(k + 2)/3$ approximation by Halldórsson [17] (also in quasi-polynomial time). In particular this suggests that a $4/3 + \varepsilon$ approximation for 3-dimensional matching might exist, whereas the currently best known polynomial-time approximation ratio is 3/2. Our result exploits the fact any (sufficiently dense) graph $G = (V, E)$, namely $|E| \geq (1 + \varepsilon)|V|$ for some constant $\varepsilon > 0$, contains a cycle with $O(\log n)$ nodes. Based on that and on a non-trivial construction, we prove that if it is not possible to improve the current hypermatching by adding $O(\log n)$ new edges (and removing the conflicting edges in the current solution), then the current solution is "good enough".

Related Work. A lot of research was devoted in the last two decades to the strengthening of linear and semidefinite programming relaxations by local constraints, e.g. Lovász-Schrijver hierarchy [24], Sherali-Adams hierarchy [29], Lasserre hierarchy [23]. These lift-and-project hierarchies are considered to be a strong computational model which captures many known algorithms. The computation of these hierarchies proceeds in rounds. Intuitively, after r rounds the solutions are required to be "locally" integral on all neighborhoods of up to r vertices but require time $n^{O(r)}$ to run. After n rounds, they are guaranteed to produce the exact optimal solution. If local integrality implies global integrality, then these hierarchies will provide a good approximation after only a few rounds. Indeed, these hierarchies provably imply some of the most celebrated approximation algorithms for NP-complete problems even after a few rounds. These hierarchies give us a framework to study the power and limitations of linear and semidefinite programming by asking what the tradeoff between the running time and the guaranteed approximation in these hierarchies is.

Unfortunately, most results along these lines have been negative, showing that even relaxations at superconstant (and sometimes even linear) levels of certain hierarchies hardly yield any reduction in the integrality gap (see e.g. [11, 28]). In contrast, new results where improved approximation guarantees arise from the first $O(1)$ levels of a hierarchy, have recently begun to emerge [12, 25, 1, 9, 10, 16, 27]. For a more detailed overview on the use of hierarchies in approximation algorithms, see the recent survey of Chlamtáč and Tulsiani [11]. Finally, we mention the work of Kar-

[4] Note that we cannot "guess" such edges in polynomial-time, being m unbounded.

lin, Nguyen and Mathieu on knapsack [22], that shows that after r^2 rounds of Lasserre, the integrality gap decreases from 2 to $r/(r-1)$, implying as a side product of their analysis a PTAS for knapsack. In particular the *Decomposition Lemma* in [22] plays an important role in this paper.

2 A Lasserre-based Approximation for HAP

In this section we present the claimed $k+1+\varepsilon$ approximation for HAP. We start by giving a high-level description of our algorithm. We consider a proper LP relaxation LP_{HAP} of HAP. In this relaxation we crucially introduce variables which describe the $1/\beta$ most profitable edges for each client, for a proper constant $\beta > 0$ which depends on ε. We compute the 1-round Lasserre strengthening $LASS_{HAP}$ of LP_{HAP}, and solve it modulo a small additive mistake. Next we independently sample a tentative assignment for each client, via a non-trivial sampling procedure. Finally, we resolve the conflicts between clients (due to nodes assigned multiple times) with a simple randomized procedure.

By adding dummy edges of cost and weight zero, we can assume w.l.o.g. that each client is assigned at least $1/\beta$ edges. Consider LP relaxation for HAP given in Fig. 1, denoted as LP_{HAP}. The set $\mathcal{F}_i$ consists of all the subsets of $1/\beta$ disjoint edges F with $b_i(F) \le B_i$. Consider any integral (in particular, 0-1) solution to LP_{HAP}. Then $y_{i,F} = 1$ iff F is the set of the $1/\beta$ edges with the largest weights assigned to client i. Moreover, $x_{i,e} = 1$ iff edge e is assigned to client i. Constraint (2.1) forces the budgets to be respected. Constraint (2.2) guarantees that the solution is a matching. Constraint (2.3) forces one choice of a set in $\mathcal{F}$ for each i. Constraint (2.4) guarantees that, for a given i, the choice of F is consistent with the assignment of the other edges not in F (namely, the edges not in F with weight larger than the minimum weight in F cannot be assigned to i).

Let $LASS_{HAP}$ be the 1-round Lasserre strengthening of LP_{HAP}. We will exploit the following *Decomposition Lemma* proved in [22].

Lemma 2.1. **(Decomposition Lemma)** *[22] Consider any linear program LP with feasible solutions in $[0,1]^n$. Let $LASS_r$ be the r-round Lasserre strengthening of LP, and $z = (z_1, \ldots, z_n)$ be the projection of a feasible solution to $LASS_r$ into the space of the variables of LP. Assume that there is a subset of variables Y of LP such that in any feasible solution to LP there are at most r variables of value 1 in Y. Then there is a polynomial time algorithm which decomposes z into a convex combination $z^1, \ldots, z^q$ of feasible solutions to LP, such that each z^i has 0-1 values on variables Y.*

Our algorithm first solves $LASS_{HAP}$; This can be done in polynomial time to an arbitrary prescribed precision using the ellipsoid method (see [15]). Let $z^* = (x^*, y^*)$ be the projection of this solution into the space of variables of LP_{HAP}. Then it independently samples a *tentative assignment* M_i for each client i as follows. Consider the variables $Y = \{y_{i,F}\}_{F \in \mathcal{F}_i}$. Observe that, due to Constraint (2.3), at most one variable Y can take value 1 in any feasible fractional solution to LP_{HAP}. We can therefore apply Lemma 2.1 to solution x and variables Y with $r = 1$, hence obtaining a set of feasible solutions $z^1, \ldots, z^q$ to LP_{HAP}, such that each z^j has integer values on variables Y. Furthermore, $z^* = \sum_{j=1}^{q} \alpha^j \cdot z^j$ for proper coefficients $\alpha^j \ge 0$, $\sum_{j=1}^{q} \alpha^j = 1$.

