



Stochastic packing integer programs with few queries

Takanori Maehara² · Yutaro Yamaguchi^{1,2}

Received: 16 February 2018 / Accepted: 5 March 2019 / Published online: 15 March 2019
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2019

Abstract

We consider a stochastic variant of the packing-type integer linear programming problem, which contains random variables in the objective vector. We are allowed to reveal each entry of the objective vector by conducting a query, and the task is to find a good solution by conducting a small number of queries. We propose a general framework of adaptive and non-adaptive algorithms for this problem, and provide a unified methodology for analyzing the performance of those algorithms. We also demonstrate our framework by applying it to a variety of stochastic combinatorial optimization problems such as matching, matroid, and stable set problems.

Keywords Stochastic problems with queries · Packing problems · Linear programming (LP) · LP duality · Approximation algorithms

Mathematics Subject Classification 90C15 · 90C05 · 05C70 · 05B35 · 68W20 · 68W25

1 Introduction

1.1 Problem formulation

We study a stochastic variant of linear programming (LP) with the 0/1-integer constraint, which enables us to discuss such variants of various packing-type combinatorial optimization problems such as matching, matroid, and stable set problems in a uni-

A preliminary version of this paper appeared in SODA 2018.

✉ Yutaro Yamaguchi
yutaro_yamaguchi@ist.osaka-u.ac.jp
Takanori Maehara
takanori.maehara@riken.jp

¹ Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

² RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan

fied manner. Specifically, we introduce the *stochastic packing integer programming problem* defined as follows:

$$\begin{aligned} & \text{maximize} && \tilde{c}^\top x \\ & \text{subject to} && Ax \leq b, \\ & && x \in \{0, 1\}^m, \end{aligned} \tag{1.1}$$

where $A \in \mathbb{Z}_+^{n \times m}$ and $b \in \mathbb{Z}_+^n$, and \mathbb{Z}_+ denotes the set of nonnegative integers. The objective vector $\tilde{c} \in \mathbb{Z}_+^m$ is *stochastic* in the following sense.

- The entries \tilde{c}_j ($j = 1, 2, \dots, m$) are independent random variables with some hidden distributions for which we are given the following information: for each j ,
 - the domain of \tilde{c}_j is an integer interval $\{c_j^-, c_j^- + 1, \dots, c_j^+\}$ given by $c_j^-, c_j^+ \in \mathbb{Z}_+$, and
 - the probability that $\tilde{c}_j = c_j^+$ is at least a given constant $p \in (0, 1]$ (which is independent from j), i.e., $c_j^- \leq \tilde{c}_j \leq c_j^+ - 1$ occurs with probability at most $1 - p$.
- When an instance (A , b , and the above information on \tilde{c}) is given, the *realized values* of all \tilde{c}_j , denoted by c_j , are hiddenly fixed by nature according to the above distributions.
- For each j , we are allowed to conduct a query to reveal the realized value c_j of \tilde{c}_j .

Note that, since all \tilde{c}_j are independent, we can consider at any time that each realized value c_j is determined just when a query for j is conducted.

Example 1.1 Our problem captures the *stochastic matching problem* introduced by Blum et al. [7] as follows. In the stochastic matching problem, we are given an undirected graph $G = (V, E)$ such that each edge $e \in E$ is *realized* with probability at least $p \in (0, 1]$, and the goal is to find a large matching that consists of realized edges. We can know whether each edge is realized or not by conducting a query. A naive formulation of this situation as our problem is obtained by restricting the domain of $\tilde{c} \in \mathbb{Z}_+^E$ to $\{0, 1\}^E$, by letting $A \in \mathbb{Z}_+^{V \times E}$ be the vertex-edge incidence matrix of G , and by setting $b = 1$. Section 4.1.2 gives a more detailed discussion with general edge weights.

Our aim is to find a feasible solution to (1.1) with a large objective value by conducting a small number of queries. Note that we can definitely obtain an optimal solution by solving the corresponding non-stochastic problem after conducting queries for all j . Our interest is therefore in the trade-off between the number of queries and the quality of the obtained solution.

1.2 Our contributions and technique

1.2.1 Contributions

We propose a general framework of adaptive and non-adaptive algorithms for the stochastic packing integer programming problem. Here, an algorithm is *non-adaptive* if it reveals all queried items simultaneously, and *adaptive* otherwise.

In the adaptive strategy¹ (which is formally shown in Algorithm 1 in Sect. 2.1), we iteratively compute an optimal fractional solution $x \in [0, 1]^m$ to the *optimistic LP* (the LP relaxation of (1.1) in which all the unrevealed \tilde{c}_j are supposed to be c_j^+), and conduct a query for each element j with probability x_j . After the iterations, we find an integral feasible solution to the *pessimistic LP* (in which all the unrevealed \tilde{c}_j are supposed to be c_j^-) by using some algorithms for the corresponding non-stochastic problem.

Similarly, in the non-adaptive strategy (Algorithm 2), we iteratively compute an optimal fractional solution x to the optimistic LP, and round down each element j (i.e., suppose \tilde{c}_j to be c_j^- instead of revealing c_j) with probability x_j . After the iterations, we reveal all the rounded-down elements and find an integral feasible solution to the pessimistic LP.

In application, we need to decide how to execute the last step, and the performance of the resulting algorithm depends on combinatorial structure of each specific problem. Our main contribution is a *proof technique* for analyzing the performance of the algorithms. Using this technique, we obtain results for the problem classes summarized in Table 1.

1.2.2 Technique

Our technique is based on *LP duality* and *enumeration*. A brief overview of the technique follows, where we focus on the adaptive strategy.

Let $\tilde{\mu}$ be the optimal value of the *omniscient LP* (the LP relaxation of (1.1) in which all \tilde{c}_j are revealed). Note that $\tilde{\mu}$ is a random variable depending on the realization of \tilde{c}_j . Our goal is to evaluate the number of iterations T such that the optimal value of the pessimistic LP after T iterations is at least $(1 - \epsilon)\tilde{\mu}$ with high probability.² Then, if we have an LP-relative α -approximation algorithm [31] (which outputs an integral feasible solution whose objective value is at least α times the LP-optimal value) for the corresponding non-stochastic problem, we obtain a $(1 - \epsilon)\alpha$ -approximate solution to our problem with high probability.

To discuss the optimal value of the pessimistic LP, we consider the dual LP. By the LP strong duality, it is sufficient to prove that the dual pessimistic LP after T

¹ Algorithms 1 and 2 have freedom of the choices of algorithms for solving LPs and for finding an integral solution in the last step; in particular, the latter depends heavily on each specific problem before formulated as an integer LP. For this reason, we use the term “strategy” rather than “algorithm” to refer them.

² Here we consider two types of randomness together. One is on the realization of \tilde{c}_j , which is contained in the “stochastic” input and determines the omniscient optimal value $\tilde{\mu}$. The other is on the choice of queried elements, which is involved in our “randomized” algorithms and affects the pessimistic LP obtained after the iterations.

Table 1 Results obtained for the adaptive strategy, where n and m denote the number of vertices in the graph and the ground set size of the matroids (or the number of edges), respectively, in question

Problem	Approximation ratio	Number of iterations T
Bipartite matching	$1 - \epsilon$	$\log(1/\epsilon p)/\epsilon p$
Non-bipartite matching	$1 - \epsilon$	$\log(n/\epsilon)/\epsilon p$
k -Hypergraph matching	$(1 - \epsilon)/(k - 1 + 1/k)$	$(k \log(k/\epsilon p) + 1/\epsilon)/\epsilon p$
k -Column sparse PIP	$(1 - \epsilon)/2k$	$(k \log(k/\epsilon p) + 1/\epsilon)/\epsilon p$
Matroid (max. independent set)	$1 - \epsilon$	$\log(m/\epsilon)/\epsilon p$
Matroid intersection	$1 - \epsilon$	$\log(m/\epsilon)/\epsilon p$
k -Matroid intersection	$(1 - \epsilon)/k$	$k \log m \log(m/\epsilon)/\epsilon^3 p$
Matchoid	$(1 - \epsilon)2/3$	$\log(m/\epsilon)/\epsilon p$
Degree bounded matroid	$1 - \epsilon \begin{pmatrix} \text{each constraint} \\ \text{is violated} \\ \text{at most } d - 1 \end{pmatrix}$	$d \log(n/\epsilon)/\epsilon^2 p$
Stable set in chordal graphs	$1 - \epsilon$	$\log n/\epsilon p$
Stable set in t -perfect graphs	$1 - \epsilon$	$\log n \log(n/\epsilon)/\epsilon^3 p$

We omit $O(\cdot)$ in the iteration column. Also, all the coefficients are assumed to be $O(1)$. For the non-adaptive strategy, the approximation ratio is halved

iterations has no feasible solution whose objective value is less than $(1 - \epsilon)\tilde{\mu}$ with high probability.

Here, we introduce a finite set $W \subseteq \mathbb{R}_+^n$ of dual vectors, called a *witness cover*, for every possible objective value μ (a candidate of $\tilde{\mu}$) that satisfies the following property: if all $y \in W$ are infeasible, there is no feasible solution whose objective value is less than $(1 - \epsilon)\mu$. Intuitively, W represents all the candidates for dual feasible solutions whose objective values are less than $(1 - \epsilon)\mu$. We evaluate the probability that each $y \in W$ becomes infeasible after T iterations, and then estimate the sufficient number of iterations by using the union bound for W .

In application, we only need to show the existence of a small witness cover for each specific problem. We also give general techniques to construct small witness covers when the considered problem enjoys some nice properties, e.g., when the constraint system $Ax \leq b$, $x \geq 0$ is totally dual integral.

1.3 Related work

As described in Example 1.1, our stochastic packing integer programming problem generalizes the *stochastic (unweighted) matching problem* [3,4,7] and the *stochastic (unweighted) k -hypergraph matching problem* [7], which have recently been studied in EC (Economics and Computation) community. These problems are motivated to find an optimal strategy for kidney exchange [16,33].

