

Algorithms to Approximate Column-sparse Packing Problems

BRIAN BRUBACH, KARTHIK A. SANKARARAMAN, and ARAVIND SRINIVASAN,

University of Maryland, College Park, USA

PAN XU, University of Maryland, College Park and New Jersey Institute of Technology, USA

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).

CCS Concepts: • **Theory of computation** → **Packing and covering problems**;

Additional Key Words and Phrases: Packing programs, randomized algorithms, approximation algorithms

ACM Reference format:

Brian Brubach, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. 2019. Algorithms to Approximate Column-sparse Packing Problems. *ACM Trans. Algorithms* 16, 1, Article 10 (November 2019), 32 pages.

<https://doi.org/10.1145/3355400>

1 INTRODUCTION

Column-sparse packing problems arise in numerous contexts (e.g., References [2, 13, 14, 17, 18, 28, 29, 44, 51, 52, 57]). 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* [36] 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. [14]) and stochastic k -set packing (SKSP: Bansal et al. [13]), and go “half the remaining distance” to optimal for a major integrality-gap conjecture of Füredi, Kahn, and Seymour on hypergraph matching [44].

Extended abstract of this article (Reference [22]) appeared in SODA’18. Research supported by NSF Awards CNS 1010789, CCF 1422569, CCF-1749864, a gift from Google, and research awards from Adobe and Amazon.

Authors’ addresses: B. Brubach, K. A. Sankararaman, and A. Srinivasan, Brendan Iribe Center for Computer Science and Engineering, University of Maryland, College Park, Maryland, USA 20740; emails: bbrubach@cs.umd.edu, karthik-abinavs@gmail.com, srin@cs.umd.edu; P. Xu, Guttenberg information Technologies Center (GTC), University Heights, Newark, New Jersey, USA 07102; email: pxu@njit.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1549-6325/2019/11-ART10 \$15.00

<https://doi.org/10.1145/3355400>

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

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

here $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ means, as usual, that $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ 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 $\text{Sl}_j \in \mathbb{R}_+^m$ 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" Sl_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 $n^{1-\epsilon}$ [79].

A k -column-sparse packing program (k -CS-PP) refers to a special case of packing programs wherein each size vector Sl_j (a column of \mathbf{A}) takes positive values only on a subset $C(j) \subseteq [m]$ of coordinates with $|C(j)| \leq k$. 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., References [14, 36]).

We show randomized-rounding techniques (including *non-uniform attenuation*, *multiple chances*) that, along with the "nibble method" [5, 70] 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 parameter). For hypergraph matching, we make progress "half the remaining way" towards meeting a classic conjecture of Füredi et al. [44]. 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. [34]) with unit demands and submodular objectives, a problem that 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 [14]):

$$\max\{\mathbf{w} \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \mathbf{x} \in [0, 1]^n\}. \quad (2)$$

Typically, a rounding algorithm takes as input an optimal solution $\mathbf{x} \in [0, 1]^n$ to LP (2)—or one of its relatives—and outputs an integral $\mathbf{X} \in \{0, 1\}^n$ that is feasible for PIP (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. [14]) 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 $o(k/\log k)$ [49]. 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 *k-CS-PIP* can be cast as a special case of PIP shown in (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. [14], we will ultimately use a stronger form of this LP relaxation that incorporates additional valid constraints; see (7) in Section 2.1):

$$\max\{\mathbf{w} \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}, \mathbf{x} \in [0, 1]^n\} \quad \text{where } \mathbf{A}[j] = \mathbf{a}_j. \quad (3)$$

For general PIPs, the best-known approximation bounds are shown in Srinivasan [75]. The problem of *k-CS-PIP*, in its full generality, was first considered by Pritchard [68] and followed by several subsequent works such as Pritchard and Chakrabarty [69] and Bansal et al. [14]. Chekuri et al. [36] 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. Bansal et al. [14] 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 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. [14], a factor of $O(k^2)$ independently due to Chekuri et al. [30]² and Pritchard and Chakrabarty [69], and a factor of $O(2^k \cdot k^2)$ due to Pritchard [68].

THEOREM 1. *There exists a randomized rounding algorithm for *k-CS-PIP* with approximation ratio at most $2k + \Theta(k^{0.8} \text{poly}(\log(k))) = 2k + o(k)$ for linear objectives.*

COROLLARY 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 $\max\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_I \cap [0, 1]^n\}$ (here, $F(\mathbf{x})$ is the multi-linear extension of the sub-modular function f and \mathcal{P}_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. [13] as a way to generalize several stochastic-optimization problems such as Stochastic Matching.⁴ The problem 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 $\mathbf{S}l_j \in \{0, 1\}^m$. The random variables $\{R_j := (W_j, \mathbf{S}l_j) : j \in [n]\}$ 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 $u_{i,j} := \mathbb{E}[\mathbf{S}l_{i,j}]$ for all i, j —where $\mathbf{S}l_{i,j}$ denotes the i th component of $\mathbf{S}l_j$ —and $w_j := \mathbb{E}[W_j]$. Moreover, for each item j , there is a known subset $C(j) \subseteq [m]$ of at most k coordinates such that $\mathbf{S}l_{i,j}$ can be nonzero only if $i \in C(j)$: All coordinates in $[m] \setminus C(j)$ 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 that is

¹In Reference [14], the authors also show extensions to non-negative monotone submodular objectives.

²Note that this work is cited in Reference [14].

³To 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 article where the values are always greater than 1.

⁴Here, we use the definition from the journal version [13]; the conference version of Reference [13] defines the problem slightly differently.

⁵Note that W_j can be correlated with $\mathbf{S}l_j$.

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 Sl_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, \dots, w_n) and x_j denote the probability that j is added in the OPT solution. Bansal et al. [13] introduced the following natural LP to upper bound the optimal performance:

$$\max\{\mathbf{w} \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \mathbf{x} \in [0, 1]^n\} \text{ where } A[i, j] = u_{i,j}. \quad (4)$$

The previous best-known bound for SKSP was $2k + o(k)$ due to Bansal et al. [13]. Our main contribution (Theorem 2) is to improve this bound to $k + o(k)$,⁷ a result that is again asymptotically optimal w.r.t. the natural LP (4) considered (Theorem 1.3 from Reference [44]).

THEOREM 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 $k \rightarrow \infty$.*

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:

$$\max\{\mathbf{w} \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{1}, \mathbf{x} \in [0, 1]^n\} \text{ where } A[j] = \mathbf{e}_j. \quad (5)$$

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. [44] presented the following well-known conjecture:

CONJECTURE 1 (FÜREDI ET AL. [44]). *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*

$$\sum_{e \in \mathcal{M}} \left(k_e - 1 + \frac{1}{k_e}\right) w_e \geq \text{OPT}(\mathcal{H}, \mathbf{w}), \quad (6)$$

where k_e denotes the number of vertices in hyperedge e and $\text{OPT}(\mathcal{H}, \mathbf{w})$ denotes an optimal solution to the LP relaxation (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 [44]. 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 (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. In fact, Part (i) of Conjecture 2 and Conjecture 1 are equivalent, which can be shown using an LP-duality

⁶This is called the *safe-policy* assumption. This allows us to handle the correlations between W_j and coordinates of Sl_j . A detailed discussion of this model can be found in Reference [13].

⁷In private communication from January 2018, Marek Adamczyk has informed us that a result of $k + 1$ here can be obtained as a corollary of their recent work [4].

argument for the standard LP for computing the fractional chromatic number (also see Reference [26] for such arguments in a general polytope).

Füredi et al. [44] 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 [28] 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 [64] and Parekh and Pritchard [65] generalized this to k -uniform b -hypergraph matching and obtain the optimal integrality gap for this family. Moreover, they do so for the generalized Conjecture 2. We go “half the remaining distance” in resolving Conjecture 2 for *all* hypergraphs, and also do so algorithmically: The work of Bansal et al. [13] gives “ $k_e + 1$ ” instead of the target $k_e - 1 + 1/k_e$ in Conjecture 2, and we improve this to $k_e + O(k_e \cdot \exp(-k_e))$.

THEOREM 3. *There exists an efficient randomized algorithm to generate a 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 (5) and where the $o(1)$ term is $O(k_e \exp(-k_e))$, a vanishing term when $k_e \rightarrow \infty$.*

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 $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ 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. [34], and the extension to submodular version was given by Chekuri et al. [36]. 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. [36] 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*).⁸ 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 4. *There exists an $8.15/\eta_f$ -approximation algorithm to the UFP-TREES with unit demands and non-negative submodular objectives.*

Extension to submodular objectives. Chekuri et al. [36] 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 11, due to Reference [36]). 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 article to refer to Monte Carlo simulation and the term *simulation-based attenuation* to refer to the simulation and

⁸See Section 2.3 for formal definitions of CR schemes.

attenuation techniques as shown in References [2, 21].⁹ 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 $\sim 1 - c/\hat{E}$, 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 article and the ingredients leading up to them.

Achieving the integrality gap of the LP of Reference [14] for k -CS-PIP. Our first main contribution in this article is to achieve the integrality gap of the strengthened LP of Reference [14] for k -CS-PIP, up to lower-order terms: We improve the $ek + o(k)$ of Reference [14] to $2k + o(k)$. We achieve this by following the same overall structure as in Reference [14] 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 Reference [14], leading to Theorem 1. However, motivated by works that obtain strong “negative correlation” properties—e.g., References [35, 66] obtain *negative cylindrical correlation*¹⁰ and the even-stronger *negative association* for rounding in matroid polytopes—we ask next if one 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 [63]; we use this in Section 5.) We make progress toward this in Theorem 9.

Achieving the integrality gap of the natural LP for SKSP via a “multiple chances” technique. Our second contribution is to develop an algorithm that achieves the integrality gap of $k + o(k)$ for SKSP, improving on the $2k$ of Reference [13]. 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 [5] and Rödl [70]).

The current-best ratios for many special cases of k -CS-PP are $\Theta(k)$; e.g., $ek + o(k)$ for k -CS-PIP [14], the optimal approximation ratio (w.r.t. the integrality gap) of $k - 1 + 1/k$ for k -uniform hypergraph matching [28], or the $2k$ -approximation for SKSP [13]. Thus, many natural approaches involve sampling items with a probability proportional to $1/k$. Consider a k -CS-PP instance with budget b . Suppose we have a randomized algorithm ALG that outputs a solution SOL wherein

⁹This is called “dumping factor” in Reference [2]. See Appendix B in Reference [2] for a formal treatment.