Let $\beta \in (0,1)$ and $\gamma \in (0,1)$ be constant parameters, depending on ε, to be defined later. We compute M_i as follows:

1. **(Sampling)** Sample a solution $z' = (x', y')$ among the z^j's according to the probability distribution induced by the coefficients α^j. Let F_i be such that $y'_{i,F_i} = 1$.

2. **(Rounding)** Add F_i to M_i. For each *small* edge e such that $b_{i,e} \le \gamma \cdot B'_i$ and $w_{i,e} \le \gamma \cdot W'_i$, with $B'_i := \sum_{e \in E} b_{i,e} \cdot x'_{i,e}$ and $W'_i := \sum_{e \in E} w_{i,e} \cdot x'_{i,e}$, add e to M_i with probability $x'_{i,e}$.

3. **(Greedy)** Remove edges in $M_i - F_i$ in a greedy way (according to ratio of weight to cost) until $b(M_i) \le B_i$.

Note that, by Constraint (2.3), in each z^j there is a unique F with $y^j_{i,F} = 1$ (and all other $y^j_{i,F} = 0$). Hence the sampling step is well defined. We also remark that after the sampling step, if $e \notin F_i$ and $x'_{i,e} > 0$, then $w_{i,e} \le \min_{f \in F_i}\{w_{i,f}\} \le \beta \cdot w(F_i)$.

Observe that M_i satisfies the budget constraint of client i by construction. We next show that in expectation its weight $w(M_i)$ is close to the fractional weight $w^i_{lass} := \sum_{F \in \mathcal{F}} w_i(F) \cdot y^*_{i,F} + \sum_{e \in E} w_{i,e} \cdot x^*_{i,e}$ associated to the same client in the (almost) optimal fractional solution z^*.

Lemma 2.2. *For any $\varepsilon > 0$, there is a choice of the parameters β and γ of the algorithm so that $E[w(M_i)] \ge (1-\varepsilon) \cdot w^i_{lass}$ for every client i.*

Proof. In order to simplify the analysis, we consider a variant of the algorithm which produces solutions of expected profit not larger than $E[w(M_i)]$. In the rounding step, we also consider *large* edges e with $b_{i,e} > \gamma \cdot B'_i$ or $w_{i,e} > \gamma \cdot W'_i$. We compute the minimum integer $h(e) > 0$ so that $b_{i,e}/h \le \gamma \cdot B'_i$ and $w_{i,e}/h(e) \le \gamma \cdot W'_i$, create $h(e)$ copies $e_1, \ldots, e_{h(e)}$ of e of cost $b_{i,e}/h(e)$ and weight $w_{i,e}/h(e)$ each, and add each copy e_i to

$$\max \sum_{i\in I, F\in \mathcal{F}_i} w_i(F)\cdot y_{i,F} + \sum_{i\in I, e\in E} w_{i,e}\cdot x_{i,e}$$

$$\text{(2.1)} \qquad s.t. \sum_{F\in\mathcal{F}_i} b_i(F)\cdot y_{i,F} + \sum_{e\in E} b_{i,e}\cdot x_{i,e} \le B_i, \qquad \forall i\in I;$$

$$\text{(2.2)} \qquad \sum_{i\in I, F\in\mathcal{F}_i: v\in e\in F} y_{i,F} + \sum_{i\in I, e\in E: v\in e} x_{i,e} \le 1, \qquad \forall v\in V;$$

$$\text{(2.3)} \qquad \sum_{F\in\mathcal{F}_i} y_{i,F} = 1, \qquad \forall i\in I;$$

$$\text{(2.4)} \qquad x_{i,e} \le 1 - y_{i,F}, \qquad \forall F\in\mathcal{F}, e\in E : e\in F \vee w_{i,e} > \min_{f\in F}\{w_{i,f}\};$$

$$y_{i,F}, x_{i,e} \ge 0, \qquad \forall F\in\mathcal{F}, e\in E, i\in I.$$

Figure 1: LP relaxation for HAP, denoted as LP_{HAP}.

M_i independently with probability $x'_{i,e}$. Then after the greedy step we remove all the copies of large edges. In both algorithms large edges do not contribute to the profit. However, in the modified algorithm each sampled small edge has a higher probability to be deleted during the greedy step.