For the stochastic unweighted matching problem, Blum et al. [7] proposed adaptive and non-adaptive algorithms that achieve approximation ratios of $(1 - \epsilon)$ and of $(1/2 - \epsilon)$, respectively, *in expectation*, by conducting $O(\log(1/\epsilon)/p^{2/\epsilon})$ queries per vertex. Their technique is based on the existence of disjoint short augmenting paths. Assadi

et al. [3] proposed adaptive and non-adaptive algorithms that respectively achieve the same approximation ratios *with high probability*, by conducting $O(\log(1/\epsilon p)/\epsilon p)$ queries per vertex. Their technique is based on the Tutte–Berge formula and vertex sparsification. Our proposed strategies coincide with those of Assadi et al. when they are applied to the stochastic unweighted matching problem and we always find integral optimal solutions to the LP relaxations, i.e., solve the (non-stochastic) unweighted matching problem every time. Our analysis looks similar to theirs since they both use the duality, but ours is simpler and can also be used for the weighted and capacitated situation. On the other hand, our analysis shows that $O(\log(n/\epsilon)/\epsilon p)$ queries per vertex are required,³ which is worse than theirs.

Recently, Assadi et al. [4] proposed a non-adaptive algorithm that achieves an approximation ratio of strictly better than $1/2$ *in expectation*. However, this technique is tailored to the unweighted matching problem, so we could not generalize it to our problem.

For the stochastic unweighted k -hypergraph matching problem, Blum et al. [7] proposed adaptive and non-adaptive algorithms that find $(2 - \epsilon)/k$ - and $(4 - \epsilon)/(k^2 + 2k)$ -approximate matchings, respectively, *in expectation*, by conducting $O(s_{k,\epsilon} \log(1/\epsilon)/p^{s_{k,\epsilon}})$ queries per vertex, where $s_{k,\epsilon}$ is a constant depending on k and ϵ . Their technique is based on the local search method of Hurkens and Schrijver [23]. For the adaptive case, our strategy achieves a worse approximation ratio than theirs because the same is true of the LP-based algorithm versus the local search. On the other hand, our algorithm requires an exponentially smaller number of queries and runs in polynomial time both in n and $1/\epsilon$. In addition, our algorithm can be used for the weighted case. For the non-adaptive case, our algorithm outperforms theirs, all in terms of approximation ratio, the number of queries, and running time.

Other variants of the stochastic packing integer programming problem with queries have been studied. However, many of them employ the *query-commit model* [12,14,15,28], in which the queried elements must be a part of the output. Some studies [1,5,10] also impose additional budget constraints on the number of queries. In the *stochastic probing problem* [2,21,22], both the queried and realized elements must satisfy given constraints. Blum et al. [8] studied a stochastic matching problem without query-commit condition, but with a budget constraint on the number of queries.

1.4 Organization

The rest of the paper is organized as follows. In Sect. 2, we describe our framework of adaptive and non-adaptive algorithms for the stochastic packing integer programming problem, and explain a general technique for providing a bound on the number of iterations. In Sect. 3, we outline how to construct a small witness cover in general. In Sect. 4, we apply the technique to a variety of specific combinatorial problems. In Sect. 5, we provide a vertex sparsification lemma that can be used to improve the performance of the algorithms for several problems.

³ Very recently, Behnezhad and Reyhani [6] claimed that the same algorithm as ours achieves an approximation ratio of $1 - \epsilon$ by conducting a constant number of queries that depends on only ϵ and p . Their analysis uses augmenting paths, like Blum et al. [7].

2 General framework

Throughout the paper (with one exception as remarked later), we assume that the constraints in (1.1) satisfy several reasonable conditions.

Assumption 2.1 We assume that $A \in \mathbb{Z}_+^{n \times m}$ and $b \in \mathbb{Z}_+^n$ in (1.1) satisfy the following three conditions:⁴

- (a) $b \geq 1$;
- (b) $A\chi_j \leq b$ for each $j = 1, 2, \dots, m$, where $\chi_j \in \{0, 1\}^m$ denotes the j -th unit vector;
- (c) $Ax \leq b$ and $x \geq 0$ imply $x \leq 1$.

We give a general framework of adaptive and non-adaptive algorithms for our problem in Sect. 2.1, and then describe a unified methodology for its performance analysis in Sect. 2.2. The main results are stated as Theorems 2.1 and 2.2, whose proofs are separately shown in Sect. 2.3.

2.1 Two strategies

To describe two strategies, we formally define two auxiliary problems, the *optimistic LP* and the *pessimistic LP*. We define the *optimistic vector* $\bar{c} \in \mathbb{Z}_+^m$ and the *pessimistic vector* $\underline{c} \in \mathbb{Z}_+^m$ as follows:

$$\bar{c}_j = \begin{cases} c_j & j \text{ has been queried,} \\ c_j^+ & \text{otherwise,} \end{cases} \quad \underline{c}_j = \begin{cases} c_j & j \text{ has been queried,} \\ c_j^- & \text{otherwise,} \end{cases} \quad (2.1)$$

where recall that $c_j \in \mathbb{Z}_+$ denotes the realized value of \tilde{c}_j . The optimistic and pessimistic LPs are obtained from the original stochastic problem (1.1) by replacing the objective vector \tilde{c} with \bar{c} and with \underline{c} , respectively, and by relaxing the constraint $x \in \{0, 1\}^m$ to $x \in \mathbb{R}_+^m$, where \mathbb{R}_+ denotes the set of nonnegative reals. By Assumption 2.1.c ($Ax \leq b$ and $x \geq 0$ imply $x \leq 1$), the relaxed constraint is equivalent to $x \in [0, 1]^m$. Note that these problems are no longer stochastic, i.e., contain no random variables.

Algorithm 1 Adaptive strategy.

- 1: **for** $t = 1, 2, \dots, T$ **do**
 - 2: Find an optimal solution x to the optimistic LP.
 - 3: For each $j = 1, \dots, m$, conduct a query to reveal \tilde{c}_j with probability x_j .
 - 4: **end for**
 - 5: Find an integral feasible solution to the pessimistic LP and return it.
-

⁴ The first two are assumed without loss of generality (by removing the corresponding constraints and variables if violated). The third one is for simplicity, which holds for most of applications. The generalizability to remove it is discussed in Sect. 4.1.4 with a specific application.

Algorithm 2 Non-adaptive strategy.

```

1: for  $t = 1, 2, \dots, T$  do
2:   Find an optimal solution  $x$  to the optimistic LP.
3:   For each  $j = 1, \dots, m$ , suppose  $\tilde{c}_j = c_j^-$  with probability  $x_j$ .
4: end for
5: For every  $j$  with  $\tilde{c}_j = c_j^-$  supposed at Line 3, conduct a query to reveal  $\tilde{c}_j$ .
6: Find an integral feasible solution to the pessimistic LP and return it.

```

First, we describe the *adaptive strategy* shown in Algorithm 1. In this strategy, we iteratively compute an optimal solution $x \in [0, 1]^m$ to the optimistic LP,⁵ and reveal each \tilde{c}_j with probability x_j . After T iterations, we find an integral feasible solution to the pessimistic LP, where we have freedom of the choice of algorithms for the corresponding non-stochastic problem. As remarked in Sect. 1.2, how to execute the last step depends heavily on each specific problem.

Next, we describe the *non-adaptive strategy* shown in Algorithm 2. As with the adaptive strategy, we solve the optimistic LP at each step. To be non-adaptive, the algorithm tentatively assigns values to \tilde{c}_j pessimistically instead of revealing their realized values. After the iterations, it reveals all these values and then computes an integral feasible solution to the pessimistic LP by some algorithms for the non-stochastic problem.

2.2 Performance analysis

We now analyze the performance of algorithms within our framework. First, we consider the adaptive strategy (Algorithm 1). As described in Sect. 1.1, we evaluate the trade-off between the following two factors, each of which is reasonably decomposed into two factors.

(1) The number of conducted queries.

- (1-a) *Expected number of queries at Line 3.* If this number is large, the algorithm may reveal all relevant \tilde{c}_j in a few iterations, making the algorithm trivial.
- (1-b) *Required number of iterations T at Line 1.* If T is very large, then, as in (1-a), the algorithm may reveal all relevant \tilde{c}_j , making the algorithm trivial.

(2) The quality of the output solution.

Basically, we want to find a feasible solution to (1.1) with a large objective value, which is at most the *omniscient* optimal value of (1.1) after all \tilde{c}_j are revealed.

- (2-a) *Closeness between the pessimistic and omniscient LPs.* The omniscient optimal value of (1.1) is at most the optimal value $\tilde{\mu}$ of the *omniscient LP*, which is obtained by revealing all \tilde{c}_j and by relaxing $x \in \{0, 1\}^m$ to $x \in \mathbb{R}_+^m$. If the pessimistic LP-optimal value at Line 5 is close to $\tilde{\mu}$, then, at least as an LP, the pessimistic problem is close to the problem that we want to solve.

⁵ Note that, if the optimal solution x is written as a convex combination $\sum_i \lambda_i x^{(i)}$ of basic feasible solutions $x^{(i)}$, then every $x^{(i)}$ is also optimal and one can replace x with any $x^{(i)}$. In particular, when the considered polyhedron is integral (i.e., every extreme point is an integral vector), Algorithms 1 and 2 can be derandomized based on this observation.

- (2-b) *LP-relative approximation ratio at Line 5.* If one can find an integral feasible solution such that the ratio between its objective value and the LP-optimal value is bounded, then, combined with (2-a), a reasonable bound on the objective value of the output solution can be obtained.

Essentially, (1-a) and (2-b) are properties of each specific problem and its LP formulation. Thus, we postpone these two factors to the discussion on applications in Sect. 4, and focus on (1-b) and (2-a) in the general study in this section. That is, our goal here is to estimate T such that the optimal value of the pessimistic LP after T iterations is at least $(1 - \epsilon)\tilde{\mu}$ with high probability, where $\tilde{\mu}$ is the optimal value of the omniscient LP and $\epsilon > 0$ is a parameter one can choose. Note again that $\tilde{\mu}$ is a random variable depending on the realization of \tilde{c}_j .

To evaluate the number of iterations T , we consider the dual of the pessimistic LP:

$$\begin{aligned} & \text{minimize} && y^\top b \\ & \text{subject to} && y^\top A \geq c^\top, \\ & && y \in \mathbb{R}_+^n. \end{aligned} \tag{2.2}$$

By the LP strong duality, it is sufficient to evaluate the probability that this dual LP has no feasible solution whose objective value is less than $(1 - \epsilon)\tilde{\mu}$.

Now we introduce the notion of a *witness cover*, which is the most important concept in this study. Intuitively, a witness cover for $\mu \in \mathbb{R}_+$ is a set of “representatives” of all the dual feasible solutions with objective values of at most $(1 - \epsilon)\mu$. More specifically, for any primal objective vector, if some dual feasible solution has the objective value at most $(1 - \epsilon)\mu$, then a witness cover contains at least one such dual solution.