¹⁰This is sometimes simply called “negative correlation”.

each item j is added SOL with probability exactly *equal* to $x_j/(ck)$ for some constant $c > 0$.¹¹ 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.

FKS conjecture and the 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. [13]). 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. [17]): 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 (22). 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 [19] and some of the previous work on k -CS-PIP (e.g., Reference [68]) used techniques similar to Beck and Fiala. There has been a long line of work following Beck and Fiala, including References [8, 10, 12, 15, 48, 58, 60, 71, 74]. One special case of k -CS-PIP is the k -set packing problem. Many works, including References [6, 20, 29, 50], studied this problem, with Reference [20] 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 [52]. Many works have studied this version of packing programs, including References [18, 31, 34].

Similar to Bansal et al. [14], our algorithms also extend to submodular objective functions. In particular, we use tools and techniques from Calinescu et al. [25] and Chekuri et al. [36] for both

¹¹Note that given an algorithm that adds each item with probability *at least* $x_j/(ck)$, we can use Monte Carlo simulation and attenuation techniques similar to References [2, 21, 36] to get an algorithm that adds each item with probability *essentially equal* to $x_j/(ck)$.

k -CS-PIP and the UFP problem on trees. Monotone sub-modular function maximization subject to k -sparse constraints has been studied in the context of k -partition matroids, k -knapsacks, and the intersection of k partition matroids in many works, including References [42, 55, 56, 78]. Beyond the monotone case, there are several algorithms for the non-negative sub-modular maximization problem, including References [23, 24, 32, 40, 41].

Stochastic variants of PIPs have also been previously studied. Baveja et al. [17] 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. [13]) to hypergraphs. The work of Reference [17] 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 References [1, 3, 37, 38, 46, 47, 59].

Later in this article, 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 References [7, 9, 11, 27, 33, 39, 45, 51, 53, 54, 73].

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 and show an extension to sub-modular 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 2. In Section 4, we present an algorithm for hypergraph matching and analyze it to prove Theorem 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 4. We end with a brief conclusion and discussion of open problems in Section 6. Appendix A contains a few useful technical lemmas used in this article, 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. [14] (see (7) in Section 2.1). Theorem 9 then develops a near-negative-correlation generalization of this result.

Recall that we have a k -column-sparse matrix $A \in [0, 1]^{m \times n}$ and a fractional solution $x \in [0, 1]^n$ such that $A \cdot x \leq 1$. Our goal is to obtain an integral solution $X \in \{0, 1\}^n$ (possibly random) such that $A \cdot X \leq 1$ and such that the expected value of the objective function $w \cdot X$ is “large.” (We will later extend this to the case where the objective function $f(X)$ is monotone submodular.) At a very high level, our algorithm performs steps similar to the contention-resolution scheme defined by Chekuri et al. [36]; the main contribution is in the details.¹² 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

¹²We would like to point out that the work of Reference [14] also performs similar steps and fits into the framework of Reference [36].

1. Note that the work of Reference [14] uses *deterministic* alterations. Moving from deterministic alteration to careful randomized alteration, combined with using a much-less aggressive uniform attenuation in the initial independent sampling, yields the optimal bound.

2.1 Algorithm

Before describing the algorithm, we review some useful notations and concepts, some of which were introduced in Reference [14]. Let $\alpha > 0$ be a given parameter.¹³ For a row i of \mathbf{A} and $\ell = \Theta(\log(k/\alpha))$, let $\text{big}(i) := \{j : a_{ij} > 1/2\}$, $\text{med}(i) := \{j : 1/\ell \leq a_{ij} \leq 1/2\}$, and $\text{tiny}(i) := \{j : 0 < a_{ij} < 1/\ell\}$, 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, these events may prevent j from being rounded up. We partition the blocking events into the following three types:

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

$$\sum_{\substack{j' \neq j : \\ j' \in (\text{med}(i) \cup \text{tiny}(i)) \cap \mathcal{R}}} a_{ij'} > 1 - a_{ij} \text{ or } |\text{med}(i) \cap \mathcal{R}| \geq 2.$$

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

The main algorithm of Bansal et al. [14]. As briefly mentioned in Section 1.1, Bansal et al. add certain valid constraints on big items to the natural LP relaxation in (3) as follows:

$$\max\{\mathbf{w} \cdot \mathbf{x} \text{ s.t. } \mathbf{A} \cdot \mathbf{x} \leq \mathbf{1} \text{ and } \forall i \in [m] \sum_{j \in \text{big}(i)} x_j \leq 1, \mathbf{x} \in [0, 1]^n\} \text{ where } \mathbf{A}[j] = \mathbf{a}_j. \quad (7)$$

Algorithm 1, BKNS, gives a formal description of the algorithm of Bansal et al. [14], 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 5 (BANSAL ET AL. [14]). *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 $[n] = \{1, 2, \dots, n\}$ denote the set

¹³We later set it to be a constant (when discussing prior work) and to be $k^{0.4}$ in our algorithm.

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 step 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 $\text{big}(i)$). We claim that the expected out-degree of every vertex in this graph constructed with \mathcal{R}_1 is at most α . If any vertex j has out-degree greater than $d := \alpha + \sqrt{\alpha \log(\alpha)}$, then we will remove j from \mathcal{R}_1 . Hence, we now have a directed graph with every vertex having out-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.

Example. Before moving to the analysis, we will show an example of how the randomized alteration (i.e., steps 3(a–d) of Algorithm 2) works. We will illustrate this on the integrality-gap example considered in Reference [14]. 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 A is defined as follows: $\forall i, j \in [2k - 1]$, we have,

$$a_{ij} := \begin{cases} 1 & \text{if } i = j, \\ \epsilon & \text{if } j \in \{i + 1, i + 2, \dots, i + k - 1 \pmod{n}\}, \\ 0 & \text{otherwise.} \end{cases}$$

As noted in Reference [14], 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 step 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, \dots, 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

$$\alpha = k^{0.4}.$$

THEOREM 1. *There exists a randomized rounding algorithm for k -CS-PIP with approximation ratio at most $2k + \Theta(k^{0.8} \text{poly} \log(k)) = 2k + o(k)$ for linear objectives.*

We will divide the analysis into three parts. The parameters in the algorithm (namely, ℓ, α, d) were chosen such that the $o(k)$ term in the final theorem is minimized. At a high level the three parts prove the following:

ALGORITHM 2: The Algorithm for k -CS-PIP

-
- 1 **Sampling:** Sample each item j independently with probability $\alpha x_j/k$ (where $\alpha = k^{0.4}$) 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 $j' \neq j$ if there exists a constraint i such that $a_{ij} > 0$ and $a_{ij'} > 1/2$.
 - (b) **Removing anomalous vertices:** For every vertex v in G , if the out-degree of v is greater than $d := \alpha + \sqrt{\alpha \log(\alpha)}$, 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 .
-

- **Part 1** (Proved¹⁴ in Lemma 6). 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 7). 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 8). 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 $\alpha = k^{0.4}$). 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 later in this section:

LEMMA 6. *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 7. *For any item j , we have $\Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_1] = 1 - o(1)$.*

LEMMA 8. *For each item j , we have the following:*

$$\Pr[\text{Medium or tiny blocking event occurs for } j \mid j \in \mathcal{R}_0] \leq O(\alpha^2 k^{-1} \log^2(k/\alpha)) = o(1).$$

We can now prove the main theorem, Theorem 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 step 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 step 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.

¹⁴Similar to the main theorem in Reference [76].

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

$$\begin{aligned} \Pr[j \in \mathcal{R}_F] &= \Pr[j \in \mathcal{R}_0] \cdot \Pr[j \in \mathcal{R}_1 \mid j \in \mathcal{R}_0] \cdot \Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_1] \cdot \Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2], \\ &\geq \frac{\alpha x_j}{k} \cdot (1 - o(1)) \cdot (1 - o(1)) \cdot \frac{1}{2\alpha + 2\sqrt{\alpha \log \alpha} + 1}, \\ &= \frac{x_j}{2k(1 + o(1))}. \end{aligned}$$

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 8, we have that $\Pr[j \in \mathcal{R}_1 \mid j \in \mathcal{R}_0] = 1 - o(1)$. Lemma 7 implies that $\Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_1] = 1 - o(1)$. Finally, from Lemma 6, 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, $\Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2] = 1/(2\alpha + 2\sqrt{\alpha \log \alpha} + 1)$ (recall that $d := \alpha + \sqrt{\alpha \log \alpha}$).

We obtain Theorem 1 by using linearity of expectation. In other words, $\mathbb{E}[\mathbf{w} \cdot \mathbf{X}] \geq (\mathbf{w} \cdot \mathbf{x}) \frac{1}{2k(1+o(1))}$. Note that $\mathbf{w} \cdot \mathbf{x}$ is the optimal value to the LP (7). \square

Proof of Lemma 6. 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 that 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 3 describes the algorithm formally.

ALGORITHM 3: Greedy algorithm to color-bounded degree-directed graph

Color-Directed-Graph(G, V, d, χ)

```

1 if  $V = \emptyset$  then
2   return  $\chi$ 
3 else
4   Let  $v_{\min}$  denote the vertex with minimum total degree.
5    $\chi = \text{Color-Directed-Graph}(G, V \setminus \{v_{\min}\}, d, \chi)$ .
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  $\chi$ 
```

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 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 7. Consider an item $j \in \mathcal{R}_1$. We want to show that the $\Pr[\delta_j > \alpha + \sqrt{\alpha \log \alpha}] \leq o(1)$, where δ_j represents the out-degree of j in the directed graph G . Recall that from the con-

struction of graph G , we have a directed edge from item j to item j' if and only if there is a constraint i where $a_{ij'} > 1/2$ and $a_{ij} > 0$. For sake of simplicity, let $\mathcal{S} := \{1, 2, \dots, N_j\}$ denote the set of items that satisfy the following: For every $j' \in \mathcal{S}$, there is some constraint i such that $a_{ij} > 0$ and $a_{ij'} > 1/2$. Let $\{X_1, X_2, \dots, X_{N_j}\}$ denote the corresponding indicator random variable for them being included in \mathcal{R}_1 . Hence, for every $i \in [N_j]$, we have $\mathbb{E}[X_i] = \alpha(1 - o(1))x_i/k$. From the strengthened constraints in LP (7) and the k -column sparsity assumption, we have that $\mathbb{E}[\delta_j] = \mathbb{E}[X_1 + X_2 + \dots + X_{N_j}] \leq \alpha(1 - o(1))$. Hence, we have,

$$\Pr[\delta_j > \alpha + \sqrt{\alpha \log \alpha}] \leq \Pr[\delta_j > \mathbb{E}[\delta_j] + \sqrt{\alpha \log \alpha}] \leq e^{-\Omega(\log \alpha)} = o(1).$$

The last inequality is from the Chernoff bounds, while the last equality is true for $\alpha = k^{0.4}$.