Let us then focus on the modified algorithm. Observe that there are two types of random variables: variable z' in the sampling step and the variables associated to edge sampling in the (modified) rounding step. In order to lighten the notation, we next implicitly condition on the value of z' (and consequently on random variables F_i, B'_i and W'_i). Let M'_i, M''_i and M'''_i be the value of M_i after the rounding step, after the greedy step, and after the removal of large edges (i.e., at the end of the process), respectively. Let L_i be the set of large edges, and L'_i be the corresponding copies. We observe that $\sum_{e\in L_i} x'_{i,e} \le \frac{2}{\gamma}$, and consequently $\sum_{e\in L_i} w'_{i,e}x'_{i,e} \le \frac{2}{\gamma}\cdot\beta\, w(F_i)$. In fact, otherwise one would have either $B'_i \ge \sum_{e\in L_i} b_{i,e}\cdot x'_{i,e} > \frac{1}{\gamma}\cdot\gamma B'_i$ or $W'_i \ge \sum_{e\in L_i} w_{i,e}\cdot x'_{i,e} > \frac{1}{\gamma}\cdot\gamma W'_i$, a contradiction. Then

$$\begin{aligned} E[w(M'''_i)] &\ge E[w(M''_i)] - E[\sum_{e_j\in L'_i} w_{i,e_j}] \\ &= E[w(M''_i)] - \sum_{e\in L_i} h(e)\cdot\frac{w_{i,e}}{h(e)}\cdot x'_{i,e} \\ &\ge E[w(M''_i)] - \frac{2\beta}{\gamma}\cdot w(F_i). \end{aligned} \tag{2.5}$$

In the greedy step, if $b(M'_i) > B_i$, we lose in the worst case a factor $\frac{b(M'_i-F_i)}{B_i-b(F_i)} \le \frac{b(M'_i-F_i)}{B'_i}$ of $w(M'_i - F_i)$ plus integrally the weight of one edge (the latter weight is upper bounded by $\beta\cdot w(F_i)$):

$$\begin{aligned} E[w(M''_i)] &\ge w(F_i) - \beta\, w(F_i) \\ &+ E[\min\{1, \frac{B'_i}{b(M'_i-F_i)}\}\cdot w(M'_i - F_i)] \end{aligned} \tag{2.6}$$

Consider the event $\mathcal{A} := \{b(M'_i - F_i) \le (1+\delta)B'_i \wedge w(M'_i - F_i) \ge (1-\delta)W'_i\}$, for a proper constant $\delta\in(0,1)$ to be fixed later. Observe that:

$$E[\min\{1, \frac{B'_i}{b(M'_i-F_i)}\}\cdot w(M'_i - F_i)] \ge Pr[\mathcal{A}]\cdot\frac{1-\delta}{1+\delta}W'_i. \tag{2.7}$$

Observe also that $E[b(M'_i - F_i)] = B'_i$ and $b(M'_i - F_i)$ is the sum of independent random variables, each one with value in $[0, \gamma\cdot B'_i]$. Symmetrically, $E[w(M'_i - F_i)] = W'_i$ and $w(M'_i - F_i)$ is the sum of independent random variables, each one with value in $[0, \gamma\cdot W'_i]$. Therefore by Chernoff's bound and the union bound we obtain

$$Pr[\mathcal{A}] \ge 1 - e^{-\frac{\delta^2 B'_i}{3\gamma B'_i}} - e^{-\frac{\delta^2 W'_i}{2\gamma W'_i}} \ge 1 - 2e^{-\frac{\delta^2}{3\gamma}}. \tag{2.8}$$

Putting everything together:

$$\begin{aligned} E[w(M'''_i)] &\overset{(2.5)}{\ge} E[w(M''_i)] - \frac{2\beta}{\gamma}w(F_i) \\ &\overset{(2.6)}{\ge} w(F_i) + E[\frac{B'_i}{b(M'_i-F_i)}\cdot w(M'_i - F_i)] \\ &- \beta w(F_i) - \frac{2\beta}{\gamma}w(F_i) \\ &\overset{(2.7)}{\ge} (1 - \frac{3\beta}{\gamma})w(F_i) + Pr[\mathcal{A}]\cdot\frac{1-\delta}{1+\delta}W'_i \\ &\overset{(2.8)}{\ge} (1 - \frac{3\beta}{\gamma} - \frac{(1-2e^{-\frac{\delta^2}{3\gamma}})(1-\delta)}{1+\delta}) \\ &\cdot (w(F_i) + W'_i). \end{aligned}$$

Removing the conditioning on z', $E[w(F_i) + W'_i] = w^i_{lass}$. The claim follows by choosing $\delta = \sqrt{3\gamma \ln \frac{1}{\gamma}}$, $\gamma = \beta^{2/3}$, and $\beta = \Theta(\varepsilon^3)$.

From Lemma 2.2, the collection of preliminary assignments altogether as an expected profit of at least $(1-\varepsilon)$ times the optimal value of $LASS_{HAP}$. However it might happen that a node v is assigned to several clients (or even multiple times to the same client through different edges). In order to solve these conflicts, we exploit a simple randomized strategy. We sort the pairs $(i, e) \in \{1, \dots, m\} \times E$ uniformly at random. Given $e \in M_i$ and $f \in M_j$ with $(i, e) \neq (j, f)$ and $v \in e \cap f$ (possibly $i = j$), we remove f from M_j if (j, f) follows (i, e) in the random ordering, and otherwise we remove e from M_i. It is clear that after this step we obtain a feasible assignment (i.e., assigned edges are node disjoint and budget constraints are satisfied). Next lemma shows that we do not lose too much in terms of profit. Let $\tilde{M}_i$ be the edges assigned to i after the conflict resolution phase.