Definition 2.1 Let $A \in \mathbb{Z}_+^{n \times m}$, $b \in \mathbb{Z}_+^n$, and $\epsilon, \epsilon' \in \mathbb{R}_+$ with $0 < \epsilon' \leq \epsilon$. A finite set $W \subseteq \mathbb{R}_+^n$ of dual vectors is an (ϵ, ϵ') -*witness cover* for $\mu \in \mathbb{R}_+$ if it satisfies the following two properties.

1. For every $c \in \mathbb{Z}_+^m$, if $y^\top A \geq c^\top$ is violated (i.e., $(y^\top A)_j < c_j$ for some j) for all $y \in W$, then $y^\top A \geq c^\top$ is violated for all $y \in \mathbb{R}_+^n$ with $y^\top b \leq (1 - \epsilon)\mu$.
2. $y^\top b \leq (1 - \epsilon')\mu$ holds for all $y \in W$.

Example 2.1 Consider the bipartite matching case (see Sect. 4.1.1 for the detail). In the LP relaxation of the naive formulation (4.1), the constraint system is totally dual integral (see Sect. 3.1 for the detail), and each dual vector is an assignment of nonnegative reals to vertices, whose sum is the objective value. Hence, the set of assignments of nonnegative integers to vertices whose sum is at most $(1 - \epsilon)\mu$ is an (ϵ, ϵ) -witness cover for μ .

During the iterations, the constraints in the dual pessimistic LP (2.2) become successively stronger. Hence, for any witness cover W for the omniscient LP-optimal value $\tilde{\mu}$, every $y \in W$ eventually becomes infeasible to (2.2) (by the second condition in Definition 2.1). By evaluating the probability that all $y \in W$ become infeasible after T iterations, we obtain a bound on the required number of iterations. Note again that $\tilde{\mu}$ is a random variable, and hence we assume that there exists a relatively small witness cover for every possible objective value μ , which can be restricted to $\mu \geq 1$ due to Assumption 2.1.b (see the proof for the detail).

Theorem 2.1 *Let $M \in \mathbb{R}_+$, and suppose that there exists an (ϵ, ϵ') -witness cover of size at most M^μ for every $\mu \geq 1$. Then, by taking*

$$T \geq \frac{\Delta_c}{\epsilon' p} \log \left(\frac{M}{\delta} \right), \quad (2.3)$$

the pessimistic LP at Line 5 of Algorithm 1 has a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \delta$, where $\Delta_c = \max_j (c_j^+ - c_j^-)$ and $0 < \delta < 1$.

For the non-adaptive algorithm (Algorithm 2), by conducting a similar analysis with a case analysis, we obtain the required number of iterations with a provable approximation ratio.

Theorem 2.2 *Under the same assumption as Theorem 2.1, by taking T as (2.3), the pessimistic LP at Line 6 of Algorithm 2 has a $(1 - \epsilon)/2$ -approximate solution with probability at least $1 - \delta$.*

These theorems show that if there exists a small witness cover (for each possible μ), Algorithms 1 and 2 will find good solutions in a reasonable number of iterations. It is worth emphasizing that we only have to prove the existence of such a witness cover, i.e., we do not have to construct it algorithmically. We discuss how to prove the existence of such witness covers (i.e., how to construct them theoretically) in general and in each specific application, in Sects. 3 and 4, respectively.

2.3 Proofs of main theorems

Proof of Theorem 2.1

Let $\tilde{\mu}$ be the optimal value of the omniscient LP. If $\tilde{\mu} = 0$ then the statement obviously holds (with probability 1). Thus we restrict ourselves to the case when $\tilde{\mu} > 0$. Note that $\tilde{\mu} > 0$ implies $\tilde{\mu} \geq 1$ as follows. If $\tilde{\mu} > 0$ then $\tilde{c}_j = c_j \geq 1$ for some j , and by Assumption 2.1.b, the j -th unit vector $\chi_j \in \{0, 1\}^m$ is feasible (i.e., $A\chi_j \leq b$); therefore $\tilde{\mu} \geq c^\top \chi_j = c_j \geq 1$.

For each $\mu \geq 1$, fix an (ϵ, ϵ') -witness cover W_μ of size $|W_\mu| \leq M^\mu$. We first evaluate the probability that each $y \in W_{\tilde{\mu}}$ is feasible after T iterations. Since some \tilde{c}_j is newly revealed, some constraints may be violated (i.e., $(y^\top A)_j < c_j$ may happen). Once y has become infeasible, it never returns to feasible due to the monotonicity of \underline{c} throughout Algorithm 1. Therefore, y is feasible after T iterations only if y is feasible at every iteration step.

Fix $t = 1, 2, \dots, T$, and we evaluate the probability that a vector y in each witness cover that is feasible at the beginning of the t -th step remains feasible at the end of the step. Let $\bar{c}, \underline{c} \in \mathbb{Z}_+^m$ be the optimistic and pessimistic vectors, respectively, at Line 2 in the t -th step, and $\bar{\mu}, \underline{\mu} \in \mathbb{R}_+^m$ the optimal values of the corresponding LPs. Note that $\bar{\mu}$ and $\underline{\mu}$ are respectively upper and lower bounds on $\tilde{\mu}$ at that time.

Claim 2.1 *For every $\mu \in [\underline{\mu}, \bar{\mu}]$ and each $y \in W_\mu$ with $y^\top A \geq \underline{c}^\top$, the probability that y is feasible after Line 3 is at most $\exp(-\epsilon' p \mu / \Delta_c)$.*

Proof Since y is feasible at the beginning of the step, y is feasible after Line 3 only if no possibly violated constraint is revealed to be c_j^+ . We can evaluate the number of possibly violated constraints at this step using the following inequality:

$$\bar{c}^\top x \leq \bar{c}^\top x + y^\top (b - Ax) = y^\top b + (\bar{c}^\top - y^\top A)x, \quad (2.4)$$

where $x \in [0, 1]^m$ is the optimal solution to the optimistic LP obtained in Line 2. Since the optimistic vector \bar{c} dominates the actual vector \tilde{c} (irrespective of which values are realized), we have $\bar{c}^\top x \geq \tilde{\mu}$. Since $y \in W_\mu$, we have $y^\top b \leq (1 - \epsilon')\mu$. Therefore, we derive from (2.4)

$$\epsilon'\mu \leq (\bar{c}^\top - y^\top A)x \leq \sum_{j: \text{violated}} (c_j^+ - c_j)x_j \leq \Delta_c \sum_{j: \text{violated}} x_j, \quad (2.5)$$

where we say that j is *violated* if $(y^\top A)_j < c_j$ for the realized value c_j of \tilde{c}_j , and note that $\bar{c}_j \leq c_j^+$, $(y^\top A)_j \geq c_j$, and $c_j \geq c_j^+ - \Delta_c$ for every j . Since the left-hand side of (2.5) is positive, there must exist possibly violated constraints in the support of x , and if one of them, say \tilde{c}_j , is revealed (with probability x_j) as c_j^+ (with probability at least p), then y becomes infeasible. Then the probability that y is still feasible after this step is at most

$$\prod_{j: \text{violated}} (1 - px_j) \leq \exp \left(-p \sum_{j: \text{violated}} x_j \right) \leq \exp \left(\frac{-p\epsilon'\mu}{\Delta_c} \right). \quad (2.6)$$

□

By applying Claim 2.1 to $\tilde{\mu}$ (the omniscient LP-optimal value) T times, we obtain that the probability that each $y \in W_{\tilde{\mu}}$ is feasible after T iterations is at most $\exp(-\epsilon'p\tilde{\mu}T/\Delta_c)$. By the union bound, the probability that $W_{\tilde{\mu}}$ has at least one feasible solution to the dual pessimistic LP (2.2) after T iterations is at most $|W_{\tilde{\mu}}| \exp(-\epsilon'p\tilde{\mu}T/\Delta_c)$, which is at most $\exp(\tilde{\mu} \log M - \epsilon'p\tilde{\mu}T/\Delta_c)$. By taking $T \geq \Delta_c \log(M/\delta)/\epsilon'p$, the latter value is bounded by $\exp(\tilde{\mu} \log \delta) \leq \delta$ (recall that $\tilde{\mu} \geq 1$ and $0 < \delta < 1$). By the definition of witness cover and strong duality, we conclude that the optimal value of the pessimistic LP at Line 5 of Algorithm 1 is at least $(1 - \epsilon)\tilde{\mu}$ with probability at least $1 - \delta$.

Proof of Theorem 2.2

The following proof is a simple extension of Theorem 5.1 in Assadi et al. [3] for the stochastic matching problem.

Let $\tilde{\mu}$ be the optimal value of the omniscient LP, and we assume $\tilde{\mu} \geq 1$ as in the proof of Theorem 2.1. In the above analysis of Algorithm 1, it is ensured that there exists a solution x with $\bar{c}^\top x \geq \tilde{\mu}$, which ensured that the last pessimistic LP in Algorithm 1 has an optimal value of at least $(1 - \epsilon)\tilde{\mu}$. However, in the non-adaptive case, we may not be able to find such a solution because each \tilde{c}_j is not revealed but is rounded-down.

To overcome this issue, we define $\tilde{\mu}' \in \mathbb{R}_+$ as the minimum objective value obtained at Line 2 of Algorithm 2. Note that, since the optimal value of the optimistic LP solved at Line 2 is monotonically non-increasing, $\tilde{\mu}'$ is the objective value obtained at the T -th step. By applying Claim 2.1 to $\tilde{\mu}'$ (instead of $\tilde{\mu}$) T times, we obtain the following claim.

Claim 2.2 *By taking $T \geq \Delta_c \log(M/\delta)/\epsilon' p$ in Algorithm 2, the optimal value of the pessimistic LP at Line 6 is at least $(1 - \epsilon)\tilde{\mu}'$ with probability at least $1 - \delta$.*

If $\tilde{\mu}' \geq \tilde{\mu}/2$, we can immediately prove the theorem. Thus, we consider the case $\tilde{\mu}' < \tilde{\mu}/2$, obtaining the following claim.