Proof of Lemma 8. Consider the medium blocking event $\text{MB}(j)$. Let i be a constraint that causes $\text{MB}(j)$ and let $j_1, j_2, \dots, j_h \neq j$ be the other variables in constraint i such that $j_1, j_2, \dots, j_h \in \text{med}(i)$. Denote $X_j, X_{j_1}, X_{j_2}, \dots, X_{j_h}$ to be the indicators that $j \in \mathcal{R}_0, j_1, j_2, \dots, j_h \in \mathcal{R}_0$, respectively.

We know that $a_{ij}x_j + a_{ij_1}x_{j_1} + a_{ij_2}x_{j_2} + \dots + a_{ij_h}x_{j_h} \leq 1$ and, since $j, j_1, j_2, \dots, j_h \in \text{med}(i)$, we have $x_j + x_{j_1} + x_{j_2} + \dots + x_{j_h} \leq \ell$ for some fixed value ℓ . Scenario $\text{MB}(j)$ is “bad” if $X_{j_1} + X_{j_2} + \dots + X_{j_h} \geq 2$. Note that $\mathbb{E}[X_j + X_{j_1} + X_{j_2} + \dots + X_{j_h}] \leq \frac{\alpha\ell}{k}$.

Using the the Chernoff bounds in the form denoted in Theorem 20 of the Appendix, we have,

$$\Pr[X_{j_1} + X_{j_2} + \dots + X_{j_h} \geq 2 \mid X_j = 1] = \Pr[X_{j_1} + X_{j_2} + \dots + X_{j_h} \geq 2] \leq O\left(\frac{\alpha^2 \ell^2}{k^2}\right).$$

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 $\alpha = k^{0.4}$ and $\ell = \Theta(\log(k/\alpha))$, this value is $o(1)$.

For scenario $\text{TB}(j)$, we will do the following: If j is tiny, then a sufficient condition for the “bad” event for constraint i is if one of the following occurs:

- (1) **Blocked by other tiny items.** The sum of all coefficients of items j' in \mathcal{R}_0 where $a_{i,j'} \leq 1/\ell$ is greater than $1 - 1/\ell$.
- (2) **Blocked by one medium item and other tiny items.** There exists an item j' in \mathcal{R}_0 such that $1/\ell < a_{i,j'} \leq 1/2$ and sum of coefficients of all items j'' such that $a_{i,j''} \leq 1/\ell$ is greater than $1/2 - 1/\ell$.
- (3) **Blocked by two medium items.** There exists at least two items j' and j'' in \mathcal{R}_0 such that $1/\ell < a_{i,j'}, a_{i,j''} \leq 1/2$.

Note in case there there exists three or more medium items blocking a tiny item, that is handled in the previous case. Hence, the only possible way a tiny item could be blocked after this is by a big item. We will now show that each of these cases occurs with probability $o(1)$.

Case (1). Consider a constraint i . Let \mathcal{H}_i denote the set of tiny items appearing in this constraint except item j . Note that $\mathbb{E}[\sum_{h \in \mathcal{H}_i} A_{ih}X_{ih}] \leq \alpha/k$. Moreover, we have,

$$\Pr[\sum_{h \in \mathcal{H}_i} A_{ih}X_{ih} > 1 - A_{ij}] \leq \Pr[\sum_{h \in \mathcal{H}_i} A_{ih}X_{ih} > 1 - 1/\ell] = \Pr[\sum_{h \in \mathcal{H}_i} \ell A_{ih}X_{ih} > \ell - 1].$$

Note that for every $h \in \mathcal{H}_i$, we have $\ell A_{ih}X_{ih} \in [0, 1]$. Define $Y_h := \ell A_{ih}X_{ih}$. Thus, we have $\mathbb{E}[\sum_{h \in \mathcal{H}_i} Y_h] \leq \frac{\alpha\ell}{k}$. We want to upper-bound the quantity,

$$\Pr\left[\sum_{h \in \mathcal{H}_i} Y_h > \ell - 1\right].$$

Define $\delta = \frac{\ell-1}{\ell} \cdot \frac{k}{2\alpha}$. Note that $(1 + \delta) \frac{\alpha\ell}{k} \leq \ell - 1$. Thus,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} Y_h > \ell - 1 \right] \leq \Pr \left[\sum_{h \in \mathcal{H}_i} Y_h > (1 + \delta) \frac{\alpha\ell}{k} \right].$$

From Chernoff-Hoeffding bounds (Theorem 19), we have that,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} Y_h > (1 + \delta) \frac{\alpha\ell}{k} \right] \leq \exp \left[-\frac{\delta^2}{2 + \delta} \cdot \frac{\alpha\ell}{k} \right].$$

In what follows, we assume that $k \geq 2$, $\ell \geq 3$, and $\alpha \leq k$. Hence, $1 \geq \frac{\ell-1}{\ell} \geq \frac{2}{3} \geq \frac{1}{2}$. This implies that we have,

$$\frac{\delta^2}{2 + \delta} \cdot \frac{\alpha\ell}{k} \geq \left(\frac{\ell-1}{\ell} \right)^2 \cdot \frac{k^2}{4\alpha^2} \cdot \frac{\alpha\ell}{k} \cdot \frac{1}{2 + \frac{\ell-1}{\ell} \cdot \frac{k}{2\alpha}} \geq \frac{\ell}{40}.$$

Combining the above arguments, we obtain,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} \ell A_{ih} X_{ih} > \ell - 1 \right] \leq \exp \left[-\frac{\ell}{40} \right].$$

Case (2). Consider a constraint i . Let \mathcal{H}_i denote the set of tiny items appearing in this constraint except the item j . Let j' denote the medium item present in this constraint. Note that $\mathbb{E}[\sum_{h \in \mathcal{H}_i} A_{ih} X_{ih}] \leq \alpha/k$. Moreover, we have,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} A_{ih} X_{ih} > 1 - A_{ij} - A_{ij'} \right] \leq \Pr \left[\sum_{h \in \mathcal{H}_i} A_{ih} X_{ih} > 1/2 - 1/\ell \right] = \Pr \left[\sum_{h \in \mathcal{H}_i} \ell A_{ih} X_{ih} > \ell/2 - 1 \right].$$

As in case (1), we invoke the Chernoff-Hoeffding bounds to find an upper-bound. Define $\delta := \frac{\ell/2-1}{\ell} \cdot \frac{k}{2\alpha}$. This implies that $(1 + \delta) \frac{\alpha\ell}{k} \leq \ell/2 - 1$. Therefore, we obtain,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} \ell A_{ih} X_{ih} > \ell/2 - 1 \right] \leq \exp \left[-\frac{\delta^2}{2 + \delta} \cdot \frac{\alpha\ell}{k} \right].$$

Moreover, the fact that $k \geq 2$ and $\ell \geq 3$ implies that $\frac{1}{2} \geq \frac{\ell/2-1}{\ell} \geq \frac{1}{6}$. Thus,

$$\frac{\delta^2}{2 + \delta} \cdot \frac{\alpha\ell}{k} \geq \left(\frac{\ell/2-1}{\ell} \right)^2 \cdot \frac{k^2}{4\alpha^2} \cdot \frac{\alpha\ell}{k} \cdot \frac{1}{2 + \frac{\ell/2-1}{\ell} \cdot \frac{k}{2\alpha}} \geq \frac{\ell}{36}.$$

Combining the above arguments, we obtain that,

$$\Pr \left[\sum_{h \in \mathcal{H}_i} \ell A_{ih} X_{ih} > \ell/2 - 1 \right] \leq \exp \left[-\frac{\ell}{36} \right].$$

Case (3). Consider a constraint i . Let \mathcal{G}_i denote the set of medium items in this constraint. The condition we want is $\sum_{g \in \mathcal{G}_i} X_{ig} \geq 2$. Note that $\mathbb{E}[\sum_{g \in \mathcal{G}_i} X_{ig}] \leq \alpha\ell/k$. Using the Chernoff bounds of the form for the medium case, we have that $\Pr[\sum_{g \in \mathcal{G}_i} X_{ig} \geq 2] \leq O(\frac{\alpha^2 \ell^2}{k^2})$.

Note that taking a union bound over the k constraints, setting $\ell = 80 \log(k/\alpha)$ and $\alpha = k^{0.4}$, we have that the probability of the tiny blocking event (Case (3)) occurring is $O(\frac{\alpha^2 \ell^2}{k^2}) = o(1)$. Likewise, the upper-bounds in Case (1) and Case (2) evaluates to $k^{-0.2}$ and $k^{-1/3}$, respectively. Each of these quantities are $o(1)$.

Near-negative correlation. The natural approach to proving Lemma 6 can introduce substantial positive correlation among the items included in \mathcal{R}_F . However, by slightly modifying the coloring algorithm, we obtain *near-negative* correlation in the upper direction among the items. In particular, we first color using the modified coloring scheme in Algorithm 4. Then, we choose c uniformly at random from the set $[2d + d^{1-\epsilon}]$ and include all items with the color c into the set \mathcal{R}_F .

ALGORITHM 4: 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, ϵ, χ)

```

1 if  $V = \emptyset$  then
2   return  $\chi$ 
3 else
4   Let  $v_{\min}$  denote the vertex with minimum total degree.
5    $\chi = \text{Color-Dir-Graph-Neg-Corr}(G, V \setminus \{v_{\min}\}, d, \epsilon, \chi)$ .
6   Among the smallest  $d^{1-\epsilon}$  colors in the set  $[2d + d^{1-\epsilon}]$  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  $\chi(v_{\min}) = c_r$ .
7   return  $\chi$ 

```

Negative correlation among the events that various items are rounded up is a desirable property, since it reduces the overall variance of the objective function. Recall that in Theorem 1 the ratio holds in expectation. However, the significant positive correlation implies that the variance can be large. Thus, by achieving near negative correlation, we can reduce the variance. In particular, by using $t = 2$ in Theorem 9, we immediately obtain a bound on the variance. In fact, using prior work, it also implies strong concentration bounds (see Remark 1 below).

