Algorithms to Approximate Column-Sparse Packing Problems

Brian Brubach*

Karthik A. Sankararaman†

Aravind Srinivasan‡

Pan Xu§

University of Maryland, College Park, USA

Abstract

Column-sparse packing problems arise in several contexts in both deterministic and stochastic discrete optimization. We present two unifying ideas, *(non-uniform) attenuation* and *multiple-chance algorithms*, to obtain improved approximation algorithms for some well-known families of such problems. As three main examples, we attain the integrality gap, up to lower-order terms, for known LP relaxations for k-column sparse packing integer programs (Bansal *et al.*, *Theory of Computing*, 2012) and stochastic k-set packing (Bansal *et al.*, *Algorithmica*, 2012), and go "half the remaining distance" to optimal for a major integrality-gap conjecture of Füredi, Kahn and Seymour on hypergraph matching (*Combinatorica*, 1993).

1 Introduction

Column-sparse packing problems arise in numerous contexts (e.g., [2, 15, 12, 13, 47, 16, 25, 24, 52, 39, 46]). We present two unifying ideas (attenuation and multiple-chances) to obtain improved approximation algorithms and/or (constructive) existence results for some well-known families of such problems. These two unifying ideas help better handle the *contention resolution* [32] that is implicit in such problems. As three main examples, we attain the integrality gap (up to lower-order terms) for known LP relaxations for k-column sparse packing integer programs (k-CS-PIP: Bansal *et al.* [13]) and stochastic k-set packing (SKSP: Bansal *et al.* [12]), and go "half the remaining distance" to optimal for a major integrality-gap conjecture of Füredi, Kahn and Seymour on hypergraph matching [39].

Letting $\mathbb{R}_+$ denote the set of non-negative reals, a general *Packing Integer Program* (PIP) takes the form:

$$(1.1) \quad \max\left\{f(\mathbf{x}) \mid \mathbf{A}\cdot\mathbf{x} \le \mathbf{b}, \mathbf{x} \in \{0,1\}^n\right\}, \quad \text{where } \mathbf{b} \in \mathbb{R}_+^m, \text{ and } \mathbf{A} \in \mathbb{R}_+^{m\times n};$$

here $\mathbf{A}\cdot\mathbf{x} \le \mathbf{b}$ means, as usual, that $\mathbf{A}\cdot\mathbf{x} \le \mathbf{b}$ coordinate-wise. Furthermore, n is the number of variables/columns, m is the number of constraints/rows, $\mathbf{A}$ is the matrix of sizes with the j^{th} column representing the size vector $\mathsf{SI}_j \in \mathbb{R}_+^m$ of *item* j, $\mathbf{b}$ is the *capacity* vector, and f is some non-decreasing function (often of the form $\mathbf{w}\cdot\mathbf{x}$, where $\mathbf{w}$ is a nonnegative vector of weights). The items' "size vectors" SI_j can be deterministic or random. PIPs generalize a large class of problems in combinatorial optimization. These range from optimally solvable problems such as classical matching to much harder problems like independent set which is NP-Hard to approximate to within a factor of $n^{1-\epsilon}$ [70].

A k-column sparse packing program (k-CS-PP) refers to a special case of packing programs wherein each size vector SI_j (a column of $\mathbf{A}$) takes positive values only on a subset $\mathcal{C}(j) \subseteq [m]$ of coordinates with $|\mathcal{C}(j)| \le k$. The k-CS-PP family captures a broad class of packing programs that are well studied such as k-column sparse packing integer programs (k-CS-PIP), k-uniform hypergraph matching, stochastic matching, and stochastic k-set packing (SKSP). While we primarily focus on programs with linear objectives, some of these approaches can be extended to monotone submodular objectives as well from prior work (*e.g.,* [13], [32]).

We show randomized-rounding techniques (including *non-uniform attenuation*, *multiple chances*) that, along with the "nibble method" [4, 63] in some cases, yield improved results for some important families of Packing Integer Programs (PIPs). In the case of k-CS-PIP and SKSP, we show *asymptotically optimal* bounds matching the LP integrality gap (as a function of the column-sparsity k, which is our asymptotic parame-

*Email: bbrubach@cs.umd.edu Supported in part by NSF Awards CNS 1010789 and CCF 1422569.

†Email: kabinav@cs.umd.edu Supported in part by NSF Awards CNS 1010789 and CCF 1422569.

‡Email: srin@cs.umd.edu Supported in part by NSF Awards CNS-1010789, CCF-1422569 and CCF-1749864, and by research awards from Adobe, Inc.

§Email: panxu@cs.umd.edu Supported in part by NSF Awards CNS 1010789 and CCF 1422569.

ter). For hypergraph matching, we make progress "half the remaining way" towards meeting a classic conjecture of Füredi *et al.* [39]. Additionally, we show a simple application of simulation-based attenuation to obtain improved ratios for the Unsplittable Flow Problem on trees (UFP-TREES: Chekuri *et al.* [30]) with unit demands and submodular objectives, a problem which admits a natural packing-LP relaxation.

1.1 Preliminaries and Main Results The natural LP relaxation is as follows (although additional valid constraints are necessary for k-CS-PIP [13]):

$$\max\{\mathbf{w}\cdot\mathbf{x} : \mathbf{A}\cdot\mathbf{x} \leq \mathbf{b}, \mathbf{x} \in [0,1]^n\} \tag{1.2}$$

Typically, a rounding algorithm takes as input an optimal solution $\mathbf{x} \in [0,1]^n$ to LP (1.2) – or one of its relatives – and outputs an integral $\mathbf{X} \in \{0,1\}^n$ which is feasible for PIP (1.1) such that the resultant *approximation ratio*, $\frac{\mathbb{E}[\mathbf{w}\cdot\mathbf{X}]}{\mathbf{w}\cdot\mathbf{x}}$, is maximized. Note that $\mathbb{E}[\mathbf{w}\cdot\mathbf{X}]$ is the expected weight of the solution over all randomness in the algorithm and/or the problem itself. For a general weight vector $\mathbf{w}$, we often seek to maximize $\min_{j:x_j\neq 0}\frac{\mathbb{E}[X_j]}{x_j}$, as the usual "local" strategy of maximizing the approximation ratio. As notation, we will denote the support of $\mathbf{X}$ as the set of *rounded* items. We say item j *participates* in constraint i if and only if $A_{ij} \neq 0$. We say that a variable is *safe* to be rounded to 1 if doing so would not violate any constraint conditional on the variables already rounded; we call it *unsafe* otherwise.

k-Column Sparse Packing Integer Programs (k-CS-PIP). Suppose we have n items and m constraints. Each item $j \in [n]$ has a weight w_j and a column $\mathbf{a}_j \in [0,1]^m$. Suppose we have a capacity vector $\mathbf{b} = \mathbf{1}$ (this is w.l.o.g., see *e.g.,* Bansal *et al.* [13]) and our goal is to select a subset of items such that the total weight is maximized while no constraint is violated. In addition, we assume each column $\mathbf{a}_j$ has at most k non-zero entries. (The important special case where for all j, $\mathbf{a}_j$ lies in $\{0,1\}^m$ – and has at most k non-zero entries – is the classic *k-set packing problem*; it is NP-hard to approximate within $o(k/\log k)$ [44]. This special case is generalized in two ways below: by allowing stochasticity in stochastic k-set packing, and by allowing the column-sparsity k to vary across columns as in hypergraph matching.) Observe that this problem (*i.e.,* k-CS-PIP) can be cast as a special case of PIP shown in (1.1) with the j^{th} column of $\mathbf{A}$ being $\mathbf{A}[j] = \mathbf{a}_j$. The resultant LP relaxation is as follows (just as in Bansal *et al.* [13], we will ultimately use a stronger form of this LP relaxation which incorporates additional valid constraints; see (2.7) in Section 2.1).

$$\max\{\mathbf{w}\cdot\mathbf{x} : \mathbf{A}\cdot\mathbf{x} \leq \mathbf{1}, \mathbf{x} \in [0,1]^n\} \quad \text{where } \mathbf{A}[j] = \mathbf{a}_j \tag{1.3}$$

For general PIPs, the best-known approximation bounds are shown in Srinivasan [67]. The problem of k-CS-PIP, in its full generality, was first considered by Pritchard [61] and followed by several subsequent works such as Pritchard and Chakrabarty [62] and Bansal *et al.* [13]. Chekuri *et al.* [32] defined a contention resolution framework for submodular objectives and showed how the previous algorithms for k-CS-PIP fit into such a framework (and hence, extending the k-CS-PIP algorithms to non-negative submodular objectives by losing a constant factor in approximation)[1].

Our main result for this problem is described in Theorem 1.1. Bansal *et al.* [13] showed that the stronger LP (which adds additional valid constraints to the natural LP relaxation) has an integrality gap of at least $2k-1$. We consider the same LP, and hence our result shown in Theorem 1.1 is asymptotically optimal *w.r.t.* this LP. The previous best known results for this problem were a factor of $ek + o(k)$ due to Bansal *et al.* [13], a factor of $O(k^2)$ independently due to Chekuri *et al.* [26] [2] and Pritchard and Chakrabarty [62], and a factor of $O(2^k \cdot k^2)$ due to Pritchard [61].

THEOREM 1.1. *There exists a randomized rounding algorithm for k-CS-PIP with approximation ratio at most $2k + o(k)$ for linear objectives.*

COROLLARY 1.1. *There exists a randomized rounding algorithm for k-CS-PIP with approximation ratio at most $(2k + o(k))/\eta_f$ for non-negative submodular objectives, where η_f is the approximation ratio for $\max\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_{\mathcal{I}} \cap \{0,1\}^n\}$ (here, $F(\mathbf{x})$ is the multi-linear extension of the sub-modular function f and $\mathcal{P}_{\mathcal{I}}$ is the k-CS-PIP polytope); $\eta_f = 1 - 1/e$ and $\eta_f = 0.385$ in the cases of non-negative monotone and non-monotone submodular functions respectively*[3].

Stochastic k-Set Packing (SKSP)**.** The Stochastic k-Set Packing problem was first introduced in Bansal *et al.* [12] as a way to generalize several stochastic-optimization problems such as Stochastic Matching[4]. The problem

[1] In [13], the authors also show extensions to non-negative monotone submodular objectives.

[2] Note that this work is cited in [13].

[3] To keep consistent with prior literature, we state all approximation ratios for sub-modular maximization (*i.e.,* η_f) as a value less than 1. This is in contrast to the approximation ratios defined in this paper where the values are always greater than 1

[4] Here, we use the definition from the journal version [12]; the conference version of [12] defines the problem slightly differently.

can be defined formally as follows. Suppose we have n items and that each item j has a random non-negative weight W_j and a random m-dimensional size vector $\mathsf{SI}_j \in \{0,1\}^m$. The random variables $\{R_j := (W_j, \mathsf{SI}_j) : j \in [n]\}$ are mutually independent[5]. Each random vector $R_j \in \mathbb{R}^+ \times \{0,1\}^m$ is drawn from some probability distribution: our algorithm only needs to know the values of $u_{i,j} := \mathbb{E}[\mathsf{SI}_{i,j}]$ for all i, j – where $\mathsf{SI}_{i,j}$ denotes the i^{th} component of SI_j – and $w_j := \mathbb{E}[W_j]$. Moreover, for each item j, there is a known subset $\mathcal{C}(j) \subseteq [m]$ of at most k coordinates such that $\mathsf{SI}_{i,j}$ can be nonzero only if $i \in \mathcal{C}(j)$: all coordinates in $[m] \setminus \mathcal{C}(j)$ will have value zero with probability 1. We are given a capacity vector $\mathbf{b} \in \mathbb{Z}_+^m$. The algorithm proceeds in multiple steps. At each step, we consider any one item j that has not been considered before, and which is safe with respect to the current remaining capacity, *i.e.,* adding item j to the current set of already-added items will not cause any capacity constraint to be violated regardless of what random SI_j materializes.[6] Upon choosing to probe j, the algorithm observes its size realization and weight, and has to irrevocably include j. The task is to sequentially probe some subset of the items such that the expected total weight of items added is maximized.

Let $\mathbf{w}$ denote $(w_1, \ldots, w_n)$ and x_j denote the probability that j is added in the OPT solution. Bansal *et al.* [12] introduced the following natural LP to upper bound the optimal performance.

(1.4)
$$\max\{\mathbf{w}\cdot\mathbf{x} : \mathbf{A}\cdot\mathbf{x} \leq \mathbf{b}, \mathbf{x} \in [0,1]^n\} \text{ where } \mathbf{A}[i,j] = u_{i,j}$$

The previous best known bound for SKSP was $2k + o(k)$ due to Bansal *et al.* [12]. Our main contribution (Theorem 1.2) is to improve this bound to $k + o(k)$, a result that is again asymptotically optimal *w.r.t.* the natural LP (1.4) considered (Theorem 1.3 from [39]).

THEOREM 1.2. *There exists a randomized rounding algorithm achieving an approximation ratio of $k + o(k)$ for the stochastic k-set packing problem, where the "$o(k)$" is a vanishing term when $k \to \infty$.*

Hypergraph Matching. Suppose we have a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = m$ and $|\mathcal{E}| = n$. (This is the opposite of the usual graph notation, but is convenient for us since the LP here has $|\mathcal{V}|$ constraints and $|\mathcal{E}|$ variables.) Each edge $e \in \mathcal{E}$ has a weight w_e. We need to find a subset of edges with maximum total weight such that every pairwise intersection is empty (*i.e.,* we obtain a hypergraph matching). Observe that the problem of finding a maximum weighted hypergraph matching can be cast as a special case of PIP. Let $\mathbf{w} = (w_e)$ and $\mathbf{e} \in \{0,1\}^m$ be the canonical (characteristic-vector) representation of e. Then the natural LP relaxation is as follows:

(1.5)
$$\max\{\mathbf{w}\cdot\mathbf{x} : \mathbf{A}\cdot\mathbf{x} \leq \mathbf{1}, \mathbf{x} \in [0,1]^n\} \text{ where } \mathbf{A}[j] = \mathbf{e}_j$$

Note that in these natural IP and LP formulations, the number of vertices in an edge e, $k_e = |e|$, can be viewed as the column-sparsity of the column associated with e. Thus, this again broadly falls into the class of column-sparse packing programs. For general hypergraphs, Füredi *et al.* [39] presented the following well-known conjecture.