Lemma 2.3. *For every $e \in M_i$, $e \in \tilde{M}_i$ with probability at least $\frac{1}{k+1}$.*

Proof. Let $C_{i,e,v}$ be the random set of pairs (j, f), $f \in M_j$ and $(j, f) \neq (i, e)$, conflicting with (i, e) on node v (i.e., $v \in e \cap f$). From Jensen's inequality

$$\begin{aligned} Pr[e \in \tilde{M}_i | e \in M_i] &= E[\frac{1}{1 + |\cup_{v \in e} C_{i,e,v}|}] \\ &\geq \frac{1}{1 + E[|\cup_{v \in e} C_{i,e,v}|]} \\ &\geq \frac{1}{1 + \sum_{v \in e} E[|C_{i,e,v}|]}. \end{aligned}$$

Therefore it is sufficient to show that $E[|C_{i,e,v}|] \leq 1$. Note that $E[|C_{i,e,v}|]$ is upper bounded by the expected number of edges $f \in \cup_j M_j$ (counting multiplicities) which contain node v. In turn, the latter quantity is equal to the sum over $j \in I$ and $f \in E$, $v \in f$, of the probability $p_{j,f}$ that $f \in M_j$. The probability $p_{j,f}$ is upper bounded by the probability $p'_{j,f}$ that $f \in M_j$ after the rounding step (since the greedy step can only remove edges). As before, let us condition implicitly on variable z'. Then one has $p'_{j,f} = 1$ if $f \in F_j$, $p'_{j,f} = 0 \leq x'_{j,f}$ if $f \notin F_j$ and f is large w.r.t. F_j, and $p'_{j,f} = x'_{j,f}$ otherwise. Then one obtains

$$E[|C_{i,e,v}|] \leq \sum_{j \in I, v \in f} p'_{j,f} \leq \sum_{j \in I, v \in f \in F_j} 1 + \sum_{j \in I, v \in f \in E} x'_{j,f}.$$

Removing the conditioning on z', and by Constraint (2.2),

$$E[\sum_{j \in I, v \in f \in F_j} 1 + \sum_{j \in I, v \in f \in E} x'_{j,f}] = \sum_{j \in I, v \in f \in F \in \mathcal{F}_j} y^*_{j,F} + \sum_{j \in I, v \in f \in E} x^*_{j,f} \leq 1.$$

Theorem 2.1. *For any constant $\varepsilon > 0$, there is a polynomial time algorithm for HAP with expected approximation factor $k + 1 + \varepsilon$ with respect to the optimal fractional solution to $LASS_{HAP}$.*

Proof. Trivially from Lemmas 2.2 and 2.3.

3 An Improved Approximation for Uniform HAP

In this section we present an improved approximation algorithm for the uniform case, where $b_{i,e} = w_{i,e}$ for all clients i and edges e. Here we assume $k = O(1)$. We remark that we do not force clients to have the same valuations, that is potentially $b_{i_1,e} \neq b_{i_2,e}$, for $i_1 \neq i_2$. Analogously to [2], we exploit a local search strategy which tries to minimize the sum of squared weights. We will use the following result, implicit in [2], which relates the sum of squared weights to a maximum weight independent set in a k-claw free graph.

Lemma 3.1. *[2] Let $G = (V, E)$ be an undirected $(k+1)$-claw free graph with weight function $w : V \to \mathbb{R}_+$, and let w^*_{MIS} be the maximum weight of an independent set in G. Consider any independent set $S \subseteq V$ such that, for any independent set $S' \subseteq V \setminus S$ of size at most $k + 1$, one has $\sum_{v \in S} w^2(v) \geq \sum_{v \in S \setminus N(S') \cup S'} w^2(v)$. Then $\frac{k+1}{2} \sum_{v \in S} w(v) \geq w^*_{MIS}$.*

Without loss of generality, we can assume that all the weights are positive integers in $[1, W]$. We start by describing a pseudo-polynomial time algorithm, i.e. Algorithm 1 in the figure.

Here $\alpha = \alpha(k)$ is a proper parameter to be fixed later. During the course of the algorithm, we maintain a hypermatching $M \subseteq E$, which is partitioned into $2m$ sets $M_1, \dots, M_{2m}$. Intuitively the additional sets $M_{m+1}, \dots M_{2m}$ correspond to dummy clients, that we exploit to deal with budget constraints. For each $1 \leq i \leq m$ the valuations for client $i + m$ are the same as for clients i, whereas the budget for client $i + m$ is αB_i, i.e. we implicitly assume $w_{i+m,e} = w_{i,e}$ and $B_{i+m} = \alpha B_i$. By $\text{comp}(A, B)$ we denote the subset of edges of A which are compatible (i.e. have empty intersection) with each edge of B. Moreover we let $w_i(A) = \sum_{e \in A} w_{i,e}$ and $w^2_i(A) = \sum_{e \in A} w^2_{i,e}$.

Let $M^{\text{opt}} \subseteq E$ be an optimum solution and consider a partition of M^{opt} into $M^{\text{opt}}_{\text{empty}}$ and $M^{\text{opt}}_{\text{full}}$, where

Algorithm 1 Pseudo-polynomial algorithm for uniform HAP.