Claim 2.3 *If $\tilde{\mu}' < \tilde{\mu}/2$, the optimal value of the pessimistic LP at Line 6 of Algorithm 2 is at least $\tilde{\mu}/2$.*

Proof We use the subscripts R and N to denote the revealed and unrevealed entries in the primal vector, respectively, i.e., $\tilde{c}_R = c_R$ has been realized and the rest \tilde{c}_N has not been revealed. Let $x^* \in \mathbb{R}_+^m$ be an optimal solution to the (primal) omniscient LP (which is a random variable depending on the realization of \tilde{c}_N). We then have

$$c_R^\top x_R^* + \tilde{c}_N^\top x_N^* = \tilde{\mu}, \quad (2.7)$$

Since $(0, x_N^*) \in \mathbb{R}_+^m$ is a feasible solution to the optimistic LP at the last iteration, we have

$$c_N^{+\top} x_N^* \leq \tilde{\mu}' < \tilde{\mu}/2. \quad (2.8)$$

Therefore, the objective value for x^* in the pessimistic LP at Line 6 is bounded by

$$\underline{c}^\top x^* \geq c_R^\top x_R^* \geq c_R^\top x_R^* + (\tilde{c}_N^\top - c_N^{+\top})x_N^* > \tilde{\mu}/2. \quad (2.9)$$

This means that the pessimistic LP-optimal value is at least $\tilde{\mu}/2$. □

This concludes the theorem.

3 Constructing witness covers

Our technique requires us to prove the existence of a small witness cover. Here, we describe general strategies for constructing small witness covers.

3.1 Totally dual integral case

A system $Ax \leq b, x \geq 0$ is *totally dual integral (TDI)* if, for every integral objective vector $c \in \mathbb{Z}^m$, the dual problem $\min\{y^\top b : y^\top A \geq c^\top, y \geq 0\}$ has an integral optimal solution $y \in \mathbb{Z}_+^n$ (unless it is infeasible). Note that every TDI system yields an integral polyhedron (see, e.g., [34] for the detail). Hence, if we obtain a basic optimal

solution to the optimistic LP (in Line 2 of Algorithms 1 and 2), then we do not need randomization in conducting query (cf. the footnote 5 in Sect. 2.1).

If the system is TDI, we can construct a witness cover by enumerating all possible integral dual vectors as follows.

Lemma 3.1 *If the system $Ax \leq b$, $x \geq 0$ is TDI, the following set $W \subseteq \mathbb{R}_+^n$ is an (ϵ, ϵ) -witness cover for $\mu \geq 1$ such that $|W| = \exp\left(O\left(\mu \log\left(1 + \frac{n}{\mu}\right)\right)\right)$:*

$$W = \{y \in \mathbb{Z}_+^n : y^\top b \leq (1 - \epsilon)\mu\}. \quad (3.1)$$

Proof It is clear that W is an (ϵ, ϵ) -witness cover for μ , so it only remains to evaluate the cardinality of W . We see that $|W|$ is at most the number of nonnegative vectors whose entries sum to at most $\lfloor \mu \rfloor$, which can be counted by distributing k ($\leq \lfloor \mu \rfloor$) tokens among n entries, giving

$$|W| \leq \sum_{k=0}^{\lfloor \mu \rfloor} \binom{n+k-1}{k} = \binom{n+\lfloor \mu \rfloor}{\lfloor \mu \rfloor} \leq \left(\frac{e(n+\lfloor \mu \rfloor)}{\lfloor \mu \rfloor}\right)^{\lfloor \mu \rfloor} = e^{O\left(\mu \log\left(1 + \frac{n}{\mu}\right)\right)}. \quad (3.2)$$

□

Note that the same counting technique can be used when the system is totally dual $1/k$ -integral (TDI/ k), i.e., the existence of a dual optimal solution where each entry is a multiple of $1/k$ is guaranteed.

3.2 Non-TDI case

If the system is not TDI, we have to deal with fractional dual vectors. To enumerate these fractional vectors, we discretize the dual vectors, requiring the discretization to have the following property: if there exists a feasible y such that $y^\top b \leq (1 - \epsilon)\mu$, there exists a feasible discretized y' such that $y'^\top b \leq (1 - \epsilon/2)\mu$. Here, we consider two possible situations: dual sparse case and general case.

3.2.1 Dual sparse case

If there exists a sparse dual optimal solution, we can simply discretize the dual vectors to obtain a good discretized solution as follows.

Lemma 3.2 *For positive μ , ϵ , and γ , if there exists $y \in \mathbb{R}_+^n$ such that $y^\top b \leq (1 - \epsilon)\mu$ and $|\text{supp}(y)| \leq \gamma\mu$, then there exists $y' \in \prod_{i=1}^n (\epsilon/2b_i\gamma)\mathbb{Z}_+$ such that $y'^\top A \geq y^\top A$ and $y'^\top b \leq (1 - \epsilon/2)\mu$.*

Proof A suitable y' can be obtained by rounding up the i -th entry of y to the next multiple of $\epsilon/2b_i\gamma$ for each i . □

Now that the existence of a discretized solution whose objective value is almost the same as any sparse solution has been guaranteed, we can construct a witness cover by enumerating all the discretized vectors.

Lemma 3.3 *Under the assumption given in Lemma 3.2, the following set $W \subseteq \mathbb{R}_+^n$ is an $(\epsilon, \epsilon/2)$ -witness cover for $\mu \geq 1$, whose cardinality is $|W| = \exp\left(O\left(\mu\left(\gamma \log \frac{n}{\gamma\mu} + \frac{1}{\epsilon}\right)\right)\right)$:*

$$W = \left\{ y \in \prod_{i=1}^n \left(\frac{\epsilon}{2b_i\gamma} \right) \mathbb{Z}_+ : y^\top b \leq \left(1 - \frac{\epsilon}{2}\right)\mu, |\text{supp}(y)| \leq \gamma\mu \right\}. \quad (3.3)$$

Proof By Lemma 3.2, W is an $(\epsilon, \epsilon/2)$ -witness cover for μ . We evaluate the cardinality of W as follows. We first select s ($\leq \gamma\mu$) entries for the support of y , and then distribute k ($< 2\mu/\epsilon$) tokens among these entries, where each token contributes $\epsilon/2$ to the objective value. In the nontrivial case when $\mu_1 := \gamma\mu \geq 1$ and $\mu_2 := 2\mu/\epsilon \geq 1$, the number of these patterns is bounded by

$$\begin{aligned} |W| &\leq \sum_{s=1}^{\lfloor \mu_1 \rfloor} \binom{n}{s} \sum_{k=0}^{\lfloor \mu_2 \rfloor} \binom{s+k-1}{k} \\ &\leq \lfloor \mu_1 \rfloor \left(\frac{en}{\lfloor \mu_1 \rfloor} \right)^{\lfloor \mu_1 \rfloor} \lfloor \mu_2 \rfloor \left(\frac{e(\lfloor \mu_1 \rfloor + \lfloor \mu_2 \rfloor)}{\lfloor \mu_2 \rfloor} \right)^{\lfloor \mu_2 \rfloor} \\ &\leq \lfloor \mu_1 \rfloor \left(\frac{en}{\lfloor \mu_1 \rfloor} \right)^{\lfloor \mu_1 \rfloor} \lfloor \mu_2 \rfloor e^{\lfloor \mu_1 \rfloor + \lfloor \mu_2 \rfloor} = \exp\left(O\left(\mu_1 \log \frac{n}{\mu_1} + \mu_2\right)\right). \end{aligned}$$

□

3.2.2 General case

When the optimal dual solutions are not sparse, the results of simple discretization are useless. However, even in such a case, there is a good discretized solution. Let y be a feasible dual vector. Then by applying *randomized rounding* [32] to y , we obtain a suitable discretized vector y' with a positive probability. Formally, the following theoretical guarantee is obtained.

Theorem 3.1 (Kolliopoulos and Young [25]) *Any feasible LP $\min_y \{y^\top b : y^\top A \geq c^\top, y \geq 0\}$ has a $(1 + \epsilon/2)$ -approximate solution whose entries are multiples of $\theta = \Theta\left(\frac{\epsilon^2}{\log m}\right)$, where m is the dimension of c .*

We use this theorem as an existence theorem. If there is an optimal dual solution with objective value of at most $(1 - \epsilon)\mu$, this theorem shows that there exists a dual feasible solution whose entries are multiples of θ with an objective value of at most $(1 + \epsilon/2)(1 - \epsilon)\mu \leq (1 - \epsilon/2)\mu$. By enumerating the dual vectors whose entries are multiples of θ , we can obtain a witness cover.

Lemma 3.4 Let $\theta = \Theta\left(\frac{\epsilon^2}{\log m}\right)$. The following set $W \subseteq \mathbb{R}_+^n$ is an $(\epsilon, \epsilon/2)$ -witness cover for $\mu \geq 1$ such that $|W| = \exp\left(O\left(\frac{\mu \log m}{\epsilon^2} \log\left(1 + \frac{n}{\mu}\right)\right)\right)$:

$$W = \left\{ y \in \theta \mathbb{Z}_+^n : y^\top b \leq \left(1 - \frac{\epsilon}{2}\right) \mu \right\}. \quad (3.4)$$

Proof By Theorem 3.1, W is an $(\epsilon, \epsilon/2)$ -witness cover for μ . We evaluate the cardinality of W as follows. Let $\mu' := \lfloor \mu/\theta \rfloor$. The number of ways of distributing k ($\leq \mu'$) tokens among n entries is bounded by

$$\begin{aligned} |W| &\leq \sum_{k=0}^{\mu'} \binom{n+k-1}{k} \leq \mu' \left(\frac{e(n+\mu')}{\mu'} \right)^{\mu'} \\ &\leq \mu' e^{\mu'(1+\log(1+n/\mu'))} = \exp\left(O\left(\frac{\mu}{\theta} \log\left(1 + \frac{\theta n}{\mu}\right)\right)\right). \end{aligned} \quad (3.5)$$

□

3.3 Exponentially many constraints

Some problems, such as the non-bipartite matching problem and matroid problems, have exponentially many constraints. In such cases, it is impossible to enumerate all the candidates naively as we have done in the previous sections.

Sometimes, this difficulty is overcome by identifying the granularity of the dual solution (i.e., TDI, dual sparse, or general), and then bounding the number of possible dual patterns by exploiting the combinatorial structure; see next section for concrete examples.

4 Applications

In this section, we demonstrate our proposed framework by applying it to several stochastic combinatorial problems. We only describe the results for the adaptive strategy (Algorithm 1) as the results for the non-adaptive strategy (Algorithm 2) can easily be obtained analogously.

4.1 Matching problems

In this section, unless otherwise noted, n denotes the number of vertices in the (hyper)graph in question.