CONJECTURE 1. (FÜREDI *et al.* [39]) *For any hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a weight vector $\mathbf{w} = (w_e)$ over all edges, there exists a matching $\mathcal{M}$ such that*

$$\sum_{e \in \mathcal{M}} \left(k_e - 1 + \frac{1}{k_e}\right) w_e \geq \mathrm{OPT}(\mathcal{H}, \mathbf{w}) \tag{1.6}$$

where k_e denotes the number of vertices in hyperedge e and $\mathrm{OPT}(\mathcal{H}, \mathbf{w})$ denotes an optimal solution to the LP relaxation (1.5) of hypergraph matching.

The function "$k_e - 1 + \frac{1}{k_e}$" is best-possible in the sense that certain hypergraph families achieve it [39]. We generalize Conjecture 1 slightly:

CONJECTURE 2. (GENERALIZATION OF CONJECTURE 1) *For any given hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with notation as in Conjecture 1, let $\mathbf{x} = (x_e : e \in \mathcal{E})$ denote a given optimal solution to the LP relaxation (1.5). Then: (i) there is a distribution $\mathcal{D}$ on the matchings of $\mathcal{H}$ such that for each edge e, the probability that it is present in a sample from $\mathcal{D}$ is at least $\frac{x_e}{k_e - 1 + 1/k_e}$, and (ii) $\mathcal{D}$ is efficiently samplable.*

Part (i) of Conjecture 2 immediately implies Conjecture 1 via the linearity of expectation.

Füredi *et al.* [39] gave (non-constructive) proofs for Conjecture 1 for the three special cases where the hypergraph is either uniform, intersecting, or uniformly weighted. Chan and Lau [24] gave an algorithmic proof of Conjecture 1 for k-uniform hypergraphs, by combining the iterative rounding method and the fractional local ratio method. Using similar techniques, Parekh [58] and Parekh and Pritchard [59] generalized this to k- uniform b-hypergraph matching. We go "half the remaining distance" in resolving Conjecture 2 for *all* hypergraphs, and also do so algorithmically: the work of Bansal *et al.* [12] gives "$k_e + 1$" instead of the target $k_e - 1 + 1/k_e$ in Conjecture 2, and we improve this to $k_e + O(k_e \cdot \exp(-k_e))$.

[5] Note that W_j can be correlated with SI_j.

[6] This is called the *safe-policy* assumption. This allows us to handle the correlations between W_j and coordinates of SI_j. A detailed discussion of this model can be found in [12].

THEOREM 1.3. *There exists an efficient algorithm to generate a random matching $\mathcal{M}$ for a hypergraph such that each edge e is added in $\mathcal{M}$ with probability at least $\frac{x_e}{k_e+o(1)}$, where $\{x_e\}$ is an optimal solution to the standard* LP (1.5) *and where the $o(1)$ term is $O(k_e \exp(-k_e))$, a vanishing term when $k_e \to \infty$.*

UFP-TREES **with unit demands.** In this problem, we are given a tree $T = (\mathcal{V}, \mathcal{E})$ with each edge e having an integral capacity u_e. We are given k distinct pairs of vertices $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ each having unit demand. Routing a demand pair (s_i, t_i) exhausts one unit of capacity on all the edges in the path. With each demand pair i, there is an associated weight $w_i \geq 0$. The goal of the problem is to choose a subset of demand pairs to route such that no edge capacity is violated, while the total weight of the chosen subset is maximized. In the non-negative submodular version of this problem, we are given a non-negative submodular function f over all subsets of demand pairs, and aim to choose a feasible subset that maximizes f. This problem was introduced by Chekuri *et al.* [30] and the extension to submodular version was given by Chekuri *et al.* [32]. We show that by incorporating simple attenuation ideas, we can improve the analysis of the previous best algorithm for the Unsplittable Flow Problem in Trees (UFP-TREES) with unit demands and non-negative submodular objectives.

Chekuri *et al.* [32] showed that they can obtain an approximation of $27/\eta_f$, where η_f is the approximation ratio for maximizing a non-negative submodular function, via their contention-resolution scheme (henceforth abbreviated as *CR schemes*)[7]. We improve their $1/27$-balanced CR scheme to a $1/8.15$-balanced CR scheme via attenuation and hence achieve an approximation of $8.15/\eta_f$ for non-negative sub-modular objectives.

THEOREM 1.4. *There exists a $8.15/\eta_f$-approximation algorithm to the* UFP-TREES *with unit demands and non-negative submodular objectives.*

Extension to submodular objectives. Chekuri *et al.* [32] showed that given a rounding scheme for a PIP with linear objectives, we can extend it to non-negative submodular objectives by losing only a constant factor, if the rounding scheme has a certain structure (see Theorem 2.4, due to [32]). Our improved algorithm for k-CS-PIP and UFP-TREES admits this structure and hence can be extended to non-negative sub-modular functions. See Section B in the Appendix for the required background on submodular functions.

A simple but useful device that we will use often is as follows.

Simulation-based attenuation. We use the term *simulation* throughout this paper to refer to Monte Carlo simulation and the term *simulation-based attenuation* to refer to the simulation and attenuation techniques as shown in [2] and [19]. At a high level, suppose we have a randomized algorithm such that for some event E (*e.g.,* the event that item j is safe to be selected into the final set in SKSP) we have $\Pr[E] \geq c$, then we modify the algorithm as follows: (i) We first use simulation to estimate a value $\hat{E}$ that lies in the range $[\Pr[E], (1+\epsilon)\Pr[E]]$ with probability at least $1-\delta$. (ii) By "ignoring" E (i.e., *attenuation*, in a problem-specific manner) with probability $\sim c/\hat{E}$, we can ensure that the final effective value of $\Pr[E]$ is arbitrarily close to c, i.e., in the range $[c/(1+\epsilon), c]$ with probability at least $1-\delta$. This simple idea of attenuating the probability of an event to come down approximately to a certain value c is what we term simulation-based attenuation. The number of samples needed to obtain the estimate $\hat{E}$ is $\Theta(\frac{1}{c\epsilon^2} \cdot \log(\frac{1}{\delta}))$ via a standard Chernoff-bound argument. In our applications, we will take $\epsilon = 1/\text{poly}(N)$ where N is the problem-size, and the error ϵ will only impact lower-order terms in our approximations.

1.2 Our Techniques

In this section, we describe our main technical contributions of the paper and the ingredients leading up to them.

Achieving the integrality gap of the LP of [13] for k-CS-PIP. Our first main contribution in this paper is to achieve the integrality gap of the strenghthened LP of [13] for k-CS-PIP, up to lower-order terms: we improve the $ek + o(k)$ of [13] to $2k + o(k)$. We achieve this by following the same overall structure as in [13] and improve the alteration steps using randomization. We view the alteration step as a question on an appropriately constructed directed graph. In particular, a key ingredient in the alteration step answers the following question. "Suppose we are given a directed graph G such that the maximum out-degree is bounded by an asymptotic parameter d. Find a random independent set $\mathcal{I}$ in the undirected version of this graph such that every vertex is added into $\mathcal{I}$ with probability at least $1/(2d) - o(1/d)$". It turns out that this question can be answered by looking at the more-general question of finding a good coloring of the undirected version of this graph. The key idea here is to "slow down" the contention-resolution approach of [13], leading to Theorem 1.1. However, motivated by works that obtain strong "negative correlation" properties – e.g., the papers [31, 60] obtain *negative cylindrical correlation*[8] and the even-stronger *negative association* for rounding in matroid polytopes – we ask next if one

[7]Please see Section 2.3 for formal definitions of CR schemes.

[8]This is sometimes simply called "negative correlation".

can achieve this for k-CS-PIP. (It is well-known that even negative cylindrical correlation yields Chernoff-type bounds for sums of random variables [57]; we use this in Section 5.) We make progress toward this in Theorem 2.3.

Achieving the integrality gap of the natural LP for SKSP **via a "multiple chances" technique.** Our second contribution in is to develop an algorithm that achieves the integrality gap of $k + o(k)$ for SKSP, improving on the $2k$ of [12]. To achieve this, we introduce the "multiple-chances" technique. We will now informally describe this technique, which is motivated by the powerful "nibble" idea from probabilistic combinatorics (see, e.g., Ajtai, Komlós, and Szemerédi [4] and Rödl [63]).

The current-best ratios for many special cases of k-CS-PP are $\Theta(k)$; e.g., $ek + o(k)$ for k-CS-PIP [13], the optimal approximation ratio (*w.r.t.* the integrality gap) of $k-1+1/k$ for k-uniform hypergraph matching [24], or the $2k$-approximation for SKSP [12]. Thus, many natural approaches involve sampling items with a probability proportional to $1/k$. Consider a k-CS-PP instance with budget $\mathbf{b}$. Suppose we have a randomized algorithm ALG which outputs a solution SOL wherein each item j is added SOL with probability exactly *equal* to $x_j/(ck)$ for some constant $c > 0$.[9] After running ALG, the expected usage of each budget i is $b_i/(ck)$; this follows directly from the budget constraint in the LP. This implies that after running ALG, we have only used a tiny fraction of the whole budget, in expectation. Thus, we may run ALG again on the remaining items to further improve the value/weight of SOL. Hence, an item that was previously not chosen, receives a "second chance" to be rounded up and included in the solution. The observation that only a tiny fraction of the budget is used can be made after running ALG for a second time as well. Hence, in principle, we can run ALG multiple times and we call the overarching approach a *multiple chance algorithm*. The analysis becomes rather delicate as we run for a large number of iterations in this manner.

"Half the remaining distance" toward resolving the FKS conjecture and non-uniform attenuation approach. Our third contribution is in making significant progress on the well-known Conjecture 1 due to Füredi, Kahn and Seymour. To achieve this, we introduce a technique of *non-uniform* attenuation. A common framework for tackling k-CS-PP and related problems is random permutation followed by sampling via *uniform attenuation*: follow a random order π on the items and add each item j with probability αx_j whenever it is safe, where x_j is an optimal solution to an appropriate LP and α is the attenuation factor. Typically α is a parameter fixed in the analysis to get the best ratio (*e.g.,* see the SKSP algorithm in Bansal *et al.* [12]). This method is called *uniform attenuation*, since all items share the same attenuation factor α.

An alternative strategy used previously is that of *weighted random permutations* (see, e.g., Adamczyk *et al.* [2] and Baveja *et al.* [15]): instead of using a uniformly-random permutation, the algorithm "weights" the items and permutes them non-uniformly based on their x_j values. We introduce a notion of *non-uniform attenuation*, which approaches the worst-case scenario in a different manner. We still stay within the regime of uniform permutations but will attenuate items non-uniformly, based on their x_j values; a careful choice of attenuation function is very helpful here, as suggested by the optimization problem (4.12). This is a key ingredient in our improvement.

1.3 Other Related Work

In this subsection, we list the related work not mentioned in previous sections and yet closely related to the problems we study. Note that packing programs, submodular maximization, and their applications to approximation algorithms have a vast literature. Our goal here is to list some papers in closely relevant areas and this is by no means an exhaustive list of references in each of these closely-aligned areas.

For k-CS-PIP, related problems have been studied in discrepancy theory. In such problems, we have a k-column sparse LP and we want to round the fractional solution such that the violation (both above and below) of any constraint is minimized. This study started with the famous work of Beck and Fiala [17] and some of the previous work on k-CS-PIP (*e.g.,* [61]) used techniques similar to Beck and Fiala. There has been a long line of work following Beck and Fiala, including [66, 7, 14, 55, 64, 43, 11, 53, 9]. One special case of k-CS-PIP is the k-set packing problem. Many works including [45, 5, 25, 18] studied this problem with [18] giving the best approximation of $(k+1)/2 + \epsilon$ for this problem. Closely related to k-CS-PIP is the notion of column-restricted packing introduced by Kolliopoulos and Stein [47]. Many works have studied this version of packing programs, including [30, 27, 16].

Similar to Bansal *et al.* [13], our algorithms also extend to submodular objective functions. In particular, we use tools and techniques from Calinescu *et al.* [22] and Chekuri *et al.* [32] for both k-CS-PIP and the UFP problem on trees. Monotone sub-modular function maximization subject to k-sparse constraints has been studied in the

[9] Note that given an algorithm which adds each item with probability *at least* $x_j/(ck)$, we can use Monte Carlo simulation and attenuation techniques similar to [32, 2, 19] to get an algorithm which adds each item with probability essentially *equal* to $x_j/(ck)$.

context of k-partition matroids, k-knapsacks, and the intersection of k partition matroids in many works including [38, 51, 69, 50]. Beyond the monotone case, there are several algorithms for the non-negative sub-modular maximization problem including [37, 28, 21, 36, 20].

Stochastic variants of PIPs have also been previously studied. Baveja *et al.* [15] considered the following stochastic setting of k-uniform hypergraph matching: the algorithm has to probe edge e to check its existence; each edge e is associated with a probability $0 < p_e \leq 1$ with which it will be present (independently of other edges) on being probed; the task is to sequentially probe edges such that the expected total weight of matching obtained is maximized. The stochastic version of hypergraph matching can be viewed as a natural generalization of stochastic matching (*e.g.,* Bansal *et al.* [12]) to hypergraphs. The work of [15] gave an $(k + \epsilon + o(1))$–approximation algorithm for any given $\epsilon > 0$ asymptotically for large k. Other work on stochastic variants of PIPs includes [33, 34, 54, 3, 42, 1, 41].

Later in this paper, we show yet another application of attenuation: UFP-TREES with unit demands. This problem is a more specific version of column-restricted packing problems mentioned previously. The Unsplittable Flow Problem in general graphs and its various specializations on different kinds of graphs has been extensively studied. Some of these works include [46, 65, 40, 48, 29, 8, 6, 49, 23, 10, 35].

1.4 Outline

In Section 2, we present a randomized rounding algorithm for k-CS-PIP using randomized alteration techniques. We analyze this algorithm to prove Theorem 1.1 and show an extension to submodular objectives. In Section 3, we apply second-chance techniques to SKSP. After analyzing this algorithm, we show how it can be extended to multiple chances, yielding the improved result of Theorem 1.2. In Section 4, we present an algorithm for hypergraph matching and analyze it to prove Theorem 1.3, making progress toward Conjecture 2 (and by extension Conjecture 1). In Section 5, we show how attenuation can lead to an improved contention resolution scheme for UFP-TREES, proving Theorem 1.4. We end with a brief conclusion and discussion of open problems in Section 6. All proofs from main section can be found in Section 7. Appendix A contains a few useful technical lemmas used in this paper while Appendix B gives a self-contained background on submodular functions.