(A) Initialize $M_i^A := \emptyset$ for each $1 \leq i \leq 2m$. Apply exhaustively the following rules:

(1) If for some $1 \leq i \leq m$ we have $M_i^A \neq \emptyset$ and $w_i(M_i^A \cup M_{i+m}^A) \leq \alpha B_i$, then $M_{i+m}^A := M_i^A \cup M_{i+m}^A$ and $M_i^A := \emptyset$.

(2) If there exists a matching $E' \subseteq E$ of at most $k+1$ edges together with a partition of E' into $2m$ sets $E_1, \ldots, E_{2m}$, such that:

$$(i)\ \forall 1 \leq i \leq 2m:\ w_i(M_i^A \cup E_i) \leq B_i \qquad (ii) \sum_{1\leq i\leq 2m} w_i^2(\text{comp}(M_i^A, E') \cup E_i) > \sum_{1\leq i\leq 2m} w_i^2(M_i^A),$$

then for each $i = 1, \ldots, 2m$ do $M_i^A := \text{comp}(M_i^A, E') \cup E_i$.

(B) For each $i = 1, \ldots, m$ if $w(M_i^A) + w(M_{i+m}^A) \leq B_i$ then $M_i^B := M_i^A \cup M_{i+m}^A$, otherwise $M_i^B := M_i^A$. Output the matching $\bigcup_{1\leq i\leq m} M_i^B$.

---

an edge $e \in M^{\text{opt}}$ assigned to client i in the optimum solution belongs to $M_{\text{empty}}^{\text{opt}}$ iff $M_i^A = \emptyset$. Moreover let $\text{OPT}_{\text{empty}}$ and OPT_{full} be the total profit of edges of $M_{\text{empty}}^{\text{opt}}$ and $M_{\text{full}}^{\text{opt}}$, respectively. Finally let $M_1^{\text{opt}}, \ldots, M_m^{\text{opt}}$ be the partition of $M_{\text{empty}}^{\text{opt}}$ corresponding to the assignment of edges of $M_{\text{empty}}^{\text{opt}}$ to clients in the optimum solution. We need the following technical lemma.

Lemma 3.2. $\sum_{1\leq i\leq 2m} w_i(M_i^A) \geq \frac{2\text{OPT}_{\text{empty}}}{k+1}$.

Proof. Create the following auxiliary undirected simple graph H. As the set of vertices of H take $V_H = \{v_{i,e} : 1 \leq i \leq m, e \in (M_i^{\text{opt}} \cup M_i^A \cup M_{i+m}^A)\}$, that is we have a vertex for each client i and for each edge e used either in $M_{\text{empty}}^{\text{opt}}$ or in the solution obtained at the end of phase (A). As a weight function we take $w(v_{i,e}) = w_{i,e}$. Two vertices v_{i_1,e_1}, v_{i_2,e_2} of H are adjacent iff $e_1 \cap e_2 \neq \emptyset$. Recall that the intersection graphs created from set systems where sets are of cardinality k are $(k+1)$-claw free.

Let $S = \{v_{i,e} \in V_H : e \in (M_i^A \cup M_{i+m}^A)\}$ and observe that S is an independent set in H. Assume that there exists a set $S' \subseteq V_H \setminus S$ of size at most $k+1$, such that the sum of squares of weights in $S \setminus N_H(S') \cup S'$ is greater than in S. Since $V_H \setminus S = \{v_{i,e} : 1 \leq i \leq m, e \in M_i^{\text{opt}}\}$ and $M_i^A = \emptyset$ for nonempty M_i^{opt}, Rule (2) is applicable, because the assumption $M_i^A = \emptyset$ ensures that the budget B_i is not exceeded. Therefore we have a contradiction and such a set S' does not exist. Hence by Lemma 3.1 we infer that $\sum_{1\leq i\leq 2m} w_i(M_i^A) = \sum_{v_{i,e}\in S} w_{i,e} \geq \frac{2\text{OPT}_{\text{empty}}}{k+1}$, since $\{v_{i,e} \in V_H : 1 \leq i \leq m, e \in M_i^{\text{opt}}\}$ is an independent set in H.

Lemma 3.3. *Algorithm 1 is a* $(\frac{k+1}{2} + \sqrt{\frac{k}{2} + \frac{3}{2}} + 1)$-*approximation algorithm for the uniform HAP problem with integral weights from* $[1, W]$, *with running time* $O(W^2 k|E|^2(|E|m)^{k+1})$.

Proof. Rule (2) cannot be applied more than $W^2|E|$ times, and Rule (1) can be applied at most m times between two applications of Rule (2), so the total running time of the algorithm is bounded by $O(W^2|E|(2m|E|)^{k+1}(k|E|)) = O(W^2k|E|^2(2m|E|)^{k+1})$.