4.1.1 Bipartite matching

We first demonstrate how to use our technique for the bipartite matching problem. Let (V, E) be a bipartite graph and $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic edge weight. The bipartite matching problem can then be represented as

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in \delta(u)} x_e \leq 1 \quad (u \in V), \\ & && x \in \{0, 1\}^E, \end{aligned} \quad (4.1)$$

where $\delta(u) = \{e \in E : u \in e\}$. The Kőnig–Egerváry theorem [19,26] shows that the LP relaxation of this system is TDI, so Algorithm 1 has an approximation ratio of $(1 - \epsilon)$ with high probability for sufficiently large T . Moreover, if the algorithm finds an integral solution to the optimistic problem in Line 2, it reveals a matching in each iteration, and hence at most T edges per vertex in total. Finally, by Lemma 3.1, there exists an (ϵ, ϵ) -witness cover of size $e^{O(\mu \log n)}$ for each $\mu \geq 1$. We can therefore obtain the following result from Theorem 2.1.

Corollary 4.1 *By taking $T = \Omega(\Delta_c \log(n/\epsilon)/\epsilon p)$, for the bipartite matching problem, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.*

This result is improved by using the vertex sparsification lemma in Sect. 5.

4.1.2 Non-bipartite matching

Next, we consider the non-bipartite matching problem. Let (V, E) be a graph and $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic edge weight. A naive formulation of this problem is as follows:

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in \delta(u)} x_e \leq 1 \quad (u \in V), \\ & && x \in \{0, 1\}^E. \end{aligned} \quad (4.2)$$

It is known that the LP relaxation of this system is TDI/2 (totally dual half-integral) and has an integrality gap of $3/2$ [34]. Therefore, by the same argument as in the bipartite matching problem, we can show that Algorithm 1 has an approximation ratio of $(2 - \epsilon)/3$ with high probability if $T = \Omega(\Delta_c \log(n/\epsilon)/\epsilon p)$.

To improve this approximation ratio, we consider a strengthened formulation by adding the blossom inequalities:

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in \delta(u)} x_e \leq 1 \quad (u \in V), \\ & && \sum_{e \in E(S)} x_e \leq \left\lfloor \frac{|S|}{2} \right\rfloor \quad (S \in \mathcal{V}_{\text{odd}}), \\ & && x \in \{0, 1\}^E, \end{aligned} \quad (4.3)$$

where $E(S) = \{e \in E : e \subseteq S\}$ and $\mathcal{V}_{\text{odd}} = \{S \subseteq V : |S| \text{ is odd and at least } 3\}$. Cunningham and Marsh [13] showed that this system is TDI, so our algorithm has an approximation ratio of $(1 - \epsilon)$ with high probability for sufficiently large T . Moreover, the number of revealed edges is at most T per vertex.

The only remaining issue is the number of iterations. Since the system (4.3) has exponentially many constraints, we have to exploit its combinatorial structure to reduce the number of possibilities. The dual problem is given by

$$\begin{aligned} & \text{minimize} \quad \sum_{u \in V} y_u + \sum_{S \in \mathcal{V}_{\text{odd}}} \left\lfloor \frac{|S|}{2} \right\rfloor z_S =: \tau(y, z) \\ & \text{subject to} \quad y_u + y_v + \sum_{S \in \mathcal{V}_{\text{odd}} : \{u, v\} \subseteq S} z_S \geq \tilde{c}_e \quad (e = \{u, v\} \in E), \\ & \quad y \in \mathbb{R}_+^V, z \in \mathbb{R}_+^{\mathcal{V}_{\text{odd}}}. \end{aligned} \quad (4.4)$$

For $\mu \geq 1$, let us define a set $W \subseteq \mathbb{R}_+^V \times \mathbb{R}_+^{\mathcal{V}_{\text{odd}}}$ by

$$W = \left\{ (y, z) \in \mathbb{R}_+^V \times \mathbb{R}_+^{\mathcal{V}_{\text{odd}}} : \tau(y, z) \leq (1 - \epsilon)\mu \right\}. \quad (4.5)$$

It is clear that W is an (ϵ, ϵ) -witness cover, and we can evaluate the size of W as follows.

Claim 4.1 $|W| = e^{O(\mu \log n)}$.

Proof We count the candidates for y and z separately.

Since $\sum_i y_i < \mu$, the number of candidates for y is at most

$$\sum_{k=0}^{\lfloor \mu \rfloor} \binom{n}{k} \leq (\lfloor \mu \rfloor + 1) \left(\frac{en}{\lfloor \mu \rfloor} \right)^{\lfloor \mu \rfloor} = e^{O(\mu \log n)}. \quad (4.6)$$

To count the number of candidates for z , we regard z as a multiset, e.g., if $z_S = 2$ then we think there are two S . Let s_i ($i = 1, 2, \dots, k$) be the size of each set contained in z . Then we have

$$s_1 + \dots + s_k \leq 3 \left(\left\lfloor \frac{s_1}{2} \right\rfloor + \dots + \left\lfloor \frac{s_k}{2} \right\rfloor \right) < 3\mu. \quad (4.7)$$

Therefore, the number of candidates for z is given by

$$\sum_{k=0}^{\lfloor \mu \rfloor} \sum_{\substack{s_1 + \dots + s_k \leq 3\mu \\ s_1, \dots, s_k \geq 3}} \binom{n}{s_1} \cdots \binom{n}{s_k} \leq \sum_k \binom{3\lfloor \mu \rfloor + k - 1}{k} n^\mu = e^{O(\mu \log n)}. \quad (4.8)$$

By multiplying the number of candidates for y and z , we obtain the required result. \square

Therefore, we obtain the following.

Corollary 4.2 *By taking $T = \Omega(\Delta_c \log(n/\epsilon)/\epsilon p)$, for the non-bipartite matching problem, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.*

The same analysis can be applied to the (simple) b -matching problem.

Relationship to the analysis of Assadi et al. The adaptive and non-adaptive algorithms of Assadi et al. [3] for the unweighted non-bipartite matching problem are within our framework (Algorithms 1 and 2, respectively), and their analysis utilizes the Tutte–Berge formula. They showed that the required number of iterations is $O(\log(n/\epsilon\mu)/\epsilon p)$, and it is reduced to $O(\text{poly}(p, 1/\epsilon))$ by using the vertex sparsification lemma.

For the unweighted problem, our analysis gives a weaker result than theirs. However, since no simple alternative to the Tutte–Berge formula for the weighted problem is known, our analysis is more general than theirs.

4.1.3 k -Hypergraph matching

Let (V, E) be a k -uniform hypergraph, i.e., E is a set family on V whose each element $e \in E$ has size exactly k . Let $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic edge weight. The k -hypergraph matching problem can be represented as

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in \delta(u)} x_e \leq 1 \quad (u \in V), \\ & && x \in \{0, 1\}^E, \end{aligned} \tag{4.9}$$

where $\delta(u) = \{e \in E : u \in e\}$. Chan and Lau [9] proved that the LP relaxation of the above system has an integrality gap of $\alpha := 1/(k-1+1/k)$, and they also proposed an LP-relative α -approximation algorithm. Since at most one edge per vertex is revealed in expectation due to the constraint $\sum_{e \in \delta(u)} x_e \leq 1$, the expected number of revealed hyperedges per vertex is $O(T)$ in total.

The only remaining issue is the number of iterations. Since the system (4.9) has polynomially many constraints and is not TDI, we have to discretize the dual variables. The corresponding dual problem is given by

$$\begin{aligned} & \text{minimize} && \sum_{u \in V} y_u \\ & \text{subject to} && \sum_{u \in e} y_u \geq \tilde{c}_e \quad (e \in E), \\ & && y \in \mathbb{R}_+^V. \end{aligned} \tag{4.10}$$

Here, we show that the dual optimal solution is sparse.

Claim 4.2 *If the optimal value is less than μ , then there exists a dual optimal solution $y \in \mathbb{R}_+^V$ such that $|\text{supp}(y)| < k\mu$.*

Proof Let $x \in \mathbb{R}_+^E$ be a primal optimal solution. We can assume that $x_e > 0$ only if $\tilde{c}_e \geq 1$. Therefore, we have $\sum_e x_e \leq \sum_e \tilde{c}_e x_e < \mu$. Since each hyperedge consists of exactly k elements, we have

$$\sum_{u \in V} \sum_{e \in \delta(u)} x_e = k \sum_{e \in E} x_e < k\mu. \quad (4.11)$$

This shows that less than $k\mu$ inequalities can hold in equality. Therefore, by complementary slackness, the corresponding dual optimal solution y satisfies $|\text{supp}(y)| < k\mu$. \square

Therefore, by Lemma 3.3, there exists an $(\epsilon, \epsilon/2)$ -witness cover for each $\mu \geq 1$ of size at most M^μ , where $M = \exp\left(O\left(k \log \frac{n}{k\mu} + \frac{1}{\epsilon}\right)\right)$. Thus we obtain the following.

Corollary 4.3 *By taking $T = \Omega(\Delta_c(k \log(n/\epsilon) + 1/\epsilon)/\epsilon p)$, for the k -hypergraph matching problem, Algorithm 1 outputs a $(1 - \epsilon)/(k - 1 + 1/k)$ -approximate solution with probability at least $1 - \epsilon$.*

This result is improved by using the vertex sparsification lemma in Sect. 5.

Comparison with Blum et al. Blum et al. [7] provided adaptive and non-adaptive algorithms for the unweighted k -hypergraph matching problem based on local search of Hurkens and Schrijver [23]. Their adaptive algorithm has approximation ratio of $(2 - \epsilon)/k$ in expectation by conducting a constant number of queries per vertex.

For unweighted problem, our algorithm has a worse approximation ratio than theirs. However, our algorithm has four advantages: it requires exponentially smaller number of queries; it runs in polynomial time both in n and $1/\epsilon$; it is applied to the weighted problem with the same approximation ratio; and it has a stronger stochastic guarantee, i.e., not in expectation but with high probability.

Remark 4.1 For unweighted k -hypergraph matching problem, Chan and Liu [9] showed that there is a packing LP with an integrality gap of $2/(k + 1)$. Note that the rounding algorithm for this LP is not known. Using this formulation, we obtain a $(2 - \epsilon)/(k + 1)$ approximation algorithm which conducts $O_{\epsilon,p}(\log^2 n)$ queries and runs in non-polynomial time (i.e., it performs exhaustive search).