2 k-Column Sparse Packing

We describe a rounding algorithm for k-CS-PIP, which achieves the asymptotically *optimal* approximation ratio of $(2k + o(k))$ with respect to the strengthened LP shown in Bansal *et al.* [13] (see (2.7) in Section 2.1). Theorem 2.3 then develops a near-negative-correlation generalization of this result.

Recall that we have a k-sparse matrix $\mathbf{A} \in [0,1]^{m \times n}$ and a fractional solution $\mathbf{x} \in [0,1]^n$ such that $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}$. Our goal is to obtain an integral solution $\mathbf{X} \in \{0,1\}^n$ (possibly random) such that $\mathbf{A} \cdot \mathbf{X} \leq \mathbf{1}$ and such that the expected value of the objective function $\mathbf{w} \cdot \mathbf{X}$ is "large". (We will later extend this to the case where the objective function $f(\mathbf{X})$ is monotone submodular.) At a very high level, our algorithm performs steps similar to the contention-resolution scheme defined by Chekuri *et al.* [32]; the main contribution is in the details[10]. We first perform an independent-sampling step to obtain a random set $\mathcal{R}$ of variables; we then conduct *randomized* alterations to the set $\mathcal{R}$ to obtain a set of rounded variables that are feasible for the original program with probability 1. Note that the work of [13] uses *deterministic* alterations. Moving from deterministic alteration to careful randomized alteration, as well as using a much-less aggressive uniform attenuation in the initial independent sampling, yield us the optimal bound.

2.1 Algorithm

Before describing the algorithm, we review some useful notations and concepts, some of which were introduced in [13]. For a row i of $\mathbf{A}$ and $\ell = k^{1/3}$, let $\mathrm{big}(i) := \{j : a_{ij} > 1/2\}$, $\mathrm{med}(i) := \{j : 1/\ell \leq a_{ij} \leq 1/2\}$, and $\mathrm{tiny}(i) := \{j : 0 < a_{ij} < 1/\ell\}$, which denote the set of *big*, *medium*, and *tiny* items with respect to constraint i. For a given randomly sampled set $\mathcal{R}$ and an item $j \in \mathcal{R}$, we have three kinds of *blocking events* for j. Blocking events occur when a set of items cannot all be rounded up without violating some constraint. In other words, theses events, with probability 1, prevent j from being rounded up. We partition the blocking events into the following three types:

- $\mathrm{BB}(j)$: There exists some constraint i with $a_{ij} > 0$ and an item $j' \neq j$ such that $j' \in \mathrm{big}(i) \cap \mathcal{R}$.
- $\mathrm{MB}(j)$: There exists some constraint i with $\mathrm{med}(i) \ni j$ such that $|\mathrm{med}(i) \cap \mathcal{R}| \geq 3$.
- $\mathrm{TB}(j)$: There exists some constraint i with $\mathrm{tiny}(i) \ni j$ such that $\sum_{\substack{j' \neq j: \\ j' \in (\mathrm{med}(i) \cup \mathrm{tiny}(i)) \cap \mathcal{R}}} a_{ij'} > 1 - a_{ij}$.

Informally, we refer to the above three blocking events as *big, medium* and *tiny* blocking events for j with respect to $\mathcal{R}$.

[10] We would like to point out that the work of [13] also performs similar steps and fits into the framework of [32].

The main algorithm of Bansal *et al.* [13]. As briefly mentioned in Section 1.1, Bansal *et al.* add certain valid constraints on big items to the natural LP relaxation in (1.3) as follows:

$$\max\{\mathbf{w}\cdot\mathbf{x} \text{ s.t. } \mathbf{A}\cdot\mathbf{x}\leq\mathbf{1}\text{and} \quad \forall i\in[m] \quad \sum_{j\in\mathrm{big}(i)} x_j \leq 1,\ \mathbf{x}\in[0,1]^n\}\text{where } \mathbf{A}[j]=\mathbf{a}_j \tag{2.7}$$

Algorithm 1, BKNS, gives a formal description of the algorithm of Bansal *et al.* [13], in which they set $\alpha = 1$.

Algorithm 1: BKNS(α)

1. **Sampling**: Sample each item j independently with probability $(\alpha x_j)/k$ and let $\mathcal{R}_0$ be the set of sampled items.
2. **Discarding low-probability events**: Remove an item j from $\mathcal{R}_0$ if either a medium or tiny blocking event occurs for j with respect to $\mathcal{R}_0$. Let $\mathcal{R}_1 \subseteq \mathcal{R}_0$ be the set of items not removed.
3. **Deterministic alteration**: Remove an item j from $\mathcal{R}_1$ if a big blocking event occurs for j with respect to $\mathcal{R}_1$.
4. Let $\mathcal{R}_F \subseteq \mathcal{R}_1$ be the set of items not removed; return $\mathcal{R}_F$.

THEOREM 2.1. (BANSAL *et al.* [13]) *By choosing $\alpha = 1$, Algorithm 1 yields a randomized $ek + o(k)$-approximation for k-CS-PIP.*

Our algorithm for k-CS-PIP via randomized alterations. Our pre-processing is similar to BKNS with the crucial difference that $\alpha \gg 1$ (but not too large), i.e., we do not attenuate too aggressively; furthermore, our alteration step is quite different. Let $[n] = \{1, 2, \ldots, n\}$ denote the set of items. We first sample each item independently using an appropriate product distribution over the items (as mentioned above, we crucially use a different value for α than BKNS). Let $\mathcal{R}_0$ denote the set of sampled items. We remove items j from $\mathcal{R}_0$ for which either a medium or tiny blocking event occurs to obtain a set $\mathcal{R}_1$. We next perform a randomized alteration, as opposed to a deterministic alteration such as in line 3 of BKNS. We then randomly and appropriately shrink $\mathcal{R}_1$ to obtain the final set $\mathcal{R}_F$.

We now informally describe our randomized alteration step. We construct a directed graph $G = (\mathcal{R}_1, E)$ from the constraints as follows. For every item $j \in \mathcal{R}_1$, we create a vertex. We create a directed edge from item j to item $j' \neq j$ in G iff j' causes a big blocking event for j (*i.e.,* there exists a constraint i where j has a non-zero coefficient and j' is in $\mathrm{big}(i)$). We claim that the expected degree of every vertex in this graph constructed with $\mathcal{R}_1$ is at most $\alpha = \omega(1)$. If any vertex j has a degree greater than $d := \alpha + \alpha^{2/3}$, we will remove j from $\mathcal{R}_1$. Hence we now have a directed graph with every vertex having a degree of at most d. We claim that we can color the undirected version of this directed graph with at most $2d + 1$ colors. We choose one of the colors c in $[2d + 1]$ uniformly at random and add all vertices of color c into $\mathcal{R}_F$. Algorithm 2 gives a formal description of our approach.

Algorithm 2: The Algorithm for k-CS-PIP

1. **Sampling**: Sample each item j independently with probability $\alpha x_j/k$ (where, say, $\alpha = \log k$) and let $\mathcal{R}_0$ be the set of sampled items.
2. **Discarding low-probability events**: Remove an item j from $\mathcal{R}_0$ if either a medium or tiny blocking event occurs for j with respect to $\mathcal{R}_0$. Let $\mathcal{R}_1$ be the set of items not removed.
3. **Randomized alteration**:
   (a) **Create a directed graph**: For every item in $\mathcal{R}_1$, create a vertex in graph G. Add a directed edge from item j to item j' if there exists a constraint i such that $a_{ij} > 0$ and $a_{ij'} \geq 1/2$.

   (b) **Removing anomalous vertices**: For every vertex v in G, if the out-degree of v is greater than $d := \alpha + \alpha^{2/3}$, call v anomalous. Remove all anomalous vertices from G to obtain G' and let the items corresponding to the remaining vertices in G' be $\mathcal{R}_2$.

   (c) **Coloring G'**: Assign a coloring χ to the vertices of G' using $2d + 1$ colors as described in the text such that for any edge e (ignoring the direction), both end points of e receive different colors.

   (d) **Choosing an independent set**: Choose a number $c \in [2d + 1]$ uniformly at random. Add all vertices v from G' into $\mathcal{R}_F$ such that $\chi(v) = c$.
4. Return $\mathcal{R}_F$.

Example. Before moving to the analysis, we will show an example of how the randomized alteration (*i.e.,* lines 3(a-d) of Algorithm 2) works. We will illustrate this on the integrality gap example considered in [13]. In this example, we have $n = 2k - 1$ items and $m = 2k - 1$ constraints. The weights of all items are 1. For some $0 < \epsilon \ll 1/(nk)$, the matrix $\mathbf{A}$ is defined as follows.

$\forall i, j \in [2k-1]$ we have,

$$a_{ij} := \begin{cases} 1 & \text{if } i = j \\ \epsilon & \text{if } j \in \{i+1, i+2, \ldots, i+k-1 (\text{mod} n)\} \\ 0 & \text{otherwise} \end{cases}$$

As noted in [13], setting $x_j = (1 - k\epsilon)$ for all $j \in [n]$ is a feasible LP solution, while the optimal integral solution has value 1. After running line 1 of the algorithm, each item j is selected with probability $(1 - o(1))\alpha/k$ independently. For simplicity, we will assume that there are no medium or tiny blocking events for every j (these only contribute to the lower-order terms). Note that in expectation the total number of chosen items will be approximately 2α; with high probability, the total number of vertices in the graph will be $n_1 := 2\alpha + o(\alpha)$. Let $b_1, b_2, \ldots, b_{n_1}$ denote the set of items in this graph. The directed graph contains the edge (b_i, b_j) for all distinct i, j; for simplicity, assume that the graph has no anomalous vertices. Since the undirected counterpart of this graph is a complete graph, every vertex will be assigned a unique color; thus the solution output will have exactly one vertex with probability $1 - o(1)$.

2.2 Analysis We prove the following main theorem using Algorithm 2 with $\alpha = \log(k)$.

THEOREM 2.2. *There exists a randomized rounding algorithm for k-CS-PIP with approximation ratio at most $2k + o(k)$ for linear objectives.*

We will divide the analysis into three parts. At a high-level the three parts prove the following.

- **Part 1** (Proved in Lemma 2.1). For directed graphs with maximum out-degree at most d, there exists a coloring χ and a corresponding algorithm such that the number of colors used, $|\chi|$, is at most $2d + 1$.
- **Part 2** (Proved in Lemma 2.2). For any item $j \in \mathcal{R}_1$, the event that the corresponding vertex in G has an out-degree larger than d occurs with probability at most $o(1)$. This implies that conditional on $j \in \mathcal{R}_1$, the probability that j is present in G' is $1 - o(1)$.
- **Part 3** (Proved in Lemma 2.3). For each item $j \in \mathcal{R}_0$, either a medium or a tiny blocking event occurs with probability at most $o(1)$ (again, for our choice $\alpha = \log(k)$). This implies that for each $j \in \mathcal{R}_0$, it will be added to $\mathcal{R}_1$ with probability $1 - o(1)$.

We assume the following lemmas which are proven in Section 7.1.

LEMMA 2.1. *Given a directed graph $G = (V, E)$ with maximum out-degree at most d, there is a polynomial-time algorithm that finds a coloring χ of G's undirected version such that $|\chi|$, the number of colors used by χ, is at most $2d + 1$.*

LEMMA 2.2. *For any item j, the probability – conditional on the event "$j \in \mathcal{R}_1$" – that j is selected into $\mathcal{R}_2$ (i.e., $j \in \mathcal{R}_2$) is $1 - o(1)$, for $\alpha = \log(k)$.*

LEMMA 2.3. *For each item j, either a medium or a tiny blocking event occurs with probability – conditional on the event "$j \in \mathcal{R}_0$" – of at most $\Theta(\alpha^2/k) = o(1)$, for $\alpha = \log k$.*

We can now prove the main theorem, Theorem 1.1.

Proof. First we show that $\mathcal{R}_F$ is feasible for our original IP. We have the following observations about Algorithm 2: (i) from line 2, "Discarding low-probability events", we have that no item in $\mathcal{R}_F$ can be blocked by either medium or tiny blocking events; (ii) from the "Randomized alteration" steps in line 3, we have that no item in $\mathcal{R}_F$ has any neighbor in G' that is also included in $\mathcal{R}_F$. This implies that no item in $\mathcal{R}_F$ can be blocked by any big blocking events. Putting together the two observations implies that $\mathcal{R}_F$ is a feasible solution to our IP.

We now show that the probability of j being in $\mathcal{R}_F$ can be calculated as follows.

$$\begin{aligned}
&\Pr[j \in \mathcal{R}_F] \\
&= \Pr[j \in \mathcal{R}_0] \cdot \Pr[j \in \mathcal{R}_1 \mid j \in \mathcal{R}_0] \\
&\cdot \Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_1] \cdot \Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2] \\
&\geq \tfrac{\alpha x_j}{k} \cdot (1 - o(1)) \cdot (1 - o(1)) \cdot \tfrac{1}{2\alpha + 2\alpha^{2/3} + 1} \\
&= \tfrac{x_j}{2k(1 + o(1))}
\end{aligned}$$

The first inequality is due to the following. From the sampling step we have that $\Pr[j \in \mathcal{R}_0] = \alpha x_j/k$. From Lemma 2.3 we have that $\Pr[j \in \mathcal{R}_1 \mid j \in \mathcal{R}_0] = 1 - o(1)$. Lemma 2.2 implies that $\Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_1] = 1 - o(1)$. Finally from Lemma 2.1 we have that the total number of colors needed for items in $\mathcal{R}_2$ is at most $2d + 1$, and hence the probability of picking j's color class is $1/(2d + 1)$. Thus, $\Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2] = 1/(2\alpha + 2\alpha^{2/3} + 1)$ (recall that $d := \alpha + \alpha^{2/3}$).

***Nearly*-negative correlation.** The natural approach to proving Lemma 2.1 can introduce substantial positive correlation among the items included in $\mathcal{R}_F$. However, by slightly modifying the algorithm, we can obtain the following Theorem, which induces *nearly*-negative correlation in the upper direction among the items.