THEOREM 9. *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 $\{v_1, v_2, \dots, v_t\}$ of items in $[n]$, we have (with $d = \alpha + \sqrt{\alpha \log \alpha}$ and $\alpha = k^{0.4}$ as above)*

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

PROOF. Consider the coloring procedure in Algorithm 4. We will show that this induces near-negative correlation. The other steps in the proof remain the same and will directly imply the theorem. Note that after coloring using Algorithm 4, we obtain the rounded items by first choosing one of the colors c in the set $[2d + d^{1-\epsilon}]$ uniformly at random and then add all the vertices that received a color c into the set \mathcal{R}_F . Since $\alpha^{1-\epsilon} \leq o(\alpha)$, a similar analysis as before follows to give part (1) of Theorem 9. We will now prove part (2). Fix an arbitrary $t \in [n]$ and any t -sized subset $U := \{v_1, v_2, \dots, v_t\}$ 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

$$\prod_{j=1}^t \frac{x_{v_j} \alpha}{k}; \tag{8}$$

suppose that this indeed happens (all the remaining probability calculations are conditional on this). Note that Algorithm 4 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, $\{v_1, v_2, \dots, v_t\}$. Recall again that our probability calculations now are conditional on all items in U being present in G' ; denote this event by $U_{G'}$ and $\mathbb{I}[U_{G'}]$ to be the corresponding indicator. Note that $\Pr[v_1 \in \mathcal{R}_F \mid \mathbb{I}[U_{G'}]] = \frac{1}{2d+d^{1-\epsilon}} \leq \frac{1}{2\alpha}$. This is because we choose one of the colors in $[2d + d^{1-\epsilon}]$ colors at random for the first vertex. Next, a moment's reflection shows that for any j with $2 \leq j \leq t$,

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

Note that,

$$\begin{aligned} & \Pr[v_1 \in \mathcal{R}_F \wedge v_2 \in \mathcal{R}_F \wedge \dots \wedge v_t \in \mathcal{R}_F \mid \mathbb{I}[U_{G'}]] \\ &= \Pr[v_t \in \mathcal{R}_F \mid v_1 \in \mathcal{R}_F \wedge v_2 \in \mathcal{R}_F \wedge \dots \wedge v_{t-1} \in \mathcal{R}_F, \mathbb{I}[U_{G'}]] \times \\ & \Pr[v_{t-1} \in \mathcal{R}_F \mid v_1 \in \mathcal{R}_F \wedge v_2 \in \mathcal{R}_F \wedge \dots \wedge v_{t-2} \in \mathcal{R}_F, \mathbb{I}[U_{G'}]] \times \dots \times \Pr[v_1 \in \mathcal{R}_F \mid \mathbb{I}[U_{G'}]]. \end{aligned}$$

Chaining these together, we obtain,

$$\Pr[v_1 \in \mathcal{R}_F \wedge v_2 \in \mathcal{R}_F \wedge \dots \wedge v_t \in \mathcal{R}_F \mid \mathbb{I}[U_{G'}]] \leq \left(\frac{d^\epsilon}{\alpha}\right)^{t-1} \frac{1}{2\alpha}.$$

Combining this with Equation (8), we get,

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

Remark 1. Let X_j denote the indicator random variable for the event that $j \in \mathcal{R}_F$. Then, by the results of References [63, 72], Theorem 9 yields upper-tail bounds for any non-negative linear combination of the X_j 's.

Implementation in the distributed setting. We briefly give a high-level description on how to obtain distributed algorithms with the same approximation ratio for k -CS-PIP. The algorithm described above cannot directly be implemented in the distributed model of computation. We now briefly describe how our algorithm can be modified to overcome this. Note that step (c) in the current algorithm makes the algorithm inherently *sequential*. The running time of our algorithm is determined by this step and runs in time $O(\text{poly } n)$. However, we can obtain the coloring by using known distributed algorithms that obtain the same ratio and runs in time $O(k^{0.4} \text{poly log } n \text{poly log } k)$. In particular, we prove the following theorem:

THEOREM 10 (IMPLEMENTATION IN THE DISTRIBUTED SETTING). *There exists a rounding algorithm for the k -CS-PIP problem in the distributed setting that achieves an approximation ratio of $2k + o(k)$ in time $\tilde{O}(k^{0.4} \log n)$, where $\tilde{O}(\cdot)$ hides $\text{poly log } k$ factors.*

Before we prove the theorem, we recall the following definitions from graph theory:

Definition 1 (pseudo-Forest). A graph is called a *pseudo-forest* if every connected component has at most one cycle.

Definition 2 (Arboricity). *Arboricity* of a graph is the minimum number of forests that the edge set of the graph can be partitioned into.

Definition 3 (Pseudo-Arboricity). *Pseudo-Arboricity* of a graph is the minimum number of pseudo-forests that the edge set of the graph can be partitioned into.

Definition 4 (Minimum out-degree orientation). Given an undirected graph $G = (V, E)$ an orientation maps the set of undirected edges to a set of directed edges between the same set of vertices.

The out-degree of a orientation is defined as the largest out-degree of any vertex under a given orientation. The *minimum out-degree orientation* is defined as the orientation in which the out-degree is minimum across all possible orientations.

We now prove Theorem 10.

PROOF. Note that both the sampling step and the discarding low-probability events step in Algorithm 2 can be implemented in parallel in $O(1)$ time. Thus, the bottleneck step in the algorithm is the coloring step, which is inherently sequential. Using results from prior work, we show that this can be parallelized as follows: Let A denote the arboricity of the constructed directed graph in step (b) of Algorithm 2. Using the Arb-color algorithm from [16]¹⁵ with $\epsilon = \frac{1}{d+1}$, we obtain a $2A + 1 + \frac{A}{d+1}$ coloring to this directed graph in time $O(A \log n)$. It is known that (see Reference [61], for instance) the arboricity of a graph A is at most one more than the pseudo-arboricity of that graph (denoted by PA). Thus, the number of colors used is $2PA + 2 + \frac{PA+1}{d+1}$. Note that the max out-degree in our constructed directed graph is at most d ; thus, the max-degree in the minimum out-degree orientation is at most d . From theorems in References [43, 67], it follows that the pseudo-arboricity of a graph is the out-degree in the minimum out-degree orientation. Therefore, the total number of colors used is at most $2d + 3$. Note that $A \leq PA + 1 \leq d + 1$. Thus, the running time of the algorithm is $O(d \log n) = \tilde{O}(k^{0.4} \log n)$. Given a coloring, we can choose an independent set (and thus the rounded items) in $O(1)$ time. Therefore, the entire rounding procedure runs in time $\tilde{O}(k^{0.4} \log n)$. \square

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. [36]¹⁶:

Definition 5 (bc-BALANCED MONOTONE CR SCHEMES [36]). Let $b, c \in [0, 1]$ and let $N := [n]$ be a set of items. A bc -balanced CR scheme π for \mathcal{P}_I (where \mathcal{P}_I denotes the convex relaxation of the set of feasible integral solutions $\mathcal{I} \subseteq 2^N$) is a procedure that for every $y \in b\mathcal{P}_I$ and $A \subseteq N$, returns a set $\pi_y(A) \subseteq A \cap \text{support}(y)$ with the following properties:

- (a) For all $A \subseteq N$, $y \in b\mathcal{P}_I$, we have that $\pi_y(A) \in \mathcal{I}$ with probability 1.
- (b) For all $i \in \text{support}(y)$, $y \in b\mathcal{P}_I$, we have that $\Pr[i \in \pi_y(R(y)) \mid i \in R(y)] \geq c$, where $R(y)$ 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 $\Pr[i \in \pi_y(A_1)] \geq \Pr[i \in \pi_y(A_2)]$.

THEOREM 11 (CHEKURI ET AL. [36]). Suppose \mathcal{P}_I admits a bc -balanced monotone CR scheme and η_f is the approximation ratio for $\max\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_I \cap \{0, 1\}^n\}$ (here, $F(\mathbf{x})$ is the multi-linear extension of the sub-modular function f). Then there exists a randomized algorithm that 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 [77]). For non-monotone sub-modular functions, the best-known algorithms have $\eta_f \geq 0.372$ due to Ene and Nguyen [40] and more recently $\eta_f \geq 0.385$ due to Buchbinder and Feldman [23]. (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 [62]).

¹⁵Note that this algorithm can be implemented without knowing A .

¹⁶Specifically, Definition 1.2 and Theorem 1.5 from Reference [36].

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 Theorem 11, we have a $(2k + o(k))/\eta_f$ -approximation algorithm for k -CS-PIP with sub-modular objectives. This yields Corollary 1.

COROLLARY 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 $\max\{F(\mathbf{x}) : \mathbf{x} \in \mathcal{P}_I \cap \{0, 1\}^n\}$ (here, $F(\mathbf{x})$ is the multi-linear extension of the sub-modular function f and \mathcal{P}_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.¹⁷*

For ease of reading, we will first restate notations used in Definition 5 in the form stated in the previous sub-section. The polytope \mathcal{P}_I represents the k -CS-PIP polytope defined by Equation (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., Step 1). The scheme π_y is the procedure defined by steps 1, 2, 3 of the algorithm. In other words, this procedure takes any 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 $\mathbf{x} \in \mathcal{P}_I$, and hence $\mathbf{y} \in (\alpha/k)\mathcal{P}_I$. 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, π_y satisfies property (b).

Now, we will show that the rounding scheme π_y satisfies property (c) in Definition 5. 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:

$$\Pr[j \in \mathcal{R}_F \mid \mathcal{R}_0 = A_1] \geq \Pr[j \in \mathcal{R}_F \mid \mathcal{R}_0 = A_2]. \quad (9)$$

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

$$\Pr[j \in \mathcal{R}_F \mid \mathcal{R}_0 = A_i] = \Pr[j \in \mathcal{R}_F \mid j \in \mathcal{R}_2, \mathcal{R}_0 = A_i] \Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_i]. \quad (10)$$

For both $i = 1$ and $i = 2$, the first term in the RHS of Equation (10) (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 Equation (10) (i.e., $\Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_i]$) can be rewritten as $\Pr[j \in \mathcal{R}_2 \mid j \in \mathcal{R}_0, \mathcal{R}_0 = A_i]$, since $j \in A_1$ and $\mathcal{R}_0 = A_i$ for $i \in \{1, 2\}$. From steps 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 $d := \alpha + \sqrt{\alpha \log \alpha}$).