4.1.4 k -Column sparse packing integer programming

The k -column sparse packing integer programming problem is a common generalization of the k -hypergraph matching problem and the knapsack problem, and can be represented as follows (the formulation itself just rewrites (1.1) by using the entries of the matrix and of the vectors):

$$\begin{aligned}
 & \text{maximize} && \sum_{j=1}^m \tilde{c}_j x_j \\
 & \text{subject to} && \sum_{j=1}^m a_{ij} x_j \leq b_i \quad (i = 1, \dots, n), \\
 & && x \in \{0, 1\}^m,
 \end{aligned} \tag{4.12}$$

where “ k -column sparse” means that $|\{i : a_{ij} \neq 0\}| \leq k$ for each $j \in \{1, \dots, m\}$. Without loss of generality, we assume that $a_{ij} \leq b_i$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$.

The main difference from the other problems is that the system $\sum_j a_{ij} x_j \leq b_i, x \geq 0$ does not imply $x_j \leq 1$. Instead, we have $x_j \leq w_j$, where $w_j = \min_{i: a_{ij} \neq 0} b_i / a_{ij}$. Let $w = \max_j w_j$. By modifying Algorithm 1 to reveal each x_j with probability x_j / w , we obtain the same approximation guarantee with w times larger number of iterations.

Parekh [30] proposed an LP-relative $(1/2k)$ -approximation algorithm for general k and a $(1/3)$ -approximation algorithm for $k = 2$, which encompasses the *demand matching problem* [35]. The expected number of revealed elements for each constraint i is $O(b_i T)$, because we have

$$\sum_{j: a_{ij} \neq 0} x_j \leq \sum_j a_{ij} x_j \leq b_i. \tag{4.13}$$

The only remaining issue is the number of iterations. Using the same approach as for the k -hypergraph matching problem, we obtain the following result.

Corollary 4.4 *By taking $T = \Omega(\Delta_c w(k \log(n/\epsilon) + 1/\epsilon)/\epsilon p)$, for the k -column sparse packing integer programming problem, Algorithm 1 outputs a $(1 - \epsilon)/2k$ -approximate solution with probability at least $1 - \epsilon$.*

4.2 Matroid problems

Now we apply our technique to matroid-related optimization problems (see, e.g., [34] for basics of matroids and related optimization problems). In this section, unless otherwise noted, m denotes the ground set size of the matroids in question.

4.2.1 Maximum independent set

Let $\mathbf{M} = (E, \mathcal{I})$ be a matroid on a finite set E , and let $r: 2^E \rightarrow \mathbb{Z}_+$ be its rank function. A set $S \subseteq E$ is a *flat* if $r(S \cup e) \neq r(S)$ for all $e \in E \setminus S$, and let $\mathcal{F}_{\mathbf{M}}$ denote the family of flats in \mathbf{M} . For a subset $S \subseteq E$, the smallest flat containing S is called the *closure* of S . We assume that the rank of the matroid is relatively small to ensure that Algorithm 1 does not reveal all the elements.

Let $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic weight. The maximum independent set problem can be represented as follows:

$$\begin{aligned}
& \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\
& \text{subject to} && \sum_{e \in S} x_e \leq r(S) \quad (S \subseteq E), \\
& && x \in \{0, 1\}^E.
\end{aligned} \tag{4.14}$$

Edmonds [18] showed that the LP relaxation of the above system is TDI, so our algorithm has an approximation ratio of $(1 - \epsilon)$ with high probability. Moreover, the number of revealed elements is $O(rT)$, where $r = r(E)$ is the rank of the matroid in question.

The only remaining issue is the number of iterations. Since the system has exponentially many constraints, we have to exploit the combinatorial structure of the problem to reduce the number of possibilities. The dual problem is given by

$$\begin{aligned}
& \text{minimize} && \sum_{S \subseteq E} r(S) y_S \\
& \text{subject to} && \sum_{S \subseteq E: e \in S} y_S \geq \tilde{c}_e \quad (e \in E), \\
& && y \in \mathbb{R}_+^{2^E}.
\end{aligned} \tag{4.15}$$

Since the closure of each $S \subseteq E$ contributes the objective value by $r(S)$ and contains all elements in S , we can restrict the supports of y to the subfamilies of \mathcal{F}_M . Then, for $\mu \geq 1$, let us define a set $W \subseteq \mathbb{R}_+^{2^E}$ by

$$W = \left\{ y \in \mathbb{Z}_+^{2^E} : \sum_{S \in \mathcal{F}_M} r(S) y_S \leq (1 - \epsilon)\mu, \text{ supp}(y) \subseteq \mathcal{F}_M \right\}. \tag{4.16}$$

It is clear that W is an (ϵ, ϵ) -witness cover, and we can evaluate the size of W as follows.

Claim 4.3 $|W| = e^{O(\mu \log m)}$.

Proof To evaluate the size of W , as for the non-bipartite matching problem, we regard y as a multiset of flats. Let r_1, \dots, r_k be the ranks of flats in y . Then we have $r_1 + \dots + r_k < \mu$. Since each flat is the closure of some independent set, the number of flats of rank r is at most the number of independent sets of size r , which is at most $\binom{m}{r}$. Therefore, the number of dual candidates for y is given by

$$\sum_{k=0}^{\lfloor \mu \rfloor} \sum_{\substack{r_1 + \dots + r_k \leq \mu \\ r_1, \dots, r_k \geq 1}} \binom{m}{r_1} \cdots \binom{m}{r_k} \leq e^{O(\mu \log m)}. \tag{4.17}$$

□

Therefore, we obtain the following.

Corollary 4.5 *By taking $T = \Omega(\Delta_c \log(m/\epsilon)/\epsilon p)$, for the maximum independent set problem, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.*

4.2.2 Matroid intersection

The same technique can also be applied to the matroid intersection problem. Let $\mathbf{M}_j = (E, \mathcal{I}_j)$ ($j = 1, 2$) be two matroids whose rank functions are $r_j: 2^E \rightarrow \mathbb{Z}_+$, and $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic weight. The weighted matroid intersection problem can be represented as

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in S} x_e \leq r_j(S) \quad (S \subseteq E, j \in \{1, 2\}), \\ & && x \in \{0, 1\}^E. \end{aligned} \quad (4.18)$$

Edmonds [18] showed that the LP relaxation of the above system is TDI, so our algorithm has an approximation ratio of $(1 - \epsilon)$ with high probability for sufficiently large T . Moreover, the number of revealed elements is $O(r^*T)$, where r^* is the maximum rank of a common independent set in the two matroids.

The only remaining issue is the number of iterations. As the analysis of the maximum independent set problem, we can restrict the supports of dual vectors $y \in \mathbb{R}_+^{2^E} \times \mathbb{R}_+^{2^E}$ to the subfamilies of $\mathcal{F}_{\mathbf{M}_1} \times \mathcal{F}_{\mathbf{M}_2}$, and we obtain the following by the same argument.

Corollary 4.6 *By taking $T = \Omega(\Delta_c \log(m/\epsilon)/\epsilon p)$, for the matroid intersection problem, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.⁶*

4.2.3 k -Matroid intersection

Let $\mathbf{M}_j = (E, \mathcal{I}_j)$ ($j = 1, 2, \dots, k$) be k matroids whose rank functions are $r_j: 2^E \rightarrow \mathbb{Z}_+$, and $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic weight. The k -matroid intersection problem can be represented as

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in S} x_e \leq r_j(S) \quad (S \subseteq E, j \in \{1, 2, \dots, k\}), \\ & && x \in \{0, 1\}^E. \end{aligned} \quad (4.19)$$

⁶ Note that the bipartite matching problem is a special case of the matroid intersection problem, and Corollary 4.1 is obtained from a naive application of this result. Using the vertex sparsification lemma shown in Sect. 5, a stronger result is obtained for bipartite matching (Corollary 5.1).

The important difference between the 2-intersection and k -intersection ($k \geq 3$) problems is that the latter is NP-hard in the non-stochastic case. Moreover, the LP relaxation of the system is not a kind of TDI.

Adamczyk et al. [2] proposed an LP-relative $(1/k)$ -approximation algorithm. The expected number of revealed elements is $O(\hat{r}T)$, where \hat{r} is the minimum of the ranks of the k matroids.

The only remaining issue is the number of iterations. Since the LP relaxation of (4.19) is not TDI, we have to discretize the dual variables. Moreover, since we could not prove the dual optimal solution is sparse, we use Theorem 3.1.

For $\theta = \Theta\left(\frac{\epsilon^2}{\log m}\right)$ and $\mu \geq 1$, let us define a set $W \subseteq \left(\mathbb{R}_+^{2^V}\right)^k$ by

$$W = \left\{ (y^1, \dots, y^k) \in \left(\theta \mathbb{Z}_+^{2^V}\right)^k : \sum_{j,S} r_j(S) y_S^j \leq \left(1 - \frac{\epsilon}{2}\right) \mu, \text{ supp}(y^j) \subseteq \mathcal{F}_{\mathbf{M}_j} \ (\forall j = 1, \dots, k) \right\}. \quad (4.20)$$

By Theorem 3.1, W is an $(\epsilon, \epsilon/2)$ -witness cover, and we can evaluate its size as follows.

Claim 4.4 $|W| = e^{O(\mu k \log^2 m / \epsilon^2)}$.

Proof To evaluate the size of W , as for the maximum independent set problem, we count each y^j separately, where we regard y^j as a multiset in which each flat S contributes θ . Let r_1, \dots, r_k be the ranks of flats in y^j . Then we have $r_1 + \dots + r_k < \mu/\theta$. By the same argument as for the maximum independent set problem, the number of dual candidates for y^j is at most $e^{O((\mu/\theta) \log m)}$. By multiplying the numbers of candidates for the k coordinates, we obtain the required result (recall $\theta = \Theta(\epsilon^2 / \log m)$). \square

Therefore, we obtain the following.

Corollary 4.7 *By taking $T = \Omega(\Delta_c k \log m \log(m/\epsilon)/\epsilon^3 p)$, for the k -matroid intersection problem, Algorithm 1 outputs a $(1 - \epsilon)/k$ -approximate solution with probability at least $1 - \epsilon$.*

4.2.4 Matchoid

The matchoid problem is a common generalization of the matching problem and the matroid intersection problem. Let (V, E) be a graph with $|V| = n$ and $|E| = m$, $\mathbf{M}_v = (\delta(v), \mathcal{I}_v)$ be a matroid whose rank function is $r_v: 2^{\delta(v)} \rightarrow \mathbb{Z}_+$ for each vertex $v \in V$, and $\tilde{c} \in \mathbb{Z}_+^E$ be a stochastic edge weight. The task is to find a maximum-weight subset $F \subseteq E$ of edges such that $F \cap \delta(v) \in \mathcal{I}_v$ for every $v \in V$. A naive LP formulation is as follows:

$$\begin{aligned}
 & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\
 & \text{subject to} && \sum_{e \in S} x_e \leq r_v(S) \quad (v \in V, S \subseteq \delta(v)), \\
 & && x \in \{0, 1\}^E.
 \end{aligned} \tag{4.21}$$

Lee, Sviridenko, and Vondrák [27]⁷ proposed an LP-relative $(2/3)$ -approximation algorithm. For each vertex $v \in V$, the expected number of revealed edges incident to v is $O(r_v T)$, where $r_v = r_v(\delta(v)) \geq \sum_{e \in \delta(v)} x_e$.