THEOREM 2.3. *Given any constant $\epsilon \in (0,1)$, there is an efficient randomized algorithm for rounding a fractional solution within the k-CS-PIP polytope, such that*

1. *For all items $j \in [n]$, $\Pr[j \in \mathcal{R}_F] \geq \frac{x_j}{2k(1+o(1))}$.*

2. *For any $t \in [n]$ and any t-sized subset $\{v_1, v_2, \ldots, v_t\}$ of items in $[n]$, we have (with $d = \alpha + \alpha^{2/3} \sim \log k$ as above)*

$$\Pr[v_1 \in \mathcal{R}_F \wedge v_2 \in \mathcal{R}_F \wedge \ldots \wedge v_t \in \mathcal{R}_F] \leq (2d^{\epsilon})^{t-1} \cdot \prod_{j=1}^{t} \frac{x_{v_j}}{2k}.$$

2.3 Extension to Submodular Objectives As described in the preliminaries, we can extend certain contention-resolution schemes to submodular objectives using prior work. We will now show that the above rounding scheme can be extended to submodular objectives; in particular, we will use the following definition and theorem from Chekuri *et al.* [32] [11].

DEFINITION 1. *(bc-BALANCED MONOTONE CR SCHEMES [32]) Let $b, c \in [0,1]$ and let $N := [n]$ be a set of items. A bc-balanced CR scheme π for $\mathcal{P}_{\mathcal{I}}$ (where $\mathcal{P}_{\mathcal{I}}$ denotes the convex relaxation of the set of feasible integral solutions $\mathcal{I} \subseteq 2^N$) is a procedure which for every $\mathbf{y} \in b\mathcal{P}_{\mathcal{I}}$ and $A \subseteq N$, returns a set $\pi_{\mathbf{y}}(A) \subseteq A \cap support(\mathbf{y})$ with the following properties:*
- *(a) For all $A \subseteq N, \mathbf{y} \in b\mathcal{P}_{\mathcal{I}}$, we have that $\pi_{\mathbf{y}} \in \mathcal{I}$ with probability 1.*
- *(b) For all $i \in support(\mathbf{y})$, $\mathbf{y} \in b\mathcal{P}_{\mathcal{I}}$, we have that $\Pr[i \in \pi_{\mathbf{y}}(R(\mathbf{y})) \mid i \in R(\mathbf{y})] \geq c$, where $R(\mathbf{y})$ is the random set obtained by including every item $i \in N$ independently with probability y_i.*
- *(c) Whenever $i \in A_1 \subseteq A_2$, we have that $\Pr[i \in \pi_{\mathbf{y}}(A_1)] \geq \Pr[i \in \pi_{\mathbf{y}}(A_2)]$.*

THEOREM 2.4. (CHEKURI *et al.* [32]) *Suppose $\mathcal{P}_{\mathcal{I}}$ admits a bc-balanced monotone CR scheme and η_f is the approximation ratio for $\max\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_{\mathcal{I}} \cap \{0,1\}^n\}$ (here, $F(\mathbf{x})$ is the multi-linear extension of the sub-modular function f). Then there exists a randomized algorithm which gives an expected $1/(bc\eta_f)$-approximation to $\max_{S \in \mathcal{I}} f(S)$, when f is a non-negative submodular function.*

For the case of monotone sub-modular functions, we have the optimal result $\eta_f = 1 - 1/e$ (Vondrák [68]). For non-monotone sub-modular functions, the best-known algorithms have $\eta_f \geq 0.372$ due to Ene and Nguyen [36] and more recently $\eta_f \geq 0.385$ due to Buchbinder and Feldman [20] (it is not known if these are tight: the best-known upper bound is $\eta_f \leq 0.478$ due to Oveis Gharan and Vondrák [56]).

We will show that Algorithm 2 is a $1/(2k+o(k))$-balanced monotone CR scheme, for some b, c such that $bc = 1/(2k+o(k))$. Hence, from **??** 2.4 we have a $(2k+o(k))/\eta_f$-approximation algorithm for k-CS-PIP with sub-modular objectives. This yields Corollary 1.1.

For ease of reading, we will first re-state notations used in Definition 1 in the form stated in the previous sub-section. The polytope $\mathcal{P}_{\mathcal{I}}$ represents the k-CS-PIP polytope defined by Eq. 2.7. The vector $\mathbf{y}$ is defined as $y_i := \alpha x_i / k$ which is used in the Sampling step of the algorithm (*i.e.,* Line 1). The scheme $\pi_{\mathbf{y}}$ is the procedure defined by lines 1, 2, 3 of the algorithm. In other words, this procedure takes a subset A of items and returns a *feasible* solution with probability 1 (and hence satisfying property (a) in the definition). Our goal then is to show that it further satisfies properties (b) and (c).

The set $R(\mathbf{y})$ corresponds to the set $\mathcal{R}_0$, where every item i is included into $\mathcal{R}_0$ with probability y_i, independently. From the sampling step of the algorithm, we have that $b = \alpha/k$, since each item i is included in the set $R(\mathbf{y})$ with probability $y_i := x_i\alpha/k$ and $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$ and hence $\mathbf{y} \in (\alpha/k)\mathcal{P}_{\mathcal{I}}$. From the alteration steps we have that $c = (1-o(1))/(2\alpha + o(\alpha))$, since for any item i, we have $\Pr[i \in \mathcal{R}_F \mid i \in \mathcal{R}_0] \geq (1-o(1))/(2\alpha+o(\alpha))$. Thus, $\pi_{\mathbf{y}}$ satisfies property (b).

Now we will show that the rounding scheme $\pi_{\mathbf{y}}$ satisfies property (c) in Definition 1. Let A_1 and A_2 be two arbitrary subsets such that $A_1 \subseteq A_2$. Consider a $j \in A_1$. We will now prove the following.

$$\Pr[j \in \mathcal{R}_F \mid \mathcal{R}_0 = A_1] \geq \Pr[j \in \mathcal{R}_F \mid \mathcal{R}_0 = A_2]$$

Note that for $i \in \{1,2\}$ we have,

$$\Pr[j \in \mathcal{R}_F | \mathcal{R}_0 = A_i] = \Pr[j \in \mathcal{R}_F | j \in \mathcal{R}_2, \mathcal{R}_0 = A_i] \Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_i] \tag{2.8}$$

For both $i = 1$ and $i = 2$, the first term in the RHS of Eq. (2.8) (*i.e.,* $\Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2, \mathcal{R}_0 = A_i]$) is same and is equal to $1/(2d+1)$. Note that the second term in the RHS of Eq. (2.8) (*i.e.,* $\Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_i]$) can be rewritten as $\Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_0, \mathcal{R}_o = A_i]$ since $j \in A_1$ and $\mathcal{R}_0 = A_i$ for $i \in \{1,2\}$. From lines 2 and 3(b) of the algorithm we have that the event $j \in \mathcal{R}_2$ conditioned on $j \in \mathcal{R}_0$ occurs if and only if:

(i) no medium or tiny blocking events occurred for j.

(ii) vertex j did not correspond to an anomalous vertex in G (one with out-degree greater than $d := \alpha + \alpha^{2/3}$).

[11] Specifically, Definition 1.2 and Theorem 1.5 from [32].

Both (i) and (ii) are monotonically decreasing in the set $\mathcal{R}_0$ (*i.e.,* if it holds for $\mathcal{R}_0 = A_2$ then it holds for $\mathcal{R}_0 = A_1$). Hence we have $\Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_1] \geq \Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_2]$.

3 The Stochastic k-Set Packing Problem

Consider the stochastic k-set packing problem defined in the introduction. We start with a second-chance-based algorithm yielding an improved ratio of $8k/5 + o(k)$. We then improve this to $k + o(k)$ via multiple chances. Recall that if we probe an item j, we have to add it irrevocably, as is standard in such stochastic-optimization problems; thus, we do not get multiple opportunities to examine j. Let $\mathbf{x}$ be an optimal solution to the benchmark LP (1.4) and $\mathcal{C}(j)$ be the set of constraints that j participates in.

Bansal *et al.* [12] presented Algorithm 3, SKSP(α). They show that SKSP(α) will add each item j with probability at least $\beta\,(x_j/k)$ where $\beta \geq \alpha(1-\alpha/2)$. By choosing $\alpha = 1$, SKSP(α) yields a ratio of $2k$.[12]

Algorithm 3: SKSP(α) [12]

---

1 Let $\mathcal{R}$ denote the set of chosen items which starts out as an empty set.
2 For each $j \in [n]$, generate an independent Bernoulli random variable Y_j with mean $\alpha x_j/k$.
3 Choose a uniformly random permutation π over $[n]$ and follow π to check each item j one-by-one: add j to $\mathcal{R}$ if and only if $Y_j = 1$ and j is safe (i.e., each resource $i \in \mathcal{C}(j)$ has at least one unit of budget available); otherwise skip j.
4 Return $\mathcal{R}$ as the set of chosen items.

At a high level, our second-chance-based algorithm proceeds as follows with parameters $\{\alpha_1, \beta_1, \alpha_2\}$ to be chosen later. During the first chance, we set $\alpha = \alpha_1$ and run SKSP(α_1). Let $E_{1,j}$ denote the event that j is added to $\mathcal{R}$ in this first chance. From the analysis in [12], we have that $\Pr[E_{1,j}] \geq (x_j/k)\,\alpha_1(1-\alpha_1/2)$. By applying simulation-based attenuation techniques, we can ensure that each item j is added to $\mathcal{R}$ in the first chance with probability *exactly equal* to $\beta_1 x_j/k$ for a certain $\beta_1 \leq \alpha_1(1-\alpha_1/2)$ of our choice.[13] In other words, suppose we obtain an estimate $\hat{E}_{1,j} := \Pr[E_{1,j}]$. When running the original randomized algorithm, whenever j can be added to $\mathcal{R}$ in the first chance, instead of adding it with probability 1, we add it with probability $((x_j/k)\,\beta_1)/\hat{E}_{1,j}$.

In the second chance, we set $\alpha = \alpha_2$ and modify SKSP(α_2) as follows. We generate an independent Bernoulli random variable $Y_{2,j}$ with mean $\alpha_2 x_j/k$ for each j; let $Y_{1,j}$ denote the Bernoulli random variable from the first chance. Proceeding in a uniformly random order π_2, we add j to $\mathcal{R}$ if and only if j is safe, $Y_{1,j} = 0$ and $Y_{2,j} = 1$. Algorithm 4, SKSP($\alpha_1, \beta_1, \alpha_2$), gives a formal description.

Algorithm 4: SKSP($\alpha_1, \beta_1, \alpha_2$)

---

1 Initialize $\mathcal{R}$ as the empty set.
2 **The first chance:** Run SKSP(α_1) with simulation-based attenuation such that $\Pr[E_{1,j}] = \beta_1 x_j/k$ for each $j \in [n]$, with $\beta_1 \leq \alpha_1(1-\alpha_1/2)$. $\mathcal{R}$ now denotes the set of variables chosen in this chance.
3 **The second chance:** Generate an independent Bernoulli random variable $Y_{2,j}$ with mean $\alpha_2 x_j/k$ for each j. Follow a uniformly random order π_2 over $[n]$ to check each item j one-by-one: add j to $\mathcal{R}$ if and only if j is safe, $Y_{1,j} = 0$, and $Y_{2,j} = 1$; otherwise, skip it.
4 Return $\mathcal{R}$ as the set of chosen items.

Lemma 3.1 lower bounds the probability that an item gets added in the second chance. For each j, let $E_{2,j}$ be the event that j is added to $\mathcal{R}$ in the second chance.

LEMMA 3.1. *After running* SKSP($\alpha_1, \beta_1, \alpha_2$) *on an optimal solution* $\mathbf{x}$ *to the benchmark* LP *(1.4), we have*

$$\Pr[E_{2,j}] \geq \tfrac{x_j}{k}\alpha_2\Big(1 - \tfrac{\alpha_1 x_j}{k} - \beta_1 - \tfrac{\alpha_2}{2}\Big)$$

We can use Lemma 3.1 to show that we get an approximation ratio of $8k/5$. Observe that the events $E_{1,j}$ and $E_{2,j}$ are mutually exclusive. Let E_j be the event that j has been added to $\mathcal{R}$ after the two chances. Then, by choosing $\alpha_1 = 1$, $\beta_1 = 1/2$, and $\alpha_2 = 1/2$, we have $\Pr[E_j] = \Pr[E_{1,j}] + \Pr[E_{2,j}] = \frac{5x_j}{8k} - \frac{x_j^2}{2k^2}$. From the Linearity of Expectation, we get that the total expected weight of the solution is at least $(5/8k - o(k))\mathbf{w}.\mathbf{x}$.

THEOREM 3.1. *By choosing* $\alpha_1 = 1$*,* $\beta_1 = 1/2$*, and* $\alpha_2 = 1/2$*,* SKSP($\alpha_1, \beta_1, \alpha_2$) *achieves a ratio of* $\frac{8k}{5} + o(k)$ *for* SKSP.

3.1 Extension to T Chances

Intuitively, we can further improve the ratio by performing a third-chance probing and beyond. We present a natural generalization of $(\alpha_1, \beta_1, \alpha_2)$ to $(\{\alpha_t, \beta_t \mid t \in [T]\})$ with T chances, where

[12] The terminology used in [12] is actually $1/\alpha$; however, we find the inverted notation α more natural for our calculations.

[13] See footnote in Section 1.2. Sampling introduces some small sampling error, but this can be made into a lower-order term with high probability. We thus assume for simplicity that such simulation-based approaches give us exact results.

$\{\alpha_t, \beta_t \mid t \in [T]\}$ are parameters to be fixed later. Note that $(\alpha_1, \beta_1, \alpha_2)$ is the special case wherein $T = 2$.