Both (i) and (ii) are monotonically decreasing in the set \mathcal{R}_0 . In other words, if both the conditions satisfy for an item j when $\mathcal{R}_0 = A_2$, then it also holds when $\mathcal{R}_0 = A_1$. This implies that $\Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_1] \geq \Pr[j \in \mathcal{R}_2 \mid \mathcal{R}_0 = A_2]$. Thus, combining this with Equation (10), we obtain Equation (9).

3 THE STOCHASTIC K -SET PACKING PROBLEM

Consider the stochastic k -set packing problem defined in the introduction. Recall that in this problem, the columns of the packing constraints (i.e., size of items) are realized by a stochastic process

¹⁷To 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 article, where the values are always greater than 1.

on probing, and the goal is to choose the optimal order in which the columns (i.e., items) have to probe. 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 (4) and $C(j)$ be the set of constraints that j participates in.

Bansal et al. [13] presented Algorithm 5, 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 $2k$.¹⁸

ALGORITHM 5: SKSP(α) [13]

- 1 Let \mathcal{R} denote the set of chosen items that 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 $i \in C(j)$ 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 Reference [13], 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.¹⁹ 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 6, SKSP($\alpha_1, \beta_1, \alpha_2$), gives a formal description.

ALGORITHM 6: 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.
-

¹⁸The terminology used in Reference [13] is actually $1/\alpha$; however, we find the inverted notation α more natural for our calculations.

¹⁹See footnote 10 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.

Lemma 12 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 12. *After running $SKSP(\alpha_1, \beta_1, \alpha_2)$ on an optimal solution \mathbf{x} to the benchmark LP (4), we have*

$$\Pr[E_{2,j}] \geq \frac{x_j}{k} \alpha_2 \left(1 - \frac{\alpha_1 x_j}{k} - \beta_1 - \frac{\alpha_2}{2} \right).$$

PROOF. 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 $i \in C(j)$ 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.

$$\Pr[E_{2,j}] = \Pr[Y_{1,j} = 0 \wedge Y_{2,j} = 1] \Pr[E_{2,j} \mid Y_{1,j} = 0 \wedge Y_{2,j} = 1] \quad (11)$$

$$= \left(1 - \frac{\alpha_1 x_j}{k} \right) \frac{\alpha_2 x_j}{k} \Pr \left[\bigwedge_i (U_{2,i} \leq b_i - 1) \mid Y_{1,j} = 0 \wedge Y_{2,j} = 1 \right] \quad (12)$$

$$\geq \left(1 - \frac{\alpha_1 x_j}{k} \right) \frac{\alpha_2 x_j}{k} \left(1 - \sum_i \Pr[U_{2,i} \geq b_i \mid Y_{1,j} = 0] \right) \quad (13)$$

$$\geq \frac{\alpha_2 x_j}{k} \left(1 - \frac{\alpha_1 x_j}{k} - \sum_i \Pr[U_{2,i} \geq b_i] \right). \quad (14)$$

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,

$$U_{2,i} \leq \sum_{\ell \neq j} \text{Sl}_{i,\ell} (X_{1,\ell} + (1 - Y_{1,\ell}) Y_{2,\ell} 1_{\{2,\ell\}}),$$

which implies

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

Plugging the above inequality into (14) and applying Markov's inequality, we complete the proof of Lemma 12. \square

We can use Lemma 12 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 $(5/8k - o(k)) \mathbf{w} \cdot \mathbf{x}$.

THEOREM 13. *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.*

PROOF. Consider a given item j . We have,

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

To obtain the worst case, we solve the following optimization problem:

$$\max \beta_1 + \alpha_2 \left(1 - \beta_1 - \frac{\alpha_2}{2}\right), \text{ s.t. } 0 \leq \beta_1 \leq \alpha_1(1 - \alpha_1/2), \alpha_1 \geq 0, \alpha_2 \geq 0. \quad (15)$$

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

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

During each chance $t \leq T$, 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 $\beta_t x_j/k$. To achieve this goal, we need to simulate our algorithm over all previous chances up to the current one t . Algorithm 7, $\text{SKSP}(\{\alpha_t, \beta_t \mid t \in [T]\})$, 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 uniformity in presentation, we still describe it in the algorithm description.