The only remaining issue is the number of iterations. Since it has exponentially many constraints (in the maximum degree), we have to exploit its combinatorial structure to reduce the number of possibilities. The dual problem is given by

$$\begin{aligned}
 & \text{minimize} && \sum_{v \in V} \sum_{S \subseteq \delta(v)} r_v(S) y_{v,S} \\
 & \text{subject to} && \sum_{v \in V} \sum_{S \subseteq \delta(v): e \in S} y_{v,S} \geq \tilde{c}_e \quad (e \in E), \\
 & && y \in \mathbb{R}_+^S,
 \end{aligned} \tag{4.22}$$

where $\mathcal{S} = \{(v, S) \mid v \in V, S \subseteq \delta(v)\} \subseteq V \times 2^E$. Similarly to the other matroid problems, we can restrict the support of y so that, if $y_{v,S} > 0$, then $S \subseteq \delta(v)$ is a flat in \mathbf{M}_v . Let \mathcal{F}_v be the set of flats in \mathbf{M}_v and $\mathcal{F} = \{(v, S) \mid v \in V, S \in \mathcal{F}_v\}$. Based on the (TDI/2)-ness of the matroid matching polyhedron due to Gijswijt and Pap [20], the following set is an (ϵ, ϵ) -witness cover:

$$W = \left\{ y \in \frac{1}{2} \mathbb{Z}_+^S : \sum_{v \in V} \sum_{S \subseteq \delta(v)} r_v(S) y_{v,S} \leq (1 - \epsilon) \mu, \text{ supp}(y) \subseteq \mathcal{F} \right\}. \tag{4.23}$$

Similarly to the maximum independent set case, $|W|$ can be bounded by $e^{O(\mu \log m)}$. Thus we obtain the following.

Corollary 4.8 *By taking $T = \Omega(\Delta_c \log(m/\epsilon)/\epsilon p)$, for the matchoid problem, Algorithm 1 outputs a $(2 - \epsilon)/3$ -approximate solution with probability at least $1 - \epsilon$.*

4.2.5 Degree bounded matroid

Let (V, E) be a hypergraph with $|V| = n$ whose maximum degree is $d = \max_{u \in V} |\delta(u)|$, and $b: E \rightarrow \mathbb{Z}_+$ give a capacity of each hyperedge. Let $\mathbf{M} = (V, \mathcal{I})$ be a matroid whose rank function is $r: 2^V \rightarrow \mathbb{Z}_+$, and $\tilde{c} \in \mathbb{Z}_+^V$ be a stochastic weight. The degree bounded matroid problem can be represented as

⁷ Precisely, the discussion is given via a reduction to the matroid matching problem, which preserves the variables and the feasible region.

$$\begin{aligned}
& \text{maximize} && \sum_{u \in V} \tilde{c}_u x_u \\
& \text{subject to} && \sum_{u \in e} x_u \leq b(e) \quad (e \in E), \\
& && \sum_{u \in S} x_u \leq r(S) \quad (S \subseteq V), \\
& && x \in \{0, 1\}^V.
\end{aligned} \tag{4.24}$$

Király et al. [24] proposed an algorithm that finds a (possibly infeasible) solution whose objective value is at least the LP-optimal value and which violates each capacity constraint by at most $d - 1$. Since $\sum_{u \in V} x_u \leq r(V) =: r$, the expected number of revealed elements is $O(rT)$.

The only remaining issue is the number of iterations. Since this system has exponentially many constraints, we have to exploit its combinatorial structure to reduce the number of possibilities. The dual problem is given by

$$\begin{aligned}
& \text{minimize} && \sum_{e \in E} b(e) y_e + \sum_{S \subseteq V} r(S) z_S =: \tau(y, z) \\
& \text{subject to} && \sum_{e \in \delta(u)} y_e + \sum_{S \subseteq V: u \in S} z_S \geq \tilde{c}_u \quad (u \in V), \\
& && y \in \mathbb{R}_+^E, z \in \mathbb{R}_+^{2^V}.
\end{aligned} \tag{4.25}$$

Since the system is not TDI, we have to discretize the dual variables. We could not prove the sparsity of z but, by observing the sparsity of y and using the matroid property, we can see that there exists a good discretization.

Claim 4.5 *Let $(y, z) \in \mathbb{R}_+^E \times \mathbb{R}_+^{2^V}$ be an optimal solution to (4.25) with $\tau(y, z) < \mu$. Then, there exists a feasible solution (y', z') with $\tau(y', z') < (1 - \epsilon/2)\mu$ whose entries are multiple of $\epsilon/2d$.*

Proof Let $x \in \mathbb{R}_+^V$ be a primal LP-optimal solution. By complementary slackness, $y_e > 0$ only if the constraint $\sum_{u \in e} x_u \leq b(e)$ holds in equality. Thus, by summing up, we have

$$\sum_{e: y_e > 0} b(e) = \sum_{e: y_e > 0} \sum_{u \in e} x_u \leq \sum_{e \in E} \sum_{u \in e} x_u \leq d \sum_{u \in V} x_u < d\mu. \tag{4.26}$$

Now we round up each entry of y to the minimum multiple of $\epsilon/2d$ to obtain y' . This increases objective value at most $(\epsilon/2d) \sum_{e: y_e > 0} b(e) < \epsilon\mu/2$. Therefore the objective value of (y', z) is at most $(1 - \epsilon/2)\mu$. To discretize z , we consider the minimization problem with respect to z :

$$\begin{aligned}
& \text{minimize} && \sum_{S \subseteq V} r(S) z_S \\
& \text{subject to} && \sum_{S \subseteq V: u \in S} z_S \geq \tilde{c}_u - \sum_{e \in \delta(u)} y'_e \quad (u \in V).
\end{aligned} \tag{4.27}$$

This problem is the dual of the maximum independent set problem whose cost vector is a multiple of $\epsilon/2d$. Therefore, by the TDIness of the maximum independent set problem, there exists an optimal solution z' whose entries are multiples of $\epsilon/2d$. Thus, (y', z') is feasible by construction and has an objective value of at most $(1 - \epsilon/2)\mu$. \square

As for the other matroid problems, we can assume that $\text{supp}(z)$ is a set of flats. For $\mu \geq 1$, let us define a set $W \subseteq \mathbb{R}_+^E \times \mathbb{R}_+^{2V}$ by

$$W = \left\{ (y, z) \in \frac{\epsilon}{2d} \left(\mathbb{Z}_+^E \times \mathbb{Z}_+^{2V} \right) : \tau(y, z) \leq \left(1 - \frac{\epsilon}{2} \right) \mu, \text{supp}(z) \subseteq \mathcal{F}_M \right\}. \quad (4.28)$$

By construction, W is an $(\epsilon, \epsilon/2)$ -witness cover, and we can evaluate its size as follows.

Claim 4.6 $|W| = e^{O(\mu d \log n / \epsilon)}$.

Proof To evaluate the size of W , we separately count y and z . The number of candidates for y is evaluated as similar to Lemma 3.1 by distributing $2d\mu/\epsilon$ tokens to $|E| = O(dn) = O(n^2)$ components, and is bounded by $e^{O(\mu d \log n / \epsilon)}$. The number of candidates for z is evaluated as similar to Claim 4.3 by counting multisets with weight $\epsilon/2d$, and is bounded by $e^{O(\mu d \log n / \epsilon)}$. By multiplying these two numbers of candidates, we obtain the required result. \square

Therefore, we obtain the following.

Corollary 4.9 *For the degree bounded matroid problem with maximum degree d , by taking $T = \Omega(\Delta_c d \log(n/\epsilon)/\epsilon^2 p)$, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution that violates each constraint at most $d - 1$ with probability at least $1 - \epsilon$.*

4.3 Stable set problems

We finally show applications to stable set problems. In this section, n and m denote the numbers of vertices and edges, respectively, in the graph in question.

4.3.1 Stable set in some perfect graphs

We assume that the stability number α (the maximum size of a stable set) is relatively small to ensure that Algorithm 1 does not reveal all the vertices. By the Turán theorem [36], the average degree is required to be relatively large.

Let (V, E) be a graph and $\tilde{c}: V \rightarrow \mathbb{Z}_+$ be a stochastic vertex weight. The maximum stable set problem can be represented as

$$\begin{aligned} & \text{maximize} && \sum_{u \in V} \tilde{c}_u x_u \\ & \text{subject to} && x_u + x_v \leq 1 \quad ((u, v) \in E), \\ & && x \in \{0, 1\}^V. \end{aligned} \quad (4.29)$$

The LP relaxation of this system is half-integral. However, this is not helpful because the number of revealed vertices can be large: there is a solution $x_u = 1/2$ for all $u \in V$, which corresponds to revealing half of the vertices in expectation.

We instead consider the following formulation, which introduces the *clique inequalities*:

$$\begin{aligned} & \text{maximize} && \sum_{u \in V} c_u x_u \\ & \text{subject to} && \sum_{u \in C} x_u \leq 1 \quad (C \in \mathcal{C}), \\ & && x_u \in \{0, 1\}^V, \end{aligned} \quad (4.30)$$

where \mathcal{C} is the set of maximal cliques. A graph is *perfect* if the LP relaxation of the above system is TDI. If we assume that the graph is perfect, Algorithm 1 has an approximation ratio of $(1 - \epsilon)$ with high probability for sufficiently large T , and the number of revealed vertices is $O(\alpha T)$ in expectation.

The dual problem is given by

$$\begin{aligned} & \text{minimize} && \sum_{C \in \mathcal{C}} y_C \\ & \text{subject to} && \sum_{C \in \mathcal{C}: u \in C} y_C \geq c_u \quad (u \in V), \\ & && y \in \mathbb{R}_+^{\mathcal{C}}. \end{aligned} \quad (4.31)$$

If the number of maximal cliques is $O(n^k)$ for some fixed constant k , we immediately see that the required number of iterations is $O(k \log(n/\epsilon)/\epsilon p)$. A perfect graph may have exponentially many maximal cliques in general, but the following graph classes have at most polynomially many maximal cliques.