During each chance $t \le T$, we generate an independent *Bernoulli* random variable $Y_{t,j}$ with mean $\alpha_t x_j/k$ for each j. Then we follow a uniform random order π_t over $[n]$ to check each item j one-by-one: we add j to $\mathcal{R}$ if and only if j is safe, $Y_{t',j} = 0$ for all $t' < t$ and $Y_{t,j} = 1$; otherwise we skip it. Suppose for a chance t, we have that each j is added to $\mathcal{R}$ with probability at least $\beta_t x_j/k$. As before, we can apply simulation-based attenuation to ensure that each j is added to $\mathcal{R}$ in chance t with probability exactly equal to $\beta_t x_j/k$. Notice that to achieve this goal we need to simulate our algorithm over all previous chances up to the current one t. Algorithm 5, SKSP($\{\alpha_t, \beta_t | t \in [T]\}$), gives a formal description of the algorithm. Notice that during the last chance T, we do not need to perform simulation-based attenuation. For the sake of clarity in presentation, we still describe it in the algorithm description.

Algorithm 5: SKSP($\{\alpha_t, \beta_t | t \in [T]\}$)

---

1 Initialize $\mathcal{R}$ as the empty set.
2 **for** *t=1, 2, ..., T* **do**
3 Generate an independent Bernoulli random variable $Y_{t,j}$ with mean $\alpha_t x_j/k$ for each j. Follow a uniform random order π_t over $[n]$ to check each item j one-by-one: add j to $\mathcal{R}$ if and only if j is safe, $Y_{t',j} = 0$ for all $t' < t$, and $Y_{t,j} = 1$; otherwise, skip it.
4 Apply simulation-based attenuation such that each j is added to $\mathcal{R}$ in the t^{th} chance with probability equal to $\beta_t x_j/k$.
5 Return $\mathcal{R}$ as the set of chosen items.

For each item j, let $E'_{t,j}$ be the event that j is added to $\mathcal{R}$ in the t^{th} chance before line 3 of the algorithm for t (*i.e.,* before the start of the t^{th} iteration of the loop). Lemma 3.2 lower bounds the probabilities of these events.

LEMMA 3.2. *After running* $(\{\alpha_t, \beta_t \mid t \in [T]\})$ *on an optimal solution* $\mathbf{x}$ *to the benchmark* LP *(1.4), we have*

$$\Pr[E'_{t,j}] \ge \frac{x_j}{k} \left(\alpha_t \left(1 - \sum_{t'<t} \beta_{t'} - \frac{\alpha_t}{2}\right) - \frac{\alpha_t \sum_{t'<t} \alpha_{t'}}{k}\right)$$

Combining Lemma 3.2 and simulation-based attenuation, we have that for any given $\{\alpha_{t'} \mid t' \le t\}$ and $\{\beta_{t'} \mid t' < t\}$, each item j is added to $\mathcal{R}$ in chance t with probability equal to $\beta_t x_j/k$ for any $\beta_t \le \alpha_t \left(1 - \sum_{t'<t} \beta_{t'} - \alpha_t/2\right) - \alpha_t \sum_{t'<t} \alpha_{t'}/k$. For each j, let $E_{j,t}$ be the event that j is added to $\mathcal{R}$ in chance t and E_j the event that j is added to $\mathcal{R}$ after T chances. From Algorithm 5, we have that the events $\{E_{j,t} \mid t \le T\}$ are mutually exclusive. Thus, $\Pr[E_j] = \sum_{t \le T} \Pr[E_{j,t}] = \sum_{t \le T} \beta_t x_j/k$. Therefore, to maximize the final ratio, we will solve the following optimization problem.

(3.9)
$$\max \sum_{t \in [T]} \beta_t \text{ s.t. } \beta_t \le \alpha_t \left(1 - \sum_{t'<t} \beta_{t'} - \frac{\alpha_t}{2}\right) - \frac{\alpha_t \sum_{t'<t} \alpha_{t'}}{k} \;\forall t \in [T],\; \alpha_t \ge 0 \;\forall t \in [T]$$

Consider a simplified version of maximization program (3.9) by ignoring the $O(1/k)$ term as follows.

(3.10)
$$\max \sum_{t \in [T]} \beta_t, \text{ s.t. } \beta_t \le \alpha_t \left(1 - \sum_{t'<t} \beta_{t'} - \frac{\alpha_t}{2}\right) \;\forall t \in [T],\; \alpha_t \ge 0 \;\forall t \in [T]$$

LEMMA 3.3. *An optimal solution to the program* (3.10) *is*

$$\beta^*_t = \frac{1}{2}(1 - \sum_{t'<t} \beta^*_{t'})^2, \;\forall t \ge 1, \alpha^*_t = 1 - \sum_{t'<t} \beta^*_{t'}, \;\forall t \ge 1$$

where $\beta^*_0 = 0$, $0 \le \alpha^*_t \le 1$ *for all* $t \ge 1$ *and* $\lim_{T \to \infty} \sum_{t=1}^{T} \beta^*_t = 1$.

THEOREM 3.2. *Let* T *be some slowly-growing function of* k*, e.g.,* $T = \log k$*. For each* $t \in [T]$*, set* $\bar{\alpha}_t = \alpha^*_t$, $\bar{\beta}_t = \beta^*_t - \frac{\alpha^*_t (\sum_{t'<t} \alpha^*_{t'})}{k}$. *Then we have (1)* $\{\bar{\alpha}_t, \bar{\beta}_t \mid t \in [T]\}$ *is feasible to the program* (3.9) *and (2)* $\sum_{t \in [T]} \bar{\beta}_t = 1 + o(1)$ *where* $o(1)$ *goes to* 0 *when* k *goes to infinity. Thus,* SKSP($\{\bar{\alpha}_t, \bar{\beta}_t | t \in [T]\}$) *achieves a ratio of* $k + o(k)$ *for* SKSP.

4 Hypergraph Matching

In this section, we give a non-uniform attenuation approach to the hypergraph matching problem which leads to improved competitive ratios. Additionally as stated in the introduction, this takes us "half the remaining distance" towards resolving the stronger Conjecture 2.

Consider a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$. Assume each $e \in \mathcal{E}$ has cardinality $|e| = k_e$. Let $\mathbf{x} = \{x_e\}$ be an optimal solution to the LP (1.5). We start with a warm-up algorithm due to Bansal *et al.* [12][14]. Algorithm 6, HM(α), summarizes their approach.

LEMMA 4.1. *Each edge* e *is added to* $\mathcal{R}$ *with probability at least* $\frac{x_e}{k_e+1}$ *in* HM(α) *with* $\alpha = 1$.

[14]Similar to our algorithm for SKSP, we use the notation α while [12] use $1/\alpha$.

Algorithm 6: $\mathrm{HM}(\alpha)$

1 Initialize $\mathcal{R}$ to be the empty set. We will add edges to this set during the algorithm and return it at the end as the matching.
2 For each $e \in \mathcal{E}$, generate an independent Bernoulli random variable Y_e with mean αx_e.
3 Choose a random permutation π over $\mathcal{E}$ and follow π to check each edge one-by-one: add e to $\mathcal{R}$ if and only if $Y_e = 1$ and e is safe (i.e., none of the vertices in e are matched); otherwise skip it.
4 Return $\mathcal{R}$ as the matching.

It can be shown that in $\mathrm{HM}(\alpha = 1)$, the worst case occurs for the edges e with $x_e \leq \epsilon \approx 0$ (henceforth referred to as "tiny" edges). In contrast, for the edges with $x_e \gg \epsilon$ (henceforth referred to as "large" edges), the ratio is much higher than the worst case bound. This motivates us to balance the ratios among tiny and large edges. Hence, we modify Algorithm 6 as follows: we generate an independent Bernoulli random variable Y_e with mean $g(x_e)$ for each e, where $g : [0,1] \to [0,1]$ and $g(x)/x$ is decreasing over $[0,1]$. Algorithm 7 gives a formal description of this modified algorithm.

Algorithm 7: $\mathrm{HM}(g)$

1 Initialize $\mathcal{R}$ to be the empty set.
2 For each $e \in \mathcal{E}$, generate an independent Bernoulli random variable Y_e with mean $g(x_e)$.
3 Choose a random permutation π over $\mathcal{E}$ and follow π to consider each edge one by one: add e to $\mathcal{R}$ if $Y_e = 1$ and e is safe; otherwise skip it.
4 Return $\mathcal{R}$ as the matching

Observe that $\mathrm{HM}(\alpha)$ is the special case wherein $g(x_e) = \alpha x_e$. We now consider the task of finding the optimal g such that the resultant ratio achieved by $\mathrm{HM}(g)$ is maximized. Consider a given e with $x_e = x$. For any $e' \neq e$, we say e' is a neighbor of e (denoted by $e' \sim e$) if $e' \ni v$ for some $v \in e$. From the LP (1.5), we have $\sum_{e' \sim e} x_{e'} \leq k_e(1-x)$. Let E_e be the event that e is added to $\mathcal{R}$. By applying an analysis similar to the proof of Lemma 4.1, we get the probability of E_e is at least

$$\Pr[E_e] \geq g(x) \int_0^1 \prod_{e' \sim e} (1 - t g(x_{e'})) dt. \tag{4.11}$$

Therefore our task of finding an optimal g to maximize the r.h.s. of (4.11) is equivalent to finding $\max_g \mathcal{F}(g)$, where $\mathcal{F}(g)$ is defined in equation (4.12).

$$\mathcal{F}(g) \doteq \min_{x \in [0,1]} \left[\frac{g(x)}{x} \times r(x) \right] \tag{4.12}$$

In equation (4.12), $r(x)$ is defined as

$$r(x) \doteq \min \int_0^1 \prod_{e' \sim e} \left(1 - t g(x_{e'})\right) dt, \quad \text{where } \sum_{e' \sim e} x_{e'} \leq k_e(1-x), x_{e'} \in [0,1], \forall e'$$

LEMMA 4.2. *By choosing $g(x) = x(1 - \frac{x}{2})$, we have that the minimum value of $\mathcal{F}(g)$ in Eq. (4.12) is $\mathcal{F}(g) = \frac{1}{k_e}(1 - \exp(-k_e))$.*

We now prove the main result, Theorem 1.3.

Proof. Consider $\mathrm{HM}(g)$ as shown in Algorithm 7 with $g(x) = x(1 - x/2)$. Let $\mathcal{R}$ be the random matching returned. From Lemma 4.2, we have that each e will be added to $\mathcal{R}$ with probability at least $x_e \mathcal{F}(g) = \frac{x_e}{k_e}(1 - \exp(-k_e))$.

5 More Applications

In this section, we briefly describe how a simple simulation-based attenuation can lead to improved contention resolution schemes for UFP-TREES with unit demands. This version of the problem was studied by Chekuri *et al.* [30] where they gave a 4-approximation for the linear objective case. They also described a simple randomized algorithm that obtains a 27-approximation[15]. Later, Chekuri *et al.* [32] developed the machinery of contention resolution schemes, through which they extended it to a $27/\eta_f$-approximation algorithm for non-negative submodular objective functions (where η_f denotes the approximation ratio for maximizing non-negative submodular functions[16]). We show that using simple attenuation ideas can further improve this $27/\eta_f$-approximation to an $8.15/\eta_f$-approximation. We achieve this by improving the $1/27$-balanced CR scheme[17] to a $1/8.15$-balanced CR scheme and hence, from Theorem 1.5 of [32], the approximation ratio follows.

Consider the natural packing LP relaxation. Associate a variable x_i with every demand pair. Our constraint set is: for every edge e, $\sum_{i: e \in \mathcal{P}_i} x_i \leq u_e$, where u_e is the capacity of e. Our algorithm (formally described in Algorithm 8) proceeds similar to the one described in [30], except at line 3, where we use our attenuation ideas.

Analysis. For the most part, the analysis is similar to the exposition in Chekuri *et al.* [32]. We will highlight the part where *attenuation* immediately leads to improved bounds.

[15] This can be obtained by maximizing over all $0 < b < 1/3e$ in Lemma 4.19 of [32], which yields approximately $1/27$.

[16] See the section on k-CS-PIP for a discussion on various values of η_f known.

[17] See the section on extension to sub-modular objectives in k-CS-PIP for defintion of a balanced CR scheme.

Algorithm 8: Improved contention resolution Scheme UFP-TREES with unit demands

1 Root the tree T arbitrarily. Let $\mathrm{LCA}(s_i, t_i)$ denote the least common ancestor of s_i and t_i.
2 Construct a random set $\mathcal{R}$ of demand pairs, by including each pair in $\mathcal{R}$ independently with probability αx_i.
3 Consider the demand pairs in increasing distance of their LCA from the root. Let $\mathcal{R}_{\text{final}}$ denote the set of pairs included in the rounded integral solution. For every demand pair i, simulate the run of the algorithm from beginning (*i.e.,* produce many samples of $\mathcal{R}$ and separately run the algorithm on these samples) to obtain the estimate η_i of the probability of i to be safe (*i.e.,* none of the edges in the path has exhausted capacities). Suppose i is safe, add it to $\mathcal{R}_{\text{final}}$ with probability β/η_i.
4 Return $\mathcal{R}_{\text{final}}$ as the set of demands chosen from routing.

Consider a fixed pair (s_{i^*}, t_{i^*}) and let $\ell := \mathrm{LCA}(s_{i^*}, t_{i^*})$ in T. Let P and P' denote the unique path in the tree from ℓ to s_{i^*} and ℓ to t_{i^*} respectively. As in [32], we will upper bound the probability of i^* being unsafe due to path P and a symmetric argument holds for P'. Let $e_1, e_2, \ldots, e_\lambda$ be the edges in P from ℓ to s_{i^*}. Let E_j denote the event that i^* is not safe to be added in line 3 of Algorithm 8, because of overflow at edge e_j. Note that for $j > h$ and $u_{e_j} \geq u_{e_h}$, event E_j implies E_h and hence $\Pr[E_j] \leq \Pr[E_h]$. Note, this argument does not change due to attenuation since the demands are processed in increasing order of the depth and any *chosen* demand pair using edge e_j also has to use e_h up until the time i^* is considered. Thus, we can make a simplifying assumption similar to [32] and consider a strictly decreasing sequence of capacities $u_{e_1} > u_{e_2} > \ldots > u_{e_\lambda} \geq 1$. Let $\mathcal{S}_j$ denote the set of demand pairs that use edge e_j. The following steps is the part where our analysis differs from [32] due to the introduction of attenuation.

Define, $\beta := 1 - 2\alpha e/(1-\alpha e)$ and $\gamma := \alpha\beta$. Note that without attenuation, we have $\eta_i \geq \beta$ for all i from the analysis in [32].