For $1 \leq i \leq m$ let a_i be the sum of weights of edges assigned to client i after Phase (B), i.e., $a_i = w_i(M_i^B)$, and let $b_i = \frac{k+1}{2} w_i(M_i^A \cup M_{i+m}^A)$. Note that by Lemma 3.2 we have $\sum_{1\leq i\leq m} b_i \geq \text{OPT}_{\text{empty}}$. For $1 \leq i \leq m$ let c_i be equal to 0 if $M_i^A = \emptyset$ and $c_i = B_i$ otherwise. Note that by the definition of OPT_{full} we have $\sum_{1\leq i\leq m} c_i \geq \text{OPT}_{\text{full}}$. Since $\sum_{1\leq i\leq m}(b_i + c_i) \geq \text{OPT}$, in order to prove the desired bound on the approximation ratio of the algorithm it is sufficient to show that for each $1 \leq i \leq m$ we have $\frac{b_i+c_i}{a_i} \leq \frac{k+1}{2} + \sqrt{\frac{k}{2} + \frac{3}{2}} + 1$. For $1 \leq i \leq m$ let us consider the following cases:

- $M_i^A = \emptyset$. We have $M_i^B = M_{i+m}^A$, $b_i = \frac{k+1}{2} w_i(M_{i+m}^A)$ and $c_i = 0$, hence $\frac{b_i+c_i}{a_i} = \frac{k+1}{2}$.
- $M_i^A \neq \emptyset$ and $w_i(M_i^A \cup M_{i+m}^A) \leq B_i$. Observe that since $M_i^A \neq \emptyset$, because of Rule (1) of Phase (A) we have $a_i > \alpha B_i$. Moreover $b_i = \frac{k+1}{2} a_i$, and consequently $\frac{b_i+c_i}{a_i} < \frac{k+1}{2} + \frac{1}{\alpha}$.
- $M_i^A \neq \emptyset$ and $w_i(M_i^A \cup M_{i+m}^A) > B_i$. Note that $a_i > (1-\alpha)B_i$, since otherwise edges of M_i^A and M_{i+m}^A would not exceed the budget B_i. Moreover $b_i = \frac{k+1}{2}(a_i + w_i(M_{i+m}^A)) \leq \frac{k+1}{2}(a_i + \alpha B_i)$. In this

case we infer that $\frac{b_i+c_i}{a_i} \le \frac{k+1}{2} + \frac{(k+1)\alpha}{2(1-\alpha)} + \frac{1}{1-\alpha} = \frac{k+1}{2} + \frac{(k+1)\alpha+2}{2(1-\alpha)}$.

Imposing $\frac{(k+1)\alpha+2}{2(1-\alpha)} = \frac{1}{\alpha}$ we obtain $\alpha = \frac{\sqrt{2k+6}-2}{k+1}$ and hence the claimed approximation ratio.

THEOREM 3.1. *For any $\varepsilon > 0$, there exists a $(\frac{k+1}{2} + \sqrt{\frac{k}{2}+\frac{3}{2}}+1+\varepsilon)$ approximation algorithm for HAP with running time $O(k^3|E|^5(|E|m)^{k+1}/\varepsilon^2)$.*

Proof. First, if for some edge $e \in E$ and $1 \le i \le m$ we have $w_{i,e} > B_i$, then we set $w_{i,e} := 0$, since this edge is not going to be used in any feasible solution. Denote $W = \max_{e\in E, 1\le i\le m}\{w_{i,e}\}$ and set new weights according to the formula $w'_{i,e} := \lfloor \frac{k|E|w_{i,e}}{\varepsilon W} \rfloor$. Let $E_0 \subseteq E$ and $E_1 \subseteq E$ be optimum solutions w.r.t. the original weights $w_{i,e}$ and the modified weights $w'_{i,e}$, respectively. Observe that $w(E_0) \ge W$ because one edge always forms a feasible solution. Then $w(E_1)(1+\frac{\varepsilon}{k}) \ge w(E_0)$. Moreover, the new weights are positive integers with polynomially bounded values. Therefore we can use Lemma 3.3 and the claim follows.

4 A Quasi-Polynomial Time Hypermatching Algorithm

In this section we present a $(k+1+\varepsilon)/3$ approximation for maximum (cardinality) independent set in $(k+1)$-claw free graphs $G=(V,E)$[5]. This immediately implies the claimed approximation for k-hypermatching. We next assume $k \ge 4$, since in 3-claw free graph one can find the largest independent set in polynomial time [26].

LEMMA 4.1. (LEMMA 3.2 OF [3]) *For any integer $h \ge 1$ any undirected multigraph graph $G=(V,E)$ with $|E| \ge \frac{h+1}{h}|V|$ contains a set X of at most $4h\log_2 n$ vertices, such that in $G[X]$ there are more edges than vertices.*

LEMMA 4.2. *Consider any fixed $\varepsilon > 0$ and $k \ge 4$. For any undirected $(k+1)$-claw free graph $G=(V,E)$ and two independent sets $A, B \subseteq V$, such that for any independent set $X \subseteq V \setminus A$ of size at most $9^{1+1/\varepsilon}(4+4/\varepsilon)\log_2 n$ we have $|A \setminus N(X) \cup X| \le |A|$, then*

$$(\frac{k+1}{3}+\varepsilon)|A| \ge |B|.$$

Proof. Consider an auxiliary undirected bipartite graph $H=(A\cup B, E_H)$, where two vertices $a \in A$, $b \in B$ are adjacent, i.e. $ab \in E_H$, iff $a \in N_G[b]$. Note that H is not a subgraph of G, since a copy of a vertex of G might appear on both sides of the bipartite graph H, and then such two copies are adjacent (this is the reason why we use closed neighborhood in the definition of edges of E_H). Since both A and B are independent sets in G and G is k-claw free, we infer that the maximum degree in H is at most k and therefore $|E_H| \le k|A|$.