ALGORITHM 7: $\text{SKSP}(\{\alpha_t, \beta_t \mid t \in [T]\})$

```

1 Initialize  $\mathcal{R}$  as the empty set.
2 for  $t=1, 2, \dots, T$  do
3   Generate an independent Bernoulli random variable  $Y_{t,j}$  with mean  $\alpha_t x_j/k$  for each  $j$ .
4   Apply simulation-based attenuation such that for each  $j$  is added to  $\mathcal{R}$  in the  $t$ th chance (denote by
   an indicator random variable  $Z_{t,j}$ ) with probability equal to  $\beta_t x_j/k$ .
5   Follow a uniform random order  $\pi_t$  over  $[n]$  to check each item  $j$  one-by-one: add  $j$  to  $\mathcal{R}$  if and only
   if  $j$  is safe,  $Z_{t',j} = 0$  for all  $t' < t$ , and  $Z_{t,j} = 1$ ; otherwise, skip it.
6 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 step 3 of the algorithm for t (i.e., before the start of the $(t+1)$ -th iteration of the loop). Lemma 14 lower bounds the probabilities of these events.

LEMMA 14. *After running $\text{SKSP}(\{\alpha_t, \beta_t \mid t \in [T]\})$ on an optimal solution \mathbf{x} to the benchmark LP (4), we have*

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

PROOF. Consider an item j and define indicator random variable $Z_{t,j} = 1$ iff $Y_{t',j} = 0$ for all $t' < t$ and $Y_{t,j} = 1$. Observe that $\mathbb{E}[Z_{t,j}] = \frac{\alpha_t x_j}{k} \prod_{t' < t} (1 - \frac{\alpha_{t'} x_j}{k})$. Consider a given $i \in C(j)$ 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,

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

Notice that

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

By applying Markov's inequality, we get

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

Combining Lemma 14 and simulation-based attenuation, we have that for any given $\{\alpha_{t'} \mid t' \leq t\}$ 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 $\beta_t \leq \alpha_t (1 - \sum_{t' < t} \beta_{t'} - \alpha_t/2) - \alpha_t \sum_{t' < t} \alpha_{t'}/k$. 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 7, we have that the events $\{E_{j,t} \mid t \leq T\}$ are mutually exclusive. Thus, $\Pr[E_j] = \sum_{t \leq T} \Pr[E_{j,t}] = \sum_{t \leq T} \beta_t x_j/k$. Therefore, to maximize the final ratio, we will solve the following optimization problem:

$$\max \sum_{t \in [T]} \beta_t \text{ s.t. } \beta_t \leq \alpha_t \left(1 - \sum_{t' < t} \beta_{t'} - \frac{\alpha_t}{2}\right) - \frac{\alpha_t \sum_{t' < t} \alpha_{t'}}{k} \quad \forall t \in [T], \alpha_t \geq 0 \quad \forall t \in [T]. \quad (16)$$

Consider a simplified version of the maximization program (16) by ignoring the $O(1/k)$ term as follows:

$$\max \sum_{t \in [T]} \beta_t, \text{ s.t. } \beta_t \leq \alpha_t \left(1 - \sum_{t' < t} \beta_{t'} - \frac{\alpha_t}{2}\right) \quad \forall t \in [T], \alpha_t \geq 0 \quad \forall t \in [T]. \quad (17)$$

LEMMA 15. An optimal solution to the program (17) is

$$\beta_t^* = \frac{1}{2} \left(1 - \sum_{t' < t} \beta_{t'}^*\right)^2, \quad \forall t \geq 1, \alpha_t^* = 1 - \sum_{t' < t} \beta_{t'}^*, \quad \forall t \geq 1,$$

where $\beta_0^* = 0$, $0 \leq \alpha_t^* \leq 1$ for all $t \geq 1$ and $\lim_{T \rightarrow \infty} \sum_{t=1}^T \beta_t^* = 1$.

PROOF. For any given $\{\beta_t \mid 1 \leq t < T\}$, we have $\beta_T \leq \alpha_T (1 - \sum_{t' < T} \beta_{t'} - \alpha_T/2)$. Thus, in the optimal solution, we have that $\beta_T^* = \frac{1}{2}(1 - \sum_{1 \leq t' < T} \beta_{t'}^*)^2$ and $\alpha_T = 1 - \sum_{1 \leq t' < T} \beta_{t'}^*$. Thus,

$$\sum_{t \in [T]} \beta_t = \sum_{t < T} \beta_t + \beta_T^* = \frac{1}{2} \left(1 + \left(\sum_{t < T} \beta_t \right)^2 \right).$$

Therefore, the maximization of $\sum_{t \leq T} \beta_t$ is reduced to that of $\sum_{t \leq T-1} \beta_t$. Repeating the above analysis 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 have

$$\gamma_1^* = \frac{1}{2}, \gamma_t^* = \frac{1}{2} \left(1 + (\gamma_{t-1}^*)^2 \right), \forall t \geq 2.$$

Since $\gamma_1^* \leq 1$, we can prove that $\gamma_t^* \leq 1$ for all t by induction. Notice that $\gamma_t^* - \gamma_{t-1}^* = \frac{1}{2}(1 - \gamma_{t-1}^*)^2 \geq 0$, which implies that $\{\gamma_t^*\}$ is 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 $\lim_{T \rightarrow \infty} \gamma_T^* = 1$. \square

THEOREM 16. *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 for the program (16) 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($\{\bar{\alpha}_t, \bar{\beta}_t \mid t \in [T]\}$) achieves a ratio of $k + o(k)$ for SKSP.*

PROOF. Observe that for each $t \in [T]$, $\bar{\beta}_t \leq \beta_t^*$. Also,

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

Thus, we claim that $\{\bar{\alpha}_t, \bar{\beta}_t \mid t \in [T]\}$ is feasible for the program (16). Notice that

$$\sum_{t \in [T]} \bar{\beta}_t = \sum_{t \in [T]} \beta_t^* - \sum_{t \in [T]} \frac{\alpha_t^* (\sum_{t' < t} \alpha_{t'}^*)}{k} \geq \gamma_T^* - \frac{T^2}{k} = 1 - \left(1 - \gamma_T^* + \frac{\log^2 k}{k} \right).$$

From Lemma 15, we have that $(1 - \gamma_T^*) = o(1)$, thus proving the theorem. \square

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 (5). We start with a warm-up algorithm due to Bansal et al. [13].²⁰ Note that Reference [13] studies the more general SKSP problem. The algorithm they describe is the one described in Algorithm 5 in the previous section. Here, we show that when restricted to the special setting of hypergraph matching, the same algorithm yields an approximation ratio of $k + 1 + o(k)$. We use this as a starting point to obtain an algorithm with improved competitive ratio. We recall their approach, denoted by HM(α), in Algorithm 8.

²⁰Similar to our algorithm for SKSP, we use the notation α , while Reference [13] uses $1/\alpha$.

ALGORITHM 8: $\text{HM}(\alpha)$

-
- 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 For every edge $e \in \mathcal{E}$ with $Y_e = 1$, choose a random number $x_e \in [0, 1]$ uniformly at random and independent of the other edges. Let π denote the ordering over \mathcal{E} such that the realized values $(x_e)_{e \in \mathcal{E}}$ are sorted in ascending order. 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.
-

LEMMA 17. *Each edge e is added to \mathcal{R} with probability at least $\frac{x_e}{k_e + 1 + o(k_e)}$ in $\text{HM}(\alpha)$ with $\alpha = 1$.*

PROOF. The proof is very similar to that in Reference [17] (for instance, see Lemma 10 in Reference [17]). For each $e \in \mathcal{E}$, let E_e be the event that e is added to \mathcal{R} in $\text{HM}(\alpha)$. Consider an edge e and for each vertex $v \in e$, let L_v be the event that v is unmatched when considering e in the permutation π . Let $\pi(e) = t \in (0, 1)$. Thus, E_e occurs if $Y_e = 1$ and every vertex $v \in e$ is unmatched when considering e in the permutation. Therefore, we have the following:

$$\begin{aligned} \Pr[E_e] &= \alpha x_e \int_0^1 \Pr \left[\bigwedge_{v \in e} L_v \mid \pi(e) = t \right] dt \\ &= \alpha x_e \int_0^1 \prod_{v \in e} \prod_{f: f \ni v} (1 - t \alpha x_f) dt \end{aligned} \quad (18)$$

$$\geq \alpha x_e \int_0^1 (1 - t \alpha)^{\sum_{v \in e} \sum_{f: f \ni v} x_f} dt \quad (19)$$

$$\begin{aligned} &\geq \alpha x_e \int_0^1 (1 - t \alpha)^{k_e} dt \\ &= \frac{x_e}{k_e + 1} (1 - (1 - \alpha)^{k_e + 1}). \end{aligned} \quad (20)$$

Equation (18) can be obtained as follows: For a given vertex $v \in e$ to be safe, we want that none of the edges incident to v be matched when considering e . This is precisely the negation of the probability that an edge incident to v is matched earlier in the permutation. A similar argument is also made in Lemma 10 of Reference [17]. Equation (19) is obtained as follows: Note that $x_f \leq 1$ and $\alpha = 1$. This implies that $t \alpha \leq 1$ when $t \in [0, 1]$. From Lemma 22 in the appendix, we have that $(1 - t \alpha x_f) \geq (1 - t \alpha)^{x_f}$. Equation (20) is obtained as follows: From the LP constraints, we have that $\sum_{f: f \ni v} x_f \leq 1$. Moreover, the number of vertices $v \in e$ is precisely k_e . Combining this with the fact that $1 - t \alpha \leq 1$, we obtain the inequality. This completes the proof of the lemma. \square

It can be shown that in $\text{HM}(\alpha = 1)$, 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 8 as follows: We generate an independent Bernoulli random variable Y_e with mean $g(x_e)$ for each e , where $g : [0, 1] \rightarrow [0, 1]$ is once differentiable and satisfying $g(0) = 0$ and $g'(0) = 1$. Function g will be fixed in the analysis. Algorithm 9 gives a formal description of this modified algorithm.

Observe that $\text{HM}(\alpha)$ 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 $\text{HM}(g)$ is maximized. Consider a given e

ALGORITHM 9: 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 For every edge $e \in \mathcal{E}$ with $Y_e = 1$, choose a random number $x_e \in [0, 1]$ uniformly at random and independent of the other edges. Let π denote the ordering over \mathcal{E} such that the realized values $(x_e)_{e \in \mathcal{E}}$ is sorted in ascending order. 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
-

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 (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 17, we get the probability of E_e is at least

$$\Pr[E_e] \geq g(x) \int_0^1 \prod_{e' \sim e} (1 - tg(x_{e'})) dt. \quad (21)$$

Therefore, our task of finding an optimal g to maximize the r.h.s. of (21) is equivalent to finding $\max_g \mathcal{F}(g)$, where $\mathcal{F}(g)$ is defined in Equation (22).

$$\mathcal{F}(g) \doteq \min_{x \in (0, 1)} \left[\frac{g(x)}{x} \times r(x) \right]. \quad (22)$$

In Equation (22), $r(x)$ is defined as

$$r(x) \doteq \min \int_0^1 \prod_{e' \sim e} (1 - tg(x_{e'})) dt, \text{ where } \sum_{e' \sim e} x_{e'} \leq k_e(1 - x), x_{e'} \in [0, 1], \forall e'.$$

LEMMA 18. *By choosing $g(x) = x(1 - \frac{x}{2})$, we have that the minimum value of $\mathcal{F}(g)$ in Equation (22) is $\mathcal{F}(g) = \frac{1}{k_e}(1 - \exp(-k_e))$.*

PROOF. Note that $g(0) = 0$. Additionally, $g'(x) = 1 - x$ and thus $g'(0) = 1$.

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

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

Note that $g(x) = x(1 - x/2)$ satisfies the condition of Lemma 21 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 $\sum_{e' \sim e} \ln(1 - tg(x_{e'}))$ subject to $0 \leq x_{e'} \leq 1$ and $\sum_{e'} x_{e'} \leq \kappa$ 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 $\epsilon \rightarrow 0$. Thus,

$$\min \prod_{e' \sim e} (1 - tg(x_{e'})) = \min \exp \left(\sum_{e' \sim e} \ln(1 - tg(x_{e'})) \right) = \lim_{\epsilon \rightarrow 0} (1 - tg(\epsilon))^{\kappa/\epsilon} = \exp(-t\kappa).$$

The last inequality is obtained as follows: Let $y := \lim_{\epsilon \rightarrow 0} (1 - tg(\epsilon))^{\kappa/\epsilon}$. Taking \ln on both sides, we obtain $\ln y = \lim_{\epsilon \rightarrow 0} \frac{\kappa}{\epsilon} \ln(1 - tg(\epsilon))$. Using the L'Hopital's rule, we obtain $\ln y = -\kappa t (\lim_{\epsilon \rightarrow 0} \frac{g'(\epsilon)}{1 - tg(\epsilon)})$. Since $g'(0) = 1$ and $g(0) = 0$, the limit thus evaluates to $-\kappa t$. Taking exponentials on both sides, we obtain $y = \exp(-t\kappa)$. Therefore, for each fixed $x_e = x$, the optimal value to the inner minimization program in Equation (22) has the following analytic form:

$$\min \int_0^1 \prod_{e' \sim e} (1 - tg(x_{e'})) dt = \int_0^1 \exp(-t\kappa) dt = \frac{1 - \exp(-\kappa)}{\kappa}. \quad (23)$$

Plugging this back into Equation (22), we obtain $\mathcal{F}(g) = \min_{x \in [0,1]} G(x)$, where

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

Note that $G'(x) \geq 0$ whenever $x \in [0, 1]$ and thus, the minimum value of $G(x)$ in the range $x \in (0, 1)$ is $G(0) = \frac{1}{k_e} (1 - \exp(-k_e))$. Moreover, for the class of functions g with $g(0) = 0$ and $g'(0) = 1$, this analysis is tight. \square

We now prove the main result, Theorem 3.

PROOF. Consider $\text{HM}(g)$ as shown in Algorithm 9 with $g(x) = x(1 - x/2)$. Let \mathcal{R} be the random matching returned. From Lemma 18, we have that each e will be added to \mathcal{R} with probability at least $x_e \mathcal{F}(g) = \frac{x_e}{k_e} (1 - \exp(-k_e))$. \square

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. [34] where they gave a 4-approximation for the linear objective case. They also described a simple randomized algorithm that obtains a 27-approximation.²¹ Later, Chekuri et al. [36] 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 functions²²). 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 scheme²³ to a $1/8.15$ -balanced CR scheme and, hence, from Theorem 1.5 of Reference [36], 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 10) proceeds similar to the one described in Reference [34], except at line 3, where we use our attenuation ideas.

ALGORITHM 10: Improved contention resolution Scheme UFP-TREES with unit demands

- 1 Root the tree T arbitrarily. Let $\text{LCA}(s_i, t_i)$ 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.
-

Analysis. For the most part, the analysis is similar to the exposition in Chekuri et al. [36]. We will highlight the part where *attenuation* immediately leads to improved bounds.

²¹This can be obtained by maximizing over all $0 < b < 1/3e$ in Lemma 4.19 of Reference [36], which yields approximately $1/27$.

²²See the section on k -CS-PIP for a discussion on various values of η_f known.

²³See the section on extension to sub-modular objectives in k -CS-PIP for definition of a balanced CR scheme.

Consider a fixed pair (s_{i^*}, t_{i^*}) and let $\ell := \text{LCA}(s_{i^*}, t_{i^*})$ 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 Reference [36], we will upper bound the probability of i^* being unsafe due to path P and a symmetric argument holds for P' . Let $e_1, e_2, \dots, e_\lambda$ 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 10, 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 that 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 Reference [36] and consider a strictly decreasing sequence of capacities $u_{e_1} > u_{e_2} > \dots > u_{e_\lambda} \geq 1$. Let S_j denote the set of demand pairs that use edge e_j . The following steps are the part where our analysis differs from Reference [36] 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 Reference [36].

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

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 $X := \sum_{i \in S_j} X_i$.

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 [63]. 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 $\Pr[X \geq u_{e_j}] \leq (e/(1 + \delta))^{(1 + \delta)u_{e_j}} \leq (\gamma e)^{u_{e_j}}$. 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 $\sum_{q=1}^{\infty} (\gamma e)^q = (\gamma e)/(1 - \gamma e)$. (We used the fact that $u_{e_1} > u_{e_2} > \dots > u_{e_\lambda} \geq 1$.) 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. [44] (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. [36] 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 Reference [44] 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.

APPENDIX

A TECHNICAL LEMMAS

In the main section, we use the following two variants of the Chernoff-Hoeffding bounds in the analysis of k -CS-PIP algorithm: Theorem 19 is the standard multiplicative form, while Theorem 20 can be derived from the standard form.

THEOREM 19 (MULTIPLICATIVE FORM OF CHERNOFF-HOEFFDING BOUNDS). *Let $X_1, X_2, \dots, X_n \in [0, 1]$ be independent random variables such that $\mathbb{E}[\sum_{i \in [n]} X_i] \leq \mu$. Then for every $\delta > 0$, we have,*

$$\Pr \left[\sum_{i \in [n]} X_i \geq (1 + \delta)\mu \right] \leq \left(\frac{e^\delta}{(1 + \delta)^{(1 + \delta)}} \right)^\mu \leq \exp \left[-\frac{\delta^2 \mu}{2 + \delta} \right].$$

THEOREM 20 (CHERNOFF-HOEFFDING BOUNDS). *Suppose c_1, c_2 , and k are positive values with $c_2 \geq \frac{c_1}{k}$. Let $X_1, X_2, \dots, X_n \in [0, 1]$ be independent random variables satisfying $\mathbb{E}[\sum_i X_i] \leq \frac{c_1}{k}$. Then,*

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

PROOF. The standard form of the Chernoff-Hoeffding bounds (Theorem 19) 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 $1 + \delta = \frac{k c_2}{c_1}$. Plugging this into the standard form of the Chernoff-Hoeffding bounds gives us the desired bound. \square

LEMMA 21 (CONVEXITY). *Assume $f : [0, 1] \rightarrow [0, 1]$ 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 $x \in [0, 1]$. \square

LEMMA 22. *Let $0 \leq a \leq 1$ be a given real number. Then for every $x \in [0, 1]$, we have that*

$$(1 - a)^x \leq 1 - ax.$$

PROOF. Consider the function $g(x) := 1 - ax - (1 - a)^x$. We will show that when $x \in [0, 1]$, we have $g(x) \geq 0$. This will complete the proof. Note that $g(0) = 0$. We will now show that $g(x)$ is increasing in $x \in [0, 1]$. We have that $g'(x) = (1 - a) - (1 - a)^x \ln(1 - a) = (1 - a) + (1 - a)^x \ln(\frac{1}{1 - a}) \geq 0$. \square

B SUBMODULAR FUNCTIONS

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

Definition 6 (Submodular functions). A function $f : 2^{[n]} \rightarrow \mathbb{R}_+^+$ on a ground-set of elements $[n] := \{1, 2, \dots, n\}$ 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 that on querying a subset $T \subseteq [n]$, returns the value $f(T)$.

Definition 7 (Multi-linear extension). The multi-linear extension of a submodular function f is the continuous function $F : [0, 1]^n \rightarrow \mathbb{R}_+^+$ defined as

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

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 $f(S)$.

ACKNOWLEDGMENTS

The authors would like to thank David Harris, the anonymous reviewers of SODA 2018, and the anonymous reviewers of *TALG* for useful suggestions that led to this improved version of the article. In particular, the authors would like to thank David Harris for pointing out the related works for the distributed implementation of graph coloring and showing the equivalence between the Conjecture 1 and Conjecture 2. The authors are also grateful to Marek Adamczyk who communicated his new (independent) result for the SKSP problem to us.

REFERENCES

- [1] Marek Adamczyk. 2015. Non-negative submodular stochastic probing via stochastic contention resolution schemes. *CoRR* abs/1508.07771 (2015). Retrieved from: <http://arxiv.org/abs/1508.07771>.
- [2] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. 2015. Improved approximation algorithms for stochastic matching. In *Proceedings of the European Symposium on Algorithms (ESA'15)*. Springer, 1–12.
- [3] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. 2014. Submodular stochastic probing on matroids. In *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science*. Leibniz Int. Proc. Inform., Vol. 25. 29–40.
- [4] Marek Adamczyk and Michał Włodarczyk. 2018. Random order contention resolution schemes. Retrieved from: *arXiv preprint arXiv:1804.02584* (2018).
- [5] Miklós Ajtai, János Komlós, and Endre Szemerédi. 1980. A note on Ramsey numbers. *J. Comb. Theory, Ser. A* 29, 3 (1980), 354–360.
- [6] Esther M. Arkin and Refael Hassin. 1998. On local search for weighted k -set packing. *Math. Oper. Res.* 23, 3 (1998), 640–648. DOI : <https://doi.org/10.1287/moor.23.3.640>
- [7] Yossi Azar and Oded Regev. 2006. Combinatorial algorithms for the unsplittable flow problem. *Algorithmica* 44, 1 (2006), 49–66. DOI : <https://doi.org/10.1007/s00453-005-1172-z>
- [8] Nikhil Bansal. 2010. Constructive algorithms for discrepancy minimization. In *Proceedings of the IEEE 51st Symposium on Foundations of Computer Science (FOCS'10)*. IEEE Computer Society, Los Alamitos, CA, 3–10.
- [9] Nikhil Bansal, Amit Chakrabarti, Amir Epstein, and Baruch Schieber. 2006. A quasi-PTAS for unsplittable flow on line graphs. In *Proceedings of the 38th Symposium on the Theory of Computing (STOC'06)*. ACM, New York, 721–729. DOI : <https://doi.org/10.1145/1132516.1132617>
- [10] Nikhil Bansal, Daniel Dadush, and Shashwat Garg. 2016. An algorithm for Komlós conjecture matching Banaszczyk's bound. In *Proceedings of the IEEE 57th Symposium on Foundations of Computer Science (FOCS'16)*. IEEE, 788–799.
- [11] Nikhil Bansal, Zachary Friggstad, Rohit Khandekar, and Mohammad R. Salavatipour. 2014. A logarithmic approximation for unsplittable flow on line graphs. *ACM Trans. Alg.* 10, 1 (2014), Article 1, 15. DOI : <https://doi.org/10.1145/2532645>
- [12] Nikhil Bansal and Shashwat Garg. 2016. Improved algorithmic bounds for discrepancy of sparse set systems. Retrieved from: *arXiv preprint arXiv:1601.03311* (2016).

- [13] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. 2012. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica* 63, 4 (2012), 733–762. DOI : <https://doi.org/10.1007/s00453-011-9511-8>
- [14] Nikhil Bansal, Nitish Korula, Viswanath Nagarajan, and Aravind Srinivasan. 2012. Solving packing integer programs via randomized rounding with alterations. *Theor. Comput.* 8 (2012), 533–565. DOI : <https://doi.org/10.4086/toc.2012.v008a024>
- [15] Nikhil Bansal and Joel Spencer. 2013. Deterministic discrepancy minimization. *Algorithmica* 67, 4 (2013), 451–471. DOI : <https://doi.org/10.1007/s00453-012-9728-1>
- [16] Leonid Barenboim and Michael Elkin. 2010. Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. *Distrib. Comput.* 22, 5–6 (2010), 363–379.
- [17] Alok Baveja, Amit Chavan, Andrei Nikiforov, Aravind Srinivasan, and Pan Xu. 2015. Improved bounds in stochastic matching and optimization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Leibniz Int. Proc. Inform., Vol. 40. 124–134.
- [18] Alok Baveja and Aravind Srinivasan. 2000. Approximation algorithms for disjoint paths and related routing and packing problems. *Math. Oper. Res.* 25, 2 (2000), 255–280. DOI : <https://doi.org/10.1287/moor.25.2.255.12228>
- [19] József Beck and Tibor Fiala. 1981. “Integer-making” theorems. *Disc. Appl. Math.* 3, 1 (1981), 1–8. DOI : [https://doi.org/10.1016/0166-218X\(81\)90022-6](https://doi.org/10.1016/0166-218X(81)90022-6)
- [20] Piotr Berman. 2000. A $d/2$ approximation for maximum weight independent set in d -claw free graphs. *Nordic J. Comput.* 7, 3 (2000), 178–184.
- [21] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. 2017. Attenuate locally, win globally: An attenuation-based framework for online stochastic matching with timeouts. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS’17)*.
- [22] Brian Brubach, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. 2018. Algorithms to approximate column-sparse packing problems. In *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 311–330.