Let E'_j denote the event that at least u_{e_j} demand pairs out of $\mathcal{S}_j$ are included in the final solution. Note that $\Pr[E_j] \leq \Pr[E'_j]$. From the LP constraints we have $\sum_{i \in \mathcal{S}_j} x_i \leq u_{e_j}$.

Let X_i denote the indicator random variable for the following event: $\{i \in \mathcal{R} \wedge i \in \mathcal{R}_{\text{final}}\}$. We define $X := \sum_{i \in \mathcal{S}_j} X_i$.

Note that, event E'_j happens if and only if $X \geq u_{e_j}$ and hence we have, $\Pr[E'_j] = \Pr[X \geq u_{e_j}]$. Additionally, we have that the X_i's are "cylindrically negatively correlated" and hence we can apply the Chernoff-Hoeffding bounds due to Panconesi and Srinivasan [57]. Observe that $\mathbb{E}[X] \leq \gamma u_{e_j}$(since each i is included in $\mathcal{R}$ independently with αx_i and then included in $\mathcal{R}_{\text{final}}$ with probability exactly β) and for $1+\delta = 1/\gamma$, we have $\Pr[X \geq u_{e_j}] \leq (e/(1+\delta))^{(1+\delta)\mu} \leq (\gamma e)^{u_{e_j}}$. Hence, taking a union bound over all the edges in the path, we have the probability of i^* being unsafe due to an edge in P to be at most $\sum_{q=1}^{\infty} (\gamma e)^{\ell} = (\gamma e)/(1-\gamma e)$ (We used the fact that $u_{e_1} > u_{e_2} > \ldots > u_{e_\lambda} \geq 1$). Combining the symmetric analysis for the other path P', we have the probability of i^* being unsafe to be at most $2\gamma e/(1-\gamma e)$. Note that we used the fact that $\gamma e < 1$ in the geometric series. Additionally, since $\gamma \leq 1$, we have that $2\gamma e/(1-\gamma e) \leq 2\alpha e/(1-\alpha e)$. Hence, using $\eta_i \geq \beta$ is justified.

Now to get the claimed approximation ratio, we solve the following maximization problem: $\max_{0 \leq \alpha \leq 1}\{\alpha \cdot (1 - 2\gamma e/(1-\gamma e)) : \beta = 1 - 2\alpha e/(1-\alpha e), \gamma = \alpha * \beta, 0 \leq \gamma e < 1/3\}$ which yields a value of $1/8.15$.

6 Conclusion and Open Problems

In this work, we described two unifying ideas, namely non-uniform attenuation and multiple-chance probing to get bounds matching integrality gap (up to lower-order terms) for the k-CS-PIP and its stochastic counterpart SKSP. We generalized the conjecture due to Füredi *et al.* [39] (FKS conjecture) and went "halfway" toward resolving this generalized form using our ideas. Finally, we showed that we can improve the contention resolution schemes for UFP-TREES with unit demands. Our algorithms for k-CS-PIP can be extended to non-negative submodular objectives via the machinery developed in Chekuri *et al.* [32] and the improved contention resolution scheme for UFP-TREES with unit demands leads to improved approximation ratio for submodular objectives via the same machinery.

This work leaves a few open directions. The first concrete problem is to completely resolve the FKS conjecture and its generalization. We believe non-uniform attenuation and multiple-chances combined with the primal-dual techniques from [39] could give the machinery needed to achieve this. Other open directions are less well-formed. One is to obtain stronger LP relaxations for the k-CS-PIP and its stochastic counterpart SKSP such that the integrality gap is reduced. The other is to consider improvements to related packing programs, such as column-restricted packing programs or general packing programs.

7 Proofs

7.1 Proofs for Section 2 (k-CS-PIP) We will now provide the missing proofs for k-CS-PIP.

Proof of Lemma 2.1. We will prove this Lemma by giving a coloring algorithm that uses at most $2d+1$ colors and prove its correctness. Recall that we have a directed graph such that the maximum out-degree $\Delta \leq d$. The algorithm is a simple greedy algorithm, which first picks the vertex with minimum *total* degree (*i.e.,* sum of in-degree plus out-degree). It then removes this vertex from the graph and recursively colors the sub-problem. Finally, it assigns a color to this picked vertex not assigned to any of its neighbors. Algorithm 9 describes the algorithm formally.

Algorithm 9: Greedy algorithm to color bounded degree directed graph

`Color-Directed-Graph`(G, V, d, χ)
1 **if** $V = \phi$ **then**
2   **return** χ
3 **else**
4   Let $v_{\min}$ denote the vertex with minimum total degree.
5   χ = `Color-Directed-Graph`$(G, V \setminus \{v_{\min}\}, d, \chi)$.
6   Pick the smallest color $c \in [2d+1]$ that is not used to color any of the neighbors of $v_{\min}$. Let $\chi(v_{\min}) = c$.
7   **return** χ

We will now prove the correctness of the above algorithm. In particular, we need to show that in every recursive call of the function, there is always a color $c \in [2d+1]$ such that the assignment in line 6 of the algorithm is feasible. We prove this via induction on number of vertices in the graph G.

Base Case: The base case is the first iteration is when the number of vertices is 1. In this case, the statement is trivially true since $v_{\min}$ has no neighbors.

Inductive Case: We have that $\Delta \leq d$ for every recursive call. Hence, the sum of *total* degree of all vertices in the graph is $2nd$ (Each edge contributes 2 towards the total degree and there are nd edges). Hence, the average total degree is $2d$. This implies that the minimum total degree in the graph is at most $2d$. Hence, the vertex $v_{\min}$ has a total degree of at most $2d$. From inductive hypothesis we have that $V \setminus \{v_{\min}\}$ can be colored with at most $2d+1$ colors. Hence, there exists a color $c \in [2d+1]$, such that $\chi(v_{\min}) = c$ is a valid coloring (since $v_{\min}$ has at most $2d$ neighbors).

Proof of Lemma 2.2. Consider an item $j \in \mathcal{R}_1$. We want to show that the $\Pr[\delta_j > \alpha + \alpha^{2/3}] \leq o(1)$, where δ_j represents the out-degree of j in the directed graph G. Recall that from the construction of graph G, we have a directed edge from item j to item j' if and only there is a constraint i where $a_{ij'} > 1/2$ and $a_{ij} > 0$. For sake of simplicity, let $\{1, 2, \ldots, N_j\}$ denote the set of out-neighbors of j in graph G and let $\{X_1, X_2, \ldots, X_{N_j}\}$ denote the corresponding indicator random variable for them being included in $\mathcal{R}_1$. Hence, for evey $i \in [N_j]$ we have $\mathbb{E}[X_i] = \alpha(1-o(1))x_i/k$. From the strengthened constraints in LP (2.7) we have that $\mathbb{E}[\delta_j] = \mathbb{E}[X_1 + X_2 + \ldots + X_{N_j}] \leq \alpha(1-o(1))$. Hence we have

$$\begin{aligned}\Pr[\delta_j > \alpha + \alpha^{2/3}] &\leq \Pr[\delta_j > \mathbb{E}[\delta_j] + \alpha^{2/3}] \\ &\leq e^{-\Omega(\alpha^{1/3})} = o(1)\end{aligned}$$

The last inequality is from the Chernoff bounds, while the last equality is true for $\alpha = \log(k)$.

Proof of Lemma 2.3. Consider the medium blocking event $\mathrm{MB}(j)$. Let i be a constraint that causes $\mathrm{MB}(j)$ and let $j_1, j_2 \neq j$ be the two other variables such that $j_1, j_2 \in \mathrm{med}(i)$. Denote, X_j, X_{j_1} and X_{j_2} to be the indicators that $j \in \mathcal{R}_0, j_1 \in \mathcal{R}_0, j_2 \in \mathcal{R}_0$ respectively.

We know that $a_{ij}x_j + a_{ij_1}x_{j_1} + a_{ij_2}x_{j_2} \leq 1$ and since $j, j_1, j_2 \in \mathrm{med}(i)$, we have $x_j + x_{j_1} + x_{j_2} \leq \ell$ for some constant value of ℓ. The probability that scenario MB is "bad" is if $X_j + X_{j_1} + X_{j_2} \geq 2$. Note that $\mathbb{E}[X_j + X_{j_1} + X_{j_2}] \leq \frac{\alpha\ell}{k}$.

Hence, using the the Chernoff bounds in the form denoted in Theorem A.1 of Appendix, we have

$$\begin{aligned}\Pr[X_{j_1} + X_{j_2} \geq 2 \mid X_{j_1} = 1] &= \Pr[X_{j_1} + X_{j_2} \geq 2] \\ &\leq O\left(\frac{\alpha^2\ell^2}{k^2}\right)\end{aligned}$$

Note that the first equality is due to the fact that these variables are independent. Using a union bound over the k constraints j appears in, the total probability of the "bad" event is at most $O(\frac{\alpha^2\ell^2}{k})$. And since $\alpha = O(\log k)$ and $\ell = \Theta(k^{1/3})$, this value is $o(1)$.

For scenario $\mathrm{TB}(j)$ we will do the following. If j is tiny, our "bad" event for constraint i is that the total size of remaining items to be at least $1 - 1/\ell$. We know that $\mathbb{E}[\sum_{h \in \mathcal{R} \setminus \{j\}} A_{ih}X_{ih}] \leq \alpha/k$. We want,

$$\begin{aligned}\Pr[\textstyle\sum_{h \in \mathcal{R} \setminus \{j\}} A_{ih}X_{ih} > 1 - 1/\ell] &= \\ \Pr[\textstyle\sum_{h \in \mathcal{R} \setminus \{j\}} \ell A_{ih}X_{ih} > \ell - 1]\end{aligned}$$

Note that $\ell A_{ih}X_{ih} \in [0, 1]$. Hence, using the standard form of the Chernoff bounds, we obtain

$$\Pr[\sum_{h\in\mathcal{R}\setminus\{j\}} \ell A_{ih}X_{ih} > \ell-1] \leq \exp[-(k/(\ell\alpha))(\ell-1)^2]$$

Note that taking a union bound over the k constraints, setting $\ell = k^{1/3}$ and $\alpha = \log(k)$, we have that $k\exp[-k/(\ell\alpha)(\ell-1)^2] = o(1)$.

Proof of Theorem 2.3. To prove this theorem, we will now describe a modified version of Algorithm 9. The other steps in the proof remain the same and will directly imply the theorem. To this end, we will now describe the modified algorithm and its analysis.

Algorithm 10: Greedy algorithm to color bounded out-degree directed graphs using $2d + d^{1-\epsilon}$ colors, and with near-negative correlation

---

`Color-Dir-Graph-Neg-Corr`$(G, V, d, \epsilon, \chi)$

if $V = \phi$ **then**
  return χ
else
  Let $v_{\min}$ denote the vertex with minimum total degree.
  χ = `Color-Dir-Graph-Neg-Corr`$(G, V \setminus \{v_{\min}\}, d, \epsilon, \chi)$.
  Among the *smallest* $d^{1-\epsilon}$ colors in $[2d + d^{1-\epsilon}]$ that have not been used thus far to color any of the neighbors of $v_{\min}$, choose a color c_r *uniformly* at random. Let $\chi(v_{\min}) = c_r$.
  return χ

As before, we will choose one of the colors c in $[2d+d^{1-\epsilon}]$ uniformly at random and include all vertices which received a color c in the set $\mathcal{R}_F$. Since $\alpha^{1-\epsilon} \leq o(\alpha)$, a similar analysis as before follows to give part (1) of Theorem 2.3. We will now prove part (2). Fix an arbitrary $t \in [n]$ and any t-sized subset $U := \{v_1, v_2, \ldots, v_t\}$ of items in $[n]$. A necessary condition for these items to be present in G' is that they were all chosen into $\mathcal{R}_0$, which happens with probability

$$\prod_{j=1}^{t} \frac{x_{v_j}\alpha}{k}; \tag{7.13}$$

suppose that this indeed happens (all the remaining probability calculations are conditional on this). Note that Algorithm 10 first removes the vertex $v_{\min}$ and then recurses; i.e., it removes the vertices one-by-one, starting with v_{min}. Let σ be the *reverse* of this order, and suppose that the order of vertices in U according to σ is, without loss of generality, $\{v_1, v_2, \ldots, v_t\}$. Recall again that our probability calculations now are conditional on all items in U being present in G'; denote such conditional probabilities by $\Pr'$. Note that $\Pr'[v_1 \in \mathcal{R}_F] = \frac{1}{2d+d^{1-\epsilon}} \leq \frac{1}{2\alpha}$. Next, a moment's reflection shows that for any j with $2 \leq j \leq t$,

$$\Pr{}'[v_j \in \mathcal{R}_F \mid v_1 \in \mathcal{R}_F, v_2 \in \mathcal{R}_F, \ldots, v_{j-1} \in \mathcal{R}_F] \leq \frac{1}{d^{1-\epsilon}} \leq \frac{2d^{\epsilon}}{2\alpha}.$$

Chaining these together and combining with (7.13) completes the proof.

7.2 Proofs for Section 3 (SKSP)

We will now provide the missing proofs for SKSP.

Proof of Lemma 3.1. Let us fix j. Note that "$Y_{1,j} = 0$ and $Y_{2,j} = 1$" occurs with probability $(1 - \alpha_1 x_j/k)(\alpha_2 x_j/k)$. Consider a given $i \in \mathcal{C}(j)$ and let $U_{2,i}$ be the budget usage of resource i when the algorithm reaches j in the random permutation of the second chance.

$$\Pr[E_{2,j}] \tag{7.14}$$
$$= \Pr[Y_{1,j} = 0 \wedge Y_{2,j} = 1] \Pr[E_{2,j} | Y_{1,j} = 0 \wedge Y_{2,j} = 1]$$

$$= \left(1 - \tfrac{\alpha_1 x_j}{k}\right) \tfrac{\alpha_2 x_j}{k} \Pr\left[\textstyle\bigwedge_i (U_{2,i} \leq b_i - 1) \mid Y_{1,j} = 0 \wedge Y_{2,j} = 1\right] \tag{7.15}$$

$$\geq \left(1 - \tfrac{\alpha_1 x_j}{k}\right) \tfrac{\alpha_2 x_j}{k} \left(1 - \textstyle\sum_i \Pr\left[U_{2,i} \geq b_i \mid Y_{1,j} = 0\right]\right) \tag{7.16}$$

$$\geq \tfrac{\alpha_2 x_j}{k} \left(1 - \tfrac{\alpha_1 x_j}{k} - \textstyle\sum_i \Pr\left[U_{2,i} \geq b_i\right]\right) \tag{7.17}$$

Let $X_{1,\ell}$ be the indicator random variable showing if ℓ is added to $\mathcal{R}$ in the first chance and $1_{\{2,\ell\}}$ indicate if item ℓ falls before j in the random order π_2. Thus we have

$$U_{2,i} \leq \textstyle\sum_{\ell \neq j} \mathsf{SI}_{i,\ell} \left(X_{1,\ell} + (1 - Y_{1,\ell}) Y_{2,\ell} 1_{\{2,\ell\}}\right)$$

which implies

$$\mathbb{E}[U_{2,i}] \leq \textstyle\sum_{\ell \neq j} u_{i,\ell} \left(\frac{\beta_1 x_\ell}{k} + \frac{1}{2}\left(1 - \frac{\alpha_1 x_\ell}{k}\right)\frac{\alpha_2 x_\ell}{k}\right) \leq \left(\frac{\beta_1}{k} + \frac{1}{2}\frac{\alpha_2}{k}\right) b_i$$

Plugging the above inequality into (7.17) and applying Markov's inequality, we complete the proof of Lemma 3.1.