We are going to create a sequence of induced subgraphs $H_i := H[A_i \cup B_i]$ of H with an Invariant $(*)$ that in H_i there is no subset $X \subseteq B_i$ of size at most $9^{1+1/\varepsilon - i}(4+4/\varepsilon)\log_2 n$ such that $|N_{H_i}(X)| < |X|$. Initially take $A_0 := A$, $B_0 := B$ and $H_0 := H[A_0 \cup B_0]$. Clearly for $i=0$ Invariant $(*)$ is satisfied. For any i we call a set $Y \subseteq B_i$ *improving* iff $|N_{H_i}(Y)| < |Y|$. Consider subsequent integral values of i, starting from $i=0$. Let B_i^1 be the subset of vertices of B_i, which have degree exactly one in H_i. Let $A_i^1 := N_{H_i}(B_i^1)$. Observe that $|A_i^1| = |B_i^1|$, since if there would be a vertex in A_i^1 with two neighbors in B_i^1 such three vertices would form an improving set. Moreover, let B_i^9 be the subset of vertices of B_i having at least 9 neighbors in A_i^1. If $|B_i^1| \le \varepsilon|A|$, then terminate the process, i.e. $\ell := i$ is the last value of i considered. Otherwise, set $B_{i+1} := B_i \setminus (B_i^1 \cup B_i^9)$ and $A_{i+1} := A_i \setminus A_i^1$ (see Fig. 2). Note that since in each round at least $\varepsilon|A|$ vertices are removed from A_i the process ends in at most $1/\varepsilon$ rounds, i.e. $\ell \le 1+1/\varepsilon$.

Now we prove, that if the Invariant $(*)$ is satisfied for i, then it is also satisfied for $i+1$. Let $Y \subseteq B_{i+1}$ be an improving set in H_{i+1}. Let $X = N_{H_i}(Y) \cap A_i^1$, and observe that $|X| \le 8|Y|$, since each vertex of Y has at most 8 neighbors in A_i^1. Consequently $Y \cup (N_{H_i}(X) \cap B_i^1)$ is an improving set of size at most $9|Y|$. Indeed, each vertex in A_i^1 has exactly one neighbor in B_i^1, and therefore $9|Y| \le 9^{1+1/\varepsilon - i}(4+4/\varepsilon)\log_2 n$, which gives Invariant $(*)$ for $i+1$.

Now we want to analyze the ratio $|B|/|A|$. Observe that since $\ell \le 1+1/\varepsilon$ by Invariant $(*)$ there is no improving set of size at most $(4+4/\varepsilon)\log_2 n$ in H_ℓ. Let us partition vertices of B_ℓ according to their degree in H_ℓ. By $B_\ell^0, B_\ell^1, B_\ell^2, B_\ell^{3+}$ we denote the vertices of B_ℓ with degree zero, one, two and at least three in H_ℓ respectively. Observe that $B^0 = \emptyset$, since otherwise there would be an improving set of size one in H_ℓ. Denote $a = |A_\ell|$, $b_1 = |B_\ell^1|$, $b_2 = |B_\ell^2|$, $b_3 = |B_\ell^{3+}|$, and observe that $|E(H_\ell)| \ge b_1 + 2b_2 + 3b_3$. Recall that $b_1 \le \varepsilon|A|$ and the degree k upper bound gives $|E(H_\ell)| \le ka$. Now we want to upper bound b_2. We consider one more auxiliary undirected multigraph graph H', where as the vertex set we take A, for each vertex $v \in B_\ell^2, N_H(v) = \{a_1, a_2\}$ we add to H' an edge a_1a_2. Observe also that H' consists of exactly a vertices and b_2 edges (potentially some of them are parallel). Since S can not be improved by a set of size at most $4(1+1/\varepsilon)\log_2 n$ by Lemma 4.1 we

[5] Recall that a h-claw is a star with h leaves, and a graph is h-claw free if it does not contain a h-claw as an induced subgraph.

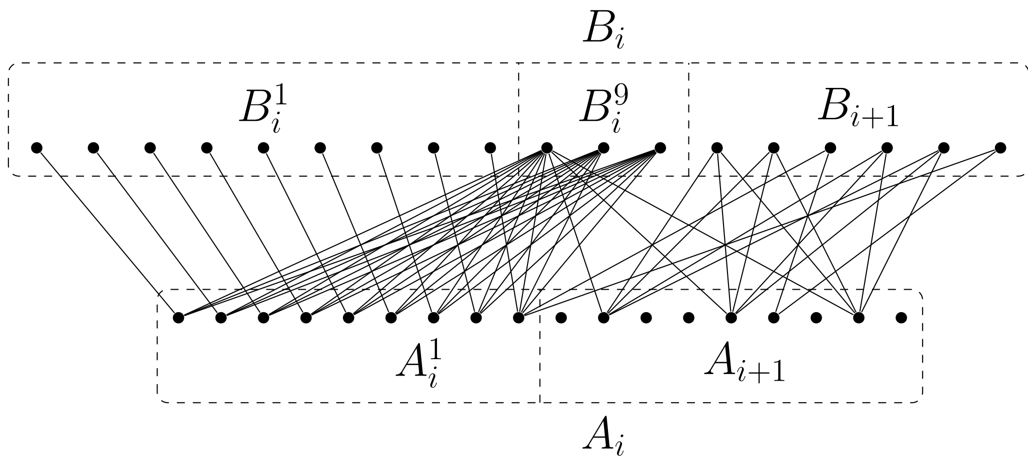

Figure 2: Constructing the sets A_{i+1}, B_{i+1} out of A_i, B_i.