- [23] Niv Buchbinder and Moran Feldman. 2016. Constrained submodular maximization via a non-symmetric technique. Retrieved from: *arXiv preprint arXiv:1611.03253* (2016).
- [24] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. 2014. Submodular maximization with cardinality constraints. In *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 1433–1452. DOI : <https://doi.org/10.1137/1.9781611973402.106>
- [25] Grigori Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.* 40, 6 (2011), 1740–1766. DOI : <https://doi.org/10.1137/080733991>
- [26] Robert Carr and Santosh Vempala. 2002. Randomized metarounding. *Rand. Struct. Alg.* 20, 3 (2002), 343–352.
- [27] Amit Chakrabarti, Chandra Chekuri, Anupam Gupta, and Amit Kumar. 2007. Approximation algorithms for the unsplittable flow problem. *Algorithmica* 47, 1 (2007), 53–78. DOI : <https://doi.org/10.1007/s00453-006-1210-5>
- [28] Yuk Hei Chan and Lap Chi Lau. 2012. On linear and semidefinite programming relaxations for hypergraph matching. *Math. Program.* 135, 1–2, Ser. A (2012), 123–148. DOI : <https://doi.org/10.1007/s10107-011-0451-5>
- [29] Barun Chandra and Magnús M. Halldórsson. 2001. Greedy local improvement and weighted set packing approximation. *J. Alg.* 39, 2 (2001), 223–240. DOI : <https://doi.org/10.1006/jagm.2000.1155>
- [30] Chandra Chekuri, Alina Ene, and Nitish Korula. 2009. Personal communication. <http://chekuri.cs.illinois.edu/papers/ufp-tree-full.pdf>.
- [31] Chandra Chekuri, Alina Ene, and Nitish Korula. 2009. Unsplittable flow in paths and trees and column-restricted packing integer programs. In *Approximation, Randomization, and Combinatorial Optimization*. Lecture Notes in Comput. Sci., Vol. 5687. Springer, Berlin, 42–55. DOI : https://doi.org/10.1007/978-3-642-03685-9_4
- [32] Chandra Chekuri, T. S. Jayram, and Jan Vondrák. 2015. On multiplicative weight updates for concave and submodular function maximization. In *Proceedings of the 6th Innovations in Theoretical Computer Science Conference (ITCS’15)*. ACM, New York, 201–210.
- [33] Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. 2006. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theor. Comput.* 2 (2006), 137–146. DOI : <https://doi.org/10.4086/toc.2006.v002a007>
- [34] Chandra Chekuri, Marcelo Mydlarz, and F. Bruce Shepherd. 2007. Multicommodity demand flow in a tree and packing integer programs. *ACM Trans. Alg.* 3, 3 (2007), Article 27, 23. DOI : <https://doi.org/10.1145/1273340.1273343>
- [35] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. 2010. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the IEEE 51st Symposium on Foundations of Computer Science (FOCS’10)*. IEEE Computer Society, Los Alamitos, CA, 575–584.
- [36] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. 2014. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM J. Comput.* 43, 6 (2014), 1831–1879. DOI : <https://doi.org/10.1137/110839655>

- [37] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. 2005. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 395–404.
- [38] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. 2008. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.* 33, 4 (2008), 945–964. DOI: <https://doi.org/10.1287/moor.1080.0330>
- [39] Khaled Elbassioni, Naveen Garg, Divya Gupta, Amit Kumar, Vishal Narula, and Arindam Pal. 2012. Approximation algorithms for the unsplittable flow problem on paths and trees. In *Proceedings of the 32nd International Conference on Foundations of Software Technology and Theoretical Computer Science*. Leibniz Int. Proc. Inform., Vol. 18, 267–275.
- [40] Alina Ene and Huy L. Nguyen. 2016. Constrained submodular maximization: Beyond 1/e. In *Proceedings of the IEEE 57th Symposium on Foundations of Computer Science (FOCS'16)*. IEEE, 248–257.
- [41] Moran Feldman, Joseph Naor, and Roy Schwartz. 2011. A unified continuous greedy algorithm for submodular maximization. In *Proceedings of the IEEE 52nd Symposium on Foundations of Computer Science (FOCS'11)*. IEEE Computer Society, Los Alamitos, CA, 570–579. DOI: <https://doi.org/10.1109/FOCS.2011.46>
- [42] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions. II. *Math. Prog. Stud.* 8 (1978), 73–87.
- [43] A. Frank and A. Gyárfás. 1978. How to orient the edges of a graph? In *Proceedings of the 5th Hungarian Colloquium on Combinatorics, Vol. I*. Colloq. Math. Soc. János Bolyai, Vol. 18. North-Holland, Amsterdam-New York, 353–364.
- [44] Z. Füredi, J. Kahn, and P. D. Seymour. 1993. On the fractional matching polytope of a hypergraph. *Combinatorica* 13, 2 (1993), 167–180. DOI: <https://doi.org/10.1007/BF01303202>
- [45] N. Garg, V. V. Vazirani, and M. Yannakakis. 1997. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica* 18, 1 (1997), 3–20. DOI: <https://doi.org/10.1007/BF02523685>
- [46] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. 2016. Adaptivity gaps for stochastic probing: Submodular and XOS functions. *CoRR* abs/1608.00673 (2016). Retrieved from: <http://arxiv.org/abs/1608.00673>
- [47] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. 2016. Algorithms and adaptivity gaps for stochastic probing. In *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA'16)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1731–1747. Retrieved from: <http://dl.acm.org/citation.cfm?id=2884435.2884555>.
- [48] Nicholas J. A. Harvey, Roy Schwartz, and Mohit Singh. 2014. Discrepancy without partial colorings. In *Approximation, Randomization, and Combinatorial Optimization*. Leibniz Int. Proc. Inform., Vol. 28, 258–273.
- [49] Elad Hazan, Shmuel Safra, and Oded Schwartz. 2006. On the complexity of approximating k -set packing. *Comput. Complex.* 15, 1 (2006), 20–39.
- [50] C. A. J. Hurkens and A. Schrijver. 1989. 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. Disc. Math.* 2, 1 (1989), 68–72. DOI: <https://doi.org/10.1137/0402008>
- [51] Jon Michael Kleinberg. 1996. *Approximation Algorithms for Disjoint Paths Problems*. Ph.D. Dissertation. MIT. Retrieved from: Citeseer <https://dspace.mit.edu/handle/1721.1/11013>.
- [52] Stavros G. Kolliopoulos and Clifford Stein. 2004. Approximating disjoint-path problems using packing integer programs. *Math. Prog.* 99, 1, Ser. A (2004), 63–87. DOI: <https://doi.org/10.1007/s10107-002-0370-6>
- [53] Petr Kolman. 2003. A note on the greedy algorithm for the unsplittable flow problem. *Inform. Proc. Lett.* 88, 3 (2003), 101–105. DOI: [https://doi.org/10.1016/S0020-0190\(03\)00351-X](https://doi.org/10.1016/S0020-0190(03)00351-X)
- [54] Petr Kolman and Christian Scheidele. 2006. Improved bounds for the unsplittable flow problem. *J. Alg.* 61, 1 (2006), 20–44. DOI: <https://doi.org/10.1016/j.jalgor.2004.07.006>
- [55] Ariel Kulik, Hadas Shachnai, and Tami Tamir. 2009. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA, 545–554.
- [56] Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. 2009/10. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM J. Disc. Math.* 23, 4 (2009/10), 2053–2078. DOI: <https://doi.org/10.1137/090750020>
- [57] Tom Leighton, Chi-Jen Lu, Satish Rao, and Aravind Srinivasan. 2001. New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM J. Comput.* 31, 2 (2001), 626–641. <https://doi.org/10.1137/S0097539700379760>
- [58] Avi Levy, Harishchandra Ramadas, and Thomas Rothvoss. 2016. Deterministic discrepancy minimization via the multiplicative weight update method. Retrieved from: *arXiv preprint arXiv:1611.08752* (2016).
- [59] Jian Li and Wen Yuan. 2013. Stochastic combinatorial optimization via Poisson approximation. In *Proceedings of the ACM Symposium on Theory of Computing (STOC'13)*. ACM, New York, 971–980. DOI: <https://doi.org/10.1145/2488608.2488731>
- [60] Shachar Lovett and Raghu Meka. 2015. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.* 44, 5 (2015), 1573–1582. DOI: <https://doi.org/10.1137/130929400>
- [61] C. S. J. A. Nash-Williams. 1964. Decomposition of finite graphs into forests. *J. London Math. Soc.* 1, 1 (1964), 12–12.

- [62] Shayan Oveis Gharan and Jan Vondrák. 2011. Submodular maximization by simulated annealing. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA, 1098–1116.
- [63] Alessandro Panconesi and Aravind Srinivasan. 1997. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM J. Comput.* 26, 2 (1997), 350–368. DOI: <https://doi.org/10.1137/S0097539793250767>
- [64] Ojas Parekh. 2011. Iterative packing for demand and hypergraph matching. In *Integer Programming and Combinatorial Optimization*. Lecture Notes in Computer Science, Vol. 6655. Springer, Heidelberg, 349–361. DOI: https://doi.org/10.1007/978-3-642-20807-2_28
- [65] Ojas Parekh and David Pritchard. 2015. Generalized hypergraph matching via iterated packing and local ratio. In *Approximation and Online Algorithms*. Lecture Notes in Computer Science, Vol. 8952. Springer, Cham, 207–223. DOI: https://doi.org/10.1007/978-3-319-18263-6_18
- [66] Yuval Peres, Mohit Singh, and Nisheeth Vishnoi. 2017. Random walks in polytopes and negative dependence. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*.
- [67] Jean-Claude Picard and Maurice Queyranne. 1982. A network flow solution to some nonlinear 0 – 1 programming problems, with applications to graph theory. *Networks* 12, 2 (1982), 141–159. <https://doi.org/10.1002/net.3230120206>
- [68] David Pritchard. 2009. Approximability of sparse integer programs. In *Proceedings of the European Symposium on Algorithms (ESA'09)*. Lecture Notes in Computer Science, Vol. 5757. Springer, Berlin, 83–94. DOI: https://doi.org/10.1007/978-3-642-04128-0_8
- [69] David Pritchard and Deeparnab Chakrabarty. 2011. Approximability of sparse integer programs. *Algorithmica* 61, 1 (2011), 75–93. DOI: <https://doi.org/10.1007/s00453-010-9431-z>
- [70] Vojtech Rödl. 1985. On a packing and covering problem. *Eur. J. Comb.* 6, 1 (1985), 69–78.
- [71] Thomas Rothvoss. 2017. Constructive discrepancy minimization for convex sets. *SIAM J. Comput.* 46, 1 (2017), 224–234. DOI: <https://doi.org/10.1137/141000282>
- [72] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. 1995. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM J. Disc. Math.* 8, 2 (1995), 223–250.
- [73] Aravind Srinivasan. 1997. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proceedings of the 38th Symposium on Foundations of Computer Science*. IEEE, 416–425.
- [74] Aravind Srinivasan. 1997. Improving the discrepancy bound for sparse matrices: Better approximations for sparse lattice approximation problems. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 692–701.
- [75] Aravind Srinivasan. 1999. Improved approximation guarantees for packing and covering integer programs. *SIAM J. Comput.* 29, 2 (1999), 648–670. DOI: <https://doi.org/10.1137/S0097539796314240>
- [76] G. Szekeres and Herbert S. Wilf. 1968. An inequality for the chromatic number of a graph. *J. Combin. Theor.* 4 (1968), 1–3.
- [77] Jan Vondrák. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the (STOC'08)*. ACM, New York, 67–74. DOI: <https://doi.org/10.1145/1374376.1374389>
- [78] Justin Ward. 2012. A $(k + 3)/2$ -approximation algorithm for monotone submodular k -set packing and general k -exchange systems. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science*. 42–53.
- [79] David Zuckerman. 2007. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theor. Comput.* 3 (2007), 103–128. DOI: <https://doi.org/10.4086/toc.2007.v003a006>

Received March 2018; revised April 2019; accepted July 2019