Proof of Theorem 3.1. Consider a given item j. We have,

$$\begin{aligned} \Pr[E_j] &= \Pr[E_{1,j}] + \Pr[E_{2,j}] \\ &\geq \tfrac{x_j}{k}\left(\beta_1 + \alpha_2\left(1 - \tfrac{\alpha_1 x_j}{k} - \beta_1 - \tfrac{\alpha_2}{2}\right)\right) \\ &= \tfrac{x_j}{k}\left(\beta_1 + \alpha_2\left(1 - \beta_1 - \tfrac{\alpha_2}{2}\right) - O(1/k)\right) \end{aligned}$$

To obtain the worst case, we solve the following optimization problem for an (imagined) adversary:

$$\text{(7.18)} \quad \max\ \beta_1 + \alpha_2\left(1-\beta_1-\tfrac{\alpha_2}{2}\right), \\ \text{s.t. } 0 \le \beta_1 \le \alpha_1(1-\alpha_1/2), \alpha_1 \ge 0, \alpha_2 \ge 0$$

Solving the above program, the optimal solution is $\alpha_1 = 1, \beta_1 = 1/2$ and $\alpha_2 = 1/2$ with a ratio of $(\frac{8}{5}+o(1))k$.

Proof of Lemma 3.2. Consider an item j and define Bernoulli random variable $Z_{t,j} = 1$ iff $Y_{t',j} = 0$ for all $t' < t$ and $Y_{t,j} = 1$. Observe that $\mathbb{E}[Z_{t,j}] = \frac{\alpha_t x_j}{k}\prod_{t'<t}\left(1-\frac{\alpha_{t'}x_j}{k}\right)$. Consider a given $i \in \mathcal{C}(j)$ and let $U_{t,j}$ be the budget usage of resource i when the algorithm reaches j in the random permutation during chance t. Thus we have,

$$\begin{aligned}
&\Pr[E'_{t,j}] = \Pr[Z_{t,j}=1]\Pr[E'_{j,t} \mid Z_{t,j}=1] \\
&= \tfrac{\alpha_t x_j}{k}\prod_{t'<t}\left(1-\tfrac{\alpha_{t'}x_j}{k}\right) \\
&\qquad \Pr\left[\textstyle\bigwedge_i (U_{t,i} \le b_i - 1) \mid Z_{t,j}=1\right] \\
&\ge \tfrac{\alpha_t x_j}{k}\prod_{t'<t}\left(1-\tfrac{\alpha_{t'}x_j}{k}\right) \\
&\qquad \left(1-\textstyle\sum_i \Pr\left[U_{t,i} \ge b_i \mid \bigwedge_{t'<t} Y_{t',j}=0\right]\right) \\
&\ge \tfrac{\alpha_t x_j}{k}\left(\prod_{t'<t}\left(1-\tfrac{\alpha_{t'}x_j}{k}\right) - \textstyle\sum_i \Pr\left[U_{t,i} \ge b_i\right]\right)
\end{aligned}$$

Notice that

$$\mathbb{E}[U_{t,j}] \le \sum_{\ell\neq j} u_{i,\ell}\Big(\tfrac{x_\ell}{k}\sum_{t'<t}\beta_{t'} + \\ \tfrac{1}{2}\prod_{t'<t}\left(1-\tfrac{\alpha_{t'}x_\ell}{k}\right)\tfrac{\alpha_t x_\ell}{k}\Big) \le \left(\tfrac{\sum_{t'<t}\beta_{t'}}{k} + \tfrac{1}{2}\tfrac{\alpha_t}{k}\right)b_i$$

By applying Markov's inequality, we get

$$\begin{aligned}
\Pr[E'_{t,j}] &\ge \tfrac{x_j}{k}\alpha_t\left(\prod_{t'<t}\left(1-\tfrac{\alpha_{t'}x_j}{k}\right) - \left(\textstyle\sum_{t'<t}\beta_{t'} + \tfrac{\alpha_t}{2}\right)\right) \\
&\ge \tfrac{x_j}{k}\left(\alpha_t\left(1-\textstyle\sum_{t'<t}\beta_{t'}-\tfrac{\alpha_t}{2}\right) - \tfrac{\alpha_t\sum_{t'<t}\alpha_{t'}}{k}\right)
\end{aligned}$$

Proof of Lemma 3.3. For any given $\{\beta_t \mid 1 \le t < T\}$, we have $\beta_T \le \alpha_T\left(1-\sum_{t'<T}\beta_{t'} - \alpha_T/2\right)$. Thus, the optimal solution is $\beta^*_T = \frac{1}{2}(1-\sum_{1\le t'<T}\beta^*_{t'})^2$ and $\alpha_T = 1-\sum_{1\le t'<T}\beta^*_{t'}$. Thus,

$$\sum_{t\in[T]}\beta_t = \sum_{t<T}\beta_t + \beta^*_T = \tfrac{1}{2}\left(1+\left(\textstyle\sum_{t<T}\beta_t\right)^2\right)$$

Therefore the maximization of $\sum_{t\le T}\beta_t$ is reduced to that of $\sum_{t\le T-1}\beta_t$. By repeating the above analysis for T times, we get our claim for β^*_t and α^*_t. Let $\gamma^*_T = \sum_{t\in[T]}\beta^*_t$. From the above analysis, we see

$$\gamma^*_1 = \tfrac{1}{2}, \quad \gamma^*_t = \tfrac{1}{2}\left(1+(\gamma^*_{t-1})^2\right), \ \forall t \ge 2$$

Since $\gamma^*_1 \le 1$, we can prove that $\gamma^*_t \le 1$ for all t by induction. Notice that $\gamma^*_t - \gamma^*_{t-1} = \frac{1}{2}(1-\gamma^*_{t-1})^2 \ge 0$, which implies that $\{\gamma_t\}$ is a non-decreasing. Since $\{\gamma_t\}$ is non-decreasing and bounded, it has a limit ℓ. The only solution to the equation $\ell = (1+\ell^2)/2$ is $\ell = 1$, and hence $\lim_{T\to\infty}\gamma^*_T = 1$.

Proof of Theorem 3.2. Observe that for each $t \in [T]$, $\bar\beta_t \le \beta^*_t$. Also,

$$\begin{aligned}
\bar\beta_t &= \tfrac{1}{2}\left(1-\textstyle\sum_{t'<t}\beta^*_{t'}\right)^2 - \tfrac{\alpha^*_t\left(\sum_{t'<t}\alpha^*_{t'}\right)}{k} \\
&= \alpha^*_t\left(1-\textstyle\sum_{t'<t}\beta^*_{t'} - \tfrac{\alpha^*_t}{2}\right) - \tfrac{\alpha^*_t\left(\sum_{t'<t}\alpha^*_{t'}\right)}{k} \\
&\le \bar\alpha_t\left(1-\textstyle\sum_{t'<t}\bar\beta_{t'} - \tfrac{\bar\alpha_t}{2}\right) - \tfrac{\bar\alpha_t\left(\sum_{t'<t}\bar\alpha_{t'}\right)}{k}
\end{aligned}$$

Thus we claim that $\{\bar\alpha_t, \bar\beta_t \mid t \in [T]\}$ is feasible for the program (3.9). Notice that

$$\begin{aligned}
\sum_{t\in[T]}\bar\beta_t = \sum_{t\in[T]}\beta^*_t - \sum_{t\in[T]}\tfrac{\alpha^*_t\left(\sum_{t'<t}\alpha^*_{t'}\right)}{k} &\ge \gamma^*_T - \tfrac{T^2}{k} \\
&= 1-\left(1-\gamma^*_T + \tfrac{\log^2 k}{k}\right)
\end{aligned}$$

From Lemma 3.3, we have that $(1-\gamma^*_T) = o(1)$ thus proving the theorem.

7.3 Proofs for Section 4 (Hypergraph Matching)

We will now provide the missing proofs for Hypergraph Matching.

Proof for Lemma 4.1. The proof is very similar to that in [15, 12]. For each $e \in \mathcal{E}$, let E_e be the event that e is added to $\mathcal{R}$ in $\mathrm{HM}(\alpha)$. Consider an edge e and for each $v \in e$, let L_v be the event that v is available when checking e (in the order given by π). Let $\pi(e) = t \in (0,1)$.

$$\begin{aligned}
\Pr[E_e] &= \alpha x_e \int_0^1 \Pr\left[\textstyle\bigwedge_{v\in e} L_v \mid \pi(e)=t\right]dt \\
&\ge \alpha x_e \int_0^1 \prod_{v\in e}\prod_{f:f\ni v}(1-t\alpha x_f)dt \\
&\ge \alpha x_e \int_0^1 (1-t\alpha)^{k_e}dt = \tfrac{x_e}{k_e+1}\left(1-(1-\alpha)^{k_e+1}\right)
\end{aligned}$$

Thus choosing $\alpha = 1$ completes the proof of the lemma.

Proof of Lemma 4.2. Consider a given $x_e = x$ with $|e| = k_e$. Notice that for each given $t \in (0,1)$,

$$\prod_{e'\sim e}\left(1-tg(x_{e'})\right) = \exp\left(\textstyle\sum_{e'\sim e}\ln(1-tg(x_{e'}))\right)$$

Note that $g(x) = x(1-x/2)$ satisfies the condition of Lemma A.1 in the Appendix and hence, for each given $t \in (0,1)$, the function $\ln(1-tg(x))$ is convex over $x \in [0,1]$. Thus to minimize $\sum_{e'\sim e}\ln(1-tg(x_{e'}))\big)$ subject to $0 \le x_{e'} \le 1$ and $\sum_{e'} x_{e'} \le \kappa$ with $\kappa = k_e(1-x)$, an adversary will choose the following worst case scenario: create κ/ϵ neighbors for e with each $x_{e'} = \epsilon$ and let $\epsilon \to 0$. Thus,

$$\begin{aligned}
&\min\prod_{e'\sim e}\left(1-tg(x_{e'})\right) \\
&\quad = \min\exp\left(\textstyle\sum_{e'\sim e}\ln(1-tg(x_{e'}))\right) \\
&\qquad = \lim_{\epsilon\to 0}(1-tg(\epsilon))^{\kappa/\epsilon} = \exp(-t\kappa)
\end{aligned}$$

Therefore for each fixed $x_e = x$, the inner minimization program in (4.12) has an analytic form of optimal value as follows:

(7.19)

$$\min \int_0^1 \prod_{e' \sim e} (1 - tg(x_{e'}))dt = \int_0^1 \exp(-t\kappa)dt = \frac{1 - \exp(-\kappa)}{\kappa}$$

Plugging this back into (4.12), we obtain $\mathcal{F}(g) = \min_{x \in [0,1]} G(x)$, where

$$G(x) \doteq \left(1 - \frac{x}{2}\right) \frac{1}{k_e(1-x)} \Big(1 - \exp(-k_e(1-x))\Big)$$

Note that $G'(x) \geq 0$ over $x \in [0,1]$ and thus, the minimum value of $G(x)$ over $x \in [0,1]$ is $G(0) = \frac{1}{k_e}(1 - \exp(-k_e))$.

Acknowledgements. We would like to thank the anonymous reviewers for valuable comments.

References

[1] ADAMCZYK, M. Non-negative submodular stochastic probing via stochastic contention resolution schemes. *CoRR abs/1508.07771* (2015).

[2] ADAMCZYK, M., GRANDONI, F., AND MUKHERJEE, J. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*. Springer, 2015, pp. 1–12.

[3] ADAMCZYK, M., SVIRIDENKO, M., AND WARD, J. Submodular stochastic probing on matroids. In *31st International Symposium on Theoretical Aspects of Computer Science*, vol. 25 of *LIPIcs. Leibniz Int. Proc. Inform.* 2014, pp. 29–40.

[4] AJTAI, M., KOMLÓS, J., AND SZEMERÉDI, E. A note on Ramsey numbers. *J. Comb. Theory, Ser. A 29*, 3 (1980), 354–360.

[5] ARKIN, E. M., AND HASSIN, R. On local search for weighted k-set packing. *Math. Oper. Res. 23*, 3 (1998), 640–648.

[6] AZAR, Y., AND REGEV, O. Combinatorial algorithms for the unsplittable flow problem. *Algorithmica 44*, 1 (2006), 49–66.

[7] BANSAL, N. Constructive algorithms for discrepancy minimization. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*. IEEE Computer Soc., Los Alamitos, CA, 2010, pp. 3–10.

[8] BANSAL, N., CHAKRABARTI, A., EPSTEIN, A., AND SCHIEBER, B. A quasi-PTAS for unsplittable flow on line graphs. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*. ACM, New York, 2006, pp. 721–729.

[9] BANSAL, N., DADUSH, D., AND GARG, S. An algorithm for komlós conjecture matching banaszczyk's bound. In *IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)* (2016), IEEE, pp. 788–799.

[10] BANSAL, N., FRIGGSTAD, Z., KHANDEKAR, R., AND SALAVATIPOUR, M. R. A logarithmic approximation for unsplittable flow on line graphs. *ACM Trans. Algorithms 10*, 1 (2014), Art. 1, 15.