infer that $|E(H')| \leq |V(H')|(1+\varepsilon)$ and consequently $b_2 \leq (1+\varepsilon)a$. Therefore

$$\begin{aligned} k|A_\ell| \geq |E(H_\ell)| &\geq b_1 + 2b_2 + 3b_3 = 3|B_\ell| - b_2 - 2b_1 \\ &\geq 3|B| - (1+\varepsilon)|A_\ell| - 2\varepsilon|A| \end{aligned}$$

and consequently $(\frac{k+1+\varepsilon}{3})|A_\ell| + 2\varepsilon|A| \geq |B_\ell|$. Finally, observe that for each $0 \leq i < \ell$ we have $\frac{|B_i \setminus B_{i+1}|}{|A_i \setminus A_{i+1}|} = \frac{|B_i^1 \cup B_i^9|}{|A_i^1|} \leq (1 + \frac{k-1}{9}) \leq \frac{k+1}{3}$ for $k \geq 4$, which gives $\frac{|B|}{|A|} \leq \frac{k+1}{3} + \varepsilon$.

Theorem 4.1. *For any fixed $\varepsilon > 0$, there is a $\frac{k+1+\varepsilon}{3}$-approximation algorithm for the maximum cardinality independent set problem in $(k+1)$-claw free graphs, with running time $n^{O(\log n)}$.*

Proof. By Lemma 4.2 it is sufficient to find a local maximum $S \subseteq V$, for which it is impossible to improve the cardinality of S by replacing $O(\log n)$ vertices. Since the cardinality of an independent set can be increased at most n times and in time $O(n^{\log n})$ we can check whether an improvement with a set of size $O(\log n)$ can be made, the theorem follows.

Corollary 4.1. *For any fixed $\varepsilon > 0$, there is a $\frac{k+1+\varepsilon}{3}$-approximation algorithm for the unweighted k-hypermatching problem, with running time $n^{O(\log n)}$.*

Proof. By Theorem 4.1, since the intersection graph of a set system with sets of cardinality k is $(k+1)$-claw free.

References

[1] M. Bateni, M. Charikar, and V. Guruswami. Maxmin allocation via degree lower-bounded arborescences. In *STOC*, pages 543–552, 2009.

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

[3] P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *SODA*, pages 365–371, 1994.

[4] P. Berman and P. Krysta. Optimizing misdirection. In *SODA*, pages 192–201, 2003.

[5] D. Chakrabarty and G. Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM J. Comput.*, 39(6):2189–2211, 2010.

[6] Y. H. Chan and L. C. Lau. On linear and semidefinite programming relaxations for hypergraph matching. In *SODA*, pages 1500–1511, 2010.

[7] B. Chandra and M. M. Halldórsson. Greedy local improvement and weighted set packing approximation. *J. Algorithms*, 39(2):223–240, 2001.

[8] C. Chekuri and S. Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2005.

[9] E. Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *FOCS*, pages 691–701, 2007.

[10] E. Chlamtac and G. Singh. Improved approximation guarantees through higher levels of SDP hierarchies. In *APPROX-RANDOM*, pages 49–62, 2008.

[11] E. Chlamtac and M. Tulsiani. Convex relaxations and integrality gaps. In *to appear in Handbook on semidefinite, conic and polynomial optimization*. Springer.

[12] W. F. de la Vega and C. Kenyon-Mathieu. Linear programming relaxations of maxcut. In *SODA*, pages 53–61, 2007.

[13] U. Feige and J. Vondrák. The submodular welfare

problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010.

[14] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *SODA*, pages 611–620, 2006.

[15] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[16] V. Guruswami and A. K. Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *FOCS*, pages 482–491, 2011.

[17] M. M. Halldórsson. Approximating discrete collections via local improvements. In *SODA*, pages 160–169, 1995.

[18] M. M. Halldórsson. Personal communication, 2012.

[19] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math.*, 182(1):105–142, 1999.

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

[21] 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.

[22] A. R. Karlin, C. Mathieu, and C. T. Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. In *IPCO*, pages 301–314, 2011.

[23] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *IPCO*, pages 293–303, 2001.

[24] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. on Optimization*, 1(12):166–190, 1991.

[25] A. Magen and M. Moharrami. Robust algorithms for on minor-free graphs based on the Sherali-Adams hierarchy. In *APPROX-RANDOM*, pages 258–271, 2009.

[26] G. J. Minty. On maximal independent sets of vertices in claw-free graphs. *J. Comb. Theory, Ser. B*, 28(3):284–304, 1980.

[27] T. Rothvoß. Directed Steiner tree and the Lasserre hierarchy. *CoRR*, abs/1111.5473, 2011.

[28] G. R. Schoenebeck. *Limitations of Linear and Semidefinite Programs.* PhD thesis, EECS Department, University of California, Berkeley, Aug 2010.

[29] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.

[30] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.