[11] BANSAL, N., AND GARG, S. Improved algorithmic bounds for discrepancy of sparse set systems. *arXiv preprint arXiv:1601.03311* (2016).

[12] BANSAL, N., GUPTA, A., LI, J., MESTRE, J., NAGARAJAN, V., AND RUDRA, A. When LP is the cure for your matching woes: improved bounds for stochastic matchings. *Algorithmica 63*, 4 (2012), 733–762.

[13] BANSAL, N., KORULA, N., NAGARAJAN, V., AND SRINIVASAN, A. Solving packing integer programs via randomized rounding with alterations. *Theory Comput. 8* (2012), 533–565.

[14] BANSAL, N., AND SPENCER, J. Deterministic discrepancy minimization. *Algorithmica 67*, 4 (2013), 451–471.

[15] BAVEJA, A., CHAVAN, A., NIKIFOROV, A., SRINIVASAN, A., AND XU, P. Improved bounds in stochastic matching and optimization. In *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, vol. 40 of *LIPIcs. Leibniz Int. Proc. Inform.* 2015, pp. 124–134.

[16] BAVEJA, A., AND SRINIVASAN, A. Approximation algorithms for disjoint paths and related routing and packing problems. *Math. Oper. Res. 25*, 2 (2000), 255–280.

[17] BECK, J., AND FIALA, T. "Integer-making" theorems. *Discrete Appl. Math. 3*, 1 (1981), 1–8.

[18] BERMAN, P. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. *Nordic J. Comput. 7*, 3 (2000), 178–184.

[19] BRUBACH, B., SANKARARAMAN, K. A., SRINIVASAN, A., AND XU, P. Attenuate locally, win globally: An attenuation-based framework for online stochastic matching with timeouts. In *Sixteenth*

International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017). 2017.

[20] BUCHBINDER, N., AND FELDMAN, M. Constrained submodular maximization via a non-symmetric technique. *arXiv preprint arXiv:1611.03253* (2016).

[21] BUCHBINDER, N., FELDMAN, M., NAOR, J., AND SCHWARTZ, R. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms* (2014), ACM, New York, pp. 1433–1452.

[22] CALINESCU, G., CHEKURI, C., PÁL, M., AND VONDRÁK, J. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput. 40*, 6 (2011), 1740–1766.

[23] CHAKRABARTI, A., CHEKURI, C., GUPTA, A., AND KUMAR, A. Approximation algorithms for the unsplittable flow problem. *Algorithmica 47*, 1 (2007), 53–78.

[24] CHAN, Y. H., AND LAU, L. C. On linear and semidefinite programming relaxations for hypergraph matching. *Math. Program. 135*, 1-2, Ser. A (2012), 123–148.

[25] CHANDRA, B., AND HALLDÓRSSON, M. M. Greedy local improvement and weighted set packing approximation. *J. Algorithms 39*, 2 (2001), 223–240.

[26] CHEKURI, C., ENE, A., AND KORULA, N. Personal communication. 537–538.

[27] CHEKURI, C., ENE, A., AND KORULA, N. Unsplittable flow in paths and trees and column-restricted packing integer programs. In *Approximation, randomization, and combinatorial optimization*, vol. 5687 of *Lecture Notes in Comput. Sci.* Springer, Berlin, 2009, pp. 42–55.

[28] CHEKURI, C., JAYRAM, T. S., AND VONDRÁK, J. On multiplicative weight updates for concave and submodular function maximization. In *ITCS'15—Proceedings of the 6th Innovations in Theoretical Computer Science*. ACM, New York, 2015, pp. 201–210.

[29] CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory Comput. 2* (2006), 137–146.

[30] CHEKURI, C., MYDLARZ, M., AND SHEPHERD, F. B. Multicommodity demand flow in a tree and packing integer programs. *ACM Trans. Algorithms 3*, 3 (2007), Art. 27, 23.

[31] CHEKURI, C., VONDRÁK, J., AND ZENKLUSEN, R. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*. IEEE Computer Soc., Los Alamitos, CA, 2010, pp. 575–584.

[32] CHEKURI, C., VONDRÁK, J., AND ZENKLUSEN, R. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput. 43*, 6 (2014), 1831–1879.

[33] DEAN, B. C., GOEMANS, M. X., AND VONDRÁK, J. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2005), ACM, New York, pp. 395–404.

[34] DEAN, B. C., GOEMANS, M. X., AND VONDRÁK, J. Approximating the stochastic knapsack problem: the benefit of adaptivity. *Math. Oper. Res. 33*, 4 (2008), 945–964.

[35] ELBASSIONI, K., GARG, N., GUPTA, D., KUMAR, A., NARULA, V., AND PAL, A. Approximation algorithms for the unsplittable flow problem on paths and trees. In *32nd International Conference on Foundations of Software Technology and Theoretical Computer Science*, vol. 18 of *LIPIcs. Leibniz Int. Proc. Inform.* 2012, pp. 267–275.

[36] ENE, A., AND NGUYEN, H. L. Constrained submodular maximization: Beyond 1/e. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on* (2016), IEEE, pp. 248–257.

[37] FELDMAN, M., NAOR, J., AND SCHWARTZ, R. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*. IEEE Computer Soc., Los Alamitos, CA, 2011, pp. 570–579.

[38] FISHER, M. L., NEMHAUSER, G. L., AND WOLSEY, L. A. An analysis of approximations for maximizing submodular set functions. II. *Math. Programming Stud.*, 8 (1978), 73–87. Polyhedral combinatorics.

[39] FÜREDI, Z., KAHN, J., AND SEYMOUR, P. D. On the fractional matching polytope of a hypergraph. *Combinatorica 13*, 2 (1993), 167–180.

[40] GARG, N., VAZIRANI, V. V., AND YANNAKAKIS, M. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica 18*, 1 (1997), 3–20.

[41] GUPTA, A., NAGARAJAN, V., AND SINGLA, S. Adaptivity gaps for stochastic probing: Submodular and XOS functions. *CoRR abs/1608.00673* (2016).

[42] GUPTA, A., NAGARAJAN, V., AND SINGLA, S. Algorithms and adaptivity gaps for stochastic probing. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms* (Philadelphia, PA, USA, 2016), SODA '16, Society for Industrial and Applied Mathematics, pp. 1731–1747.

[43] HARVEY, N. J. A., SCHWARTZ, R., AND SINGH, M. Discrepancy without partial colorings. In *Approximation, randomization, and combinatorial optimization*, vol. 28 of *LIPIcs. Leibniz Int. Proc. Inform.* 2014, pp. 258–273.

[44] HAZAN, E., SAFRA, S., AND SCHWARTZ, O. On the complexity of approximating k-set packing. *Computational Complexity 15*, 1 (2006), 20–39.

[45] HURKENS, C. A. J., AND SCHRIJVER, A. 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 (1989), 68–72.

[46] KLEINBERG, J. M. *Approximation algorithms for disjoint paths problems*. PhD thesis, Citeseer, 1996.

[47] KOLLIOPOULOS, S. G., AND STEIN, C. Approximating disjoint-path problems using packing integer programs. *Math. Program. 99*, 1, Ser. A (2004), 63–87.

[48] KOLMAN, P. A note on the greedy algorithm for the unsplittable flow problem. *Inform. Process. Lett. 88*, 3 (2003), 101–105.

[49] KOLMAN, P., AND SCHEIDELER, C. Improved bounds for the unsplittable flow problem. *J. Algorithms 61*, 1 (2006), 20–44.

[50] KULIK, A., SHACHNAI, H., AND TAMIR, T. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009), SIAM, Philadelphia, PA, pp. 545–554.

[51] LEE, J., MIRROKNI, V. S., NAGARAJAN, V., AND SVIRIDENKO, M. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM J. Discrete Math. 23*, 4 (2009/10), 2053–2078.

[52] LEIGHTON, T., LU, C.-J., RAO, S., AND SRINIVASAN, A. New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM J. Comput. 31*, 2 (2001), 626–641.

[53] LEVY, A., RAMADAS, H., AND ROTHVOSS, T. Deterministic discrepancy minimization via the multiplicative weight update method. *arXiv preprint arXiv:1611.08752* (2016).

[54] LI, J., AND YUAN, W. Stochastic combinatorial optimization via Poisson approximation. In *Proceedings of the 2013 ACM Symposium on Theory of Computing—STOC'13*. ACM, New York, 2013, pp. 971–980.

[55] LOVETT, S., AND MEKA, R. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput. 44*, 5 (2015), 1573–1582.

[56] OVEIS GHARAN, S., AND VONDRÁK, J. Submodular maximization by simulated annealing. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms* (2011), SIAM, Philadelphia, PA, pp. 1098–1116.

[57] PANCONESI, A., AND SRINIVASAN, A. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM J. Comput. 26*, 2 (1997), 350–368.

[58] PAREKH, O. Iterative packing for demand and hypergraph matching. In *Integer programming and combinatorial optimization*, vol. 6655 of *Lecture Notes in Comput. Sci.* Springer, Heidelberg, 2011, pp. 349–361.

[59] PAREKH, O., AND PRITCHARD, D. Generalized hypergraph matching via iterated packing and local ratio. In *Approximation and online algorithms*, vol. 8952 of *Lecture Notes in Comput. Sci.* Springer, Cham, 2015, pp. 207–223.

[60] PERES, Y., SINGH, M., AND VISHNOI, N. Random walks in polytopes and negative dependence. In *Proceedings of the Eighth Annual Innovations in Theoretical Computer Science (Berkeley, CA, 2017)* (2017).

[61] PRITCHARD, D. Approximability of sparse integer programs. In *Algorithms—ESA 2009*, vol. 5757 of *Lecture Notes in Comput. Sci.* Springer, Berlin, 2009, pp. 83–94.

[62] PRITCHARD, D., AND CHAKRABARTY, D. Approximability of sparse integer programs. *Algorithmica 61*, 1 (2011), 75–93.

[63] RÖDL, V. On a packing and covering problem. *Eur. J. Comb. 6*, 1 (1985), 69–78.

[64] ROTHVOSS, T. Constructive Discrepancy Minimization for Convex Sets. *SIAM J. Comput. 46*, 1 (2017), 224–234.

[65] SRINIVASAN, A. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on* (1997), IEEE, pp. 416–425.

[66] SRINIVASAN, A. Improving the discrepancy bound for sparse matrices: better approximations for sparse lattice approximation problems. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (1997), ACM, New York, pp. 692–701.

[67] SRINIVASAN, A. Improved approximation guarantees for packing and covering integer programs. *SIAM J. Comput. 29*, 2 (1999), 648–670.

[68] VONDRÁK, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC'08*. ACM, New York, 2008, pp. 67–74.

[69] WARD, J. A $(k+3)/2$-approximation algorithm for monotone submodular k-set packing and general k-exchange systems. In *29th International Symposium on Theoretical Aspects of Computer Science*, vol. 14. 2012, pp. 42–53.

[70] ZUCKERMAN, D. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput. 3* (2007), 103–128.

A Technical Lemmas

In our main section, our algorithm for k-CS-PIP uses the following version of the Chernoff-Hoeffding bounds. This can be easily derived from the standard form and we give a proof here for completeness.

THEOREM A.1. (CHERNOFF-HOEFFDING BOUNDS) *Suppose c_1, c_2, and k are positive values with $c_2 \geq \frac{c_1}{k}$. Let $X_1, X_2, \ldots, X_n \in [0,1]$ be independent random variables satisfying $\mathbb{E}[\sum_i X_i] \leq \frac{c_1}{k}$. Then,*

$$\Pr[\textstyle\sum_i X_i \geq c_2] \leq \left(\frac{c_1 e}{k c_2}\right)^{c_2}$$

Proof. The standard form of the Chernoff-Hoeffding bounds yields $\Pr[\sum_i X_i \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$, for $\delta \geq 0$. Note that we want $(1+\delta)(c_1/k) = c_2$, hence giving $1+\delta = \frac{kc_2}{c_1}$. Plugging this into the standard form of the Chernoff-Hoeffding bounds gives us the desired bound.

LEMMA A.1. (CONVEXITY) *Assume $f : [0,1] \to [0,1]$ and it has second derivatives over $[0,1]$. Then we have that $\ln(1-tf(x))$ is a convex function of $x \in [0,1]$ for any given $t \in (0,1)$ iff $(1-f)(-f'') \geq f'^2$ for all $x \in [0,1]$.*

Proof. Consider a given $t \in (0,1)$ and let $F(x) = \ln(1-tf(x))$. $F(x)$ is convex over $[0,1]$ iff $F'' \geq 0$ for all $x \in [0,1]$. We can verify that it is equivalent to the condition that $(1-f)(-f'') \geq f'^2$ for all $x \in [0,1]$.

B Submodular Functions

In this section, we give the required background needed for submodular functions.

DEFINITION 2. (SUBMODULAR FUNCTIONS) *A function $f : 2^{[n]} \to \mathbb{R}_+^+$ on a ground-set of elements $[n] := \{1, 2, \ldots, n\}$ is called submodular if for every $A, B \subseteq [n]$, we have that $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$. Additionally, f is said to be monotone if for every $A \subseteq B \subseteq [n]$, we have that $f(A) \leq f(B)$.*

For our algorithms, we assume a *value-oracle* access to a submodular function. This means that, there is an oracle which on querying a subset $T \subseteq [n]$, returns the value $f(T)$.

DEFINITION 3. (MULTI-LINEAR EXTENSION) *The multi-linear extension of a submodular function f is the continuous function $F : [0,1]^n \to \mathbb{R}_+^+$ defined as*

$$F(x) := \textstyle\sum_{T \subseteq [n]} (\prod_{j \in T} x_j \prod_{j \notin T} (1-x_j)) f(T)$$

Note that the multi-linear function $F(x) = f(x)$ for every $x \in \{0,1\}^n$. The multi-linear extension is a useful tool in maximization of submodular objectives. In particular, the above has the following probabilistic interpretation. Let $S \subseteq [n]$ be a random subset of items where each item $i \in [n]$ is added into S with probability x_i. We then have $F(x) = \mathbb{E}_{S \sim x}[f(S)]$. It can be shown that the two definitions of $F(x)$ are equivalent. Hence, a lower bound on the value of $F(x)$ directly leads to a lower bound on the expected value of $f(S)$.