- If a graph is chordal, it has only linear number of maximal cliques.
- If a graph has a bounded clique number (i.e., the size of cliques are bounded by a constant k), the number of cliques is at most $\binom{n}{k} = O(n^k)$. This includes a graph class that can be characterized by forbidden minors and subgraphs.

4.3.2 Stable set in t-perfect graphs

Another tractable graph class for the stable set problem is t-perfect graphs. A graph (V, E) is *t-perfect* if the relaxation of the following formulation is integral, i.e., it has an integral optimal solution:

$$\begin{aligned} & \text{maximize} && \sum_{u \in V} \tilde{c}_u x_u \\ & \text{subject to} && x_u + x_v \leq 1 \quad ((u, v) \in E), \\ & && \sum_{u \in C} x_u \leq \left\lfloor \frac{|C|}{2} \right\rfloor \quad (C \in \mathcal{C}), \\ & && x_u \in \{0, 1\}^V, \end{aligned} \quad (4.32)$$

where \mathcal{C} is the set of odd cycles. We assume that the graph is t -perfect. Then, Algorithm 1 has an approximation ratio of $(1 - \epsilon)$ with high probability for sufficiently large T , and the number of revealed vertices is $O(\alpha T)$.

The only remaining issue is the number of iterations. Since the system (4.32) is not required to be TDI,⁸ we have to discretize the dual variables. We use Theorem 3.1. Let $\theta = \Theta\left(\frac{\epsilon^2}{\log n}\right)$. The corresponding dual problem is given by

$$\begin{aligned} & \text{minimize} \quad \sum_{e \in E} y_e + \sum_{C \in \mathcal{C}} \left\lfloor \frac{|C|}{2} \right\rfloor \tilde{z}_C \\ & \text{subject to} \quad \sum_{e \in \delta(u)} y_e + \sum_{C \in \mathcal{C}: u \in C} z_C \geq \tilde{c}_u \quad (u \in V), \\ & \quad y \in \mathbb{R}_+^E, \quad z \in \mathbb{R}_+^{\mathcal{C}}. \end{aligned} \quad (4.33)$$

We regard z_C as a multiset in which each odd cycle C contributes θ . Let c_1, \dots, c_k be the sizes of each odd cycles. We then have $c_1 + \dots + c_k < \mu/\theta$. We define the witness cover by

$$W = \left\{ (y, z) \in \theta \left(\mathbb{Z}_+^E \times \mathbb{Z}_+^{\mathcal{C}} \right) : \sum_{e \in E} y_e + \sum_{C \in \mathcal{C}} \left\lfloor \frac{|C|}{2} \right\rfloor z_C \leq \left(1 - \frac{\epsilon}{2}\right) \mu \right\}. \quad (4.34)$$

Claim 4.7 $|W| \leq e^{O((\mu/\theta) \log n)}$.

Proof To evaluate the size of W , we count y and z separately. The number of candidates for y is clearly $e^{O((\mu/\theta) \log m)}$ (and $m = O(n^2)$), while the number of candidates for z is bounded by $e^{O((\mu/\theta) \log n)}$ as the similar argument to Claim 4.1. \square

Therefore, we obtain the following.

Corollary 4.10 *By taking $T = \Omega(\Delta_c \log n \log(n/\epsilon)/\epsilon^3 p)$, for the t -stable set problem, Algorithm 1 outputs a $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.*

5 Vertex sparsification lemma

5.1 Vertex sparsification lemma

For the (unweighted) stochastic matching problem, Assadi et al. [3] proposed a procedure called *vertex sparsification*, which reduces the number of vertices proportional to the maximum matching size μ while approximately preserving any matchings of size $v = \omega(1)$ with high probability. This procedure is very useful as a preprocessing step

⁸ A graph is *strongly t -perfect* if the system in (4.32) is TDI. Any strongly t -perfect graph is t -perfect, but the converse is open.

Algorithm 3 Vertex sparsification.

-
- 1: Assign a random color in $\{1, \dots, \frac{\beta(k, \epsilon, \delta)k^2s}{2\delta}\}$ to each vertex, where $\beta(k, \epsilon, \delta) = \frac{2e^{\epsilon/k} \log(1/\delta)}{\epsilon}$.
 - 2: Return all colorful hyperedges.
-

for this problem since it makes $n/\mu = O(1)$, and so the required number of iterations becomes constant.

Here, we extend this procedure to an independence system on a k -uniform hypergraph and improve the result to preserve *any* independence set with high probability without assuming $v = \omega(1)$. In next section, we improve the performances of the algorithms for the bipartite matching problem, k -hypergraph matching problem, and k -column sparse packing integer programming problem by using this lemma.

In general, sparsification procedures are kinds of *kernelization* procedure, which is studied in the area of parametrized complexity [17]. In particular, our and Assadi et al. [3]'s procedures are similar to the one in [11], which aims to reduce space complexity of packing problems in streaming setting, but the conducted analyses and the provided guarantees are both different.

Let (V, E) be a k -uniform hypergraph and (E, \mathcal{I}) be an independence system (which is a nonempty, downward-closed set system, i.e., $\mathcal{I} \neq \emptyset$, and $X \subseteq Y \in \mathcal{I} \implies X \in \mathcal{I}$), whose rank function $r: 2^E \rightarrow \mathbb{Z}_+$ is defined by $r(S) = \max\{|I| : I \subseteq S, I \in \mathcal{I}\}$. We focus on the following special case of the stochastic packing integer programming problem (1.1) in this section:

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} \tilde{c}_e x_e \\ & \text{subject to} && \sum_{e \in S} x_e \leq r(S) \quad (S \subseteq E), \\ & && x \in \{0, 1\}^E. \end{aligned} \tag{5.1}$$

Note that the constraint is equivalent to $\text{supp}(x) \in \mathcal{I}$, and this formulation still includes the k -column sparse PIP (4.12) (and hence all the matching problems shown in Sect. 4.1) as follows: let $V = \{1, \dots, n\}$ and $E = \{1, \dots, m\}$ such that each hyperedge $j \in E$ is associated with a subset $\{i \in V : a_{ij} \neq 0\}$ (if the size is less than k , add arbitrary vertices i with $a_{ij} = 0$), and define $\mathcal{I} = \{S \subseteq E : \sum_{j \in S} a_{ij} \leq b_i \ (\forall i \in V)\}$.

Our procedure is shown in Algorithm 3, which is a kind of color coding. Let $s \in \mathbb{Z}_+$ be an upper bound on $r = r(E)$, and $\epsilon, \delta \in (0, 1)$ be parameters for the accuracy and the probability, respectively. It first assigns a random color in $\{1, \dots, n^\circ\}$ to each vertex, where $n^\circ = \beta(k, \epsilon, \delta)k^2s/\delta$ with $\beta(k, \epsilon, \delta) = 2e^{\epsilon/k} \log(1/\delta)/\epsilon$. It then returns all “colorful” hyperedges that consists entirely of differently colored vertices. This yields an independence system on the color class consisting from $n^\circ = \Theta(s)$ vertices.

Lemma 5.1 (Vertex Sparsification Lemma) *Suppose that $n \geq 2k$. Then, after Algorithm 3, for any independent set $I \in \mathcal{I}$ in the original instance, there exists an independent set $I^\circ \subseteq I$ of size at least $(1 - \epsilon)|I|$ in the sparsified instance with probability at least $1 - \delta$.*

Proof Let $v = |I|$. For notational simplicity, we denote by $\beta = \beta(k, \epsilon, \delta)$. We now make the following case analysis.

Case 1: $v \leq \beta$ (the rank of I is small). If all vertices incident to I have different colors, the size of I is preserved after the mapping. Since the number of the incident vertices is at most kv , the probability that this has occurred is at least

$$\frac{n^\circ(n^\circ - 1) \cdots (n^\circ - kv + 1)}{n^{okv}} \geq \exp\left(-\frac{k^2v^2}{n^\circ}\right) \geq \exp\left(-\frac{\delta v^2}{\beta s}\right) \geq e^{-\delta} \geq 1 - \delta. \quad (5.2)$$

Here, the first inequality follows from the falling factorial approximation (the next lemma), and the second inequality follows from $v \leq r \leq s$ and $v \leq \beta$.

Lemma 5.2 (Falling Factorial Approximation)

$$\frac{n(n-1) \cdots (n-k+1)}{n^k} \geq \exp\left(-\frac{k^2}{n}\right). \quad (5.3)$$

Proof Recall that $\log(1-x) \geq -x/(1-x)$ for all $x \in (0, 1)$. The logarithm of the above is

$$\sum_{i=1}^{k-1} \log\left(1 - \frac{i}{n}\right) \geq -\sum_{i=1}^{k-1} \frac{i}{n-i} \geq -\sum_{i=1}^{k-1} \frac{i}{n-k} = -\frac{k(k-1)}{2(n-k)} \geq -\frac{k^2}{n}. \quad (5.4)$$

□

Case 2: $v \geq \beta$ (the rank of I is large). We further reduce the number of colors by mapping each color class to $\{1, \dots, k^2v/\epsilon\}$. (Note that $k^2v/\epsilon \leq n^\circ$ since $\beta \geq 1/\epsilon$.) We say that a color class c is *good* if some vertex in color c is covered by some hyperedge $e \in I$, and otherwise we say that c is *bad*.

For each color class c , let X_c be the indicator of the event that c is bad, i.e., $X_c = 1$ if c is bad and $X_c = 0$ otherwise. Then $\Pr(X_c = 1) = (1 - \epsilon/k^2v)^{kv} \leq e^{-\epsilon/k}$. Therefore $\mathbb{E}[\sum_c X_c] \leq e^{-\epsilon/k} k^2v/\epsilon$. Since X_c are negatively correlated random variables, we can apply the Chernoff bound [29]:

$$\begin{aligned} \Pr\left(\sum_c X_c \geq \frac{k^2v}{\epsilon} - \left(1 - \frac{\epsilon}{k}\right)kv\right) &= \Pr\left(\sum_c X_c \geq \left(1 - \frac{\epsilon}{k} + \frac{\epsilon^2}{k^2}\right) \frac{k^2v}{\epsilon}\right) \\ &\leq \Pr\left(\sum_c X_c \geq \left(1 + \frac{\epsilon^2}{2k^2}\right) e^{-\epsilon/k} \frac{k^2v}{\epsilon}\right) \\ &\leq \exp\left(-\epsilon e^{-\epsilon/k} \frac{v}{2}\right) \\ &\leq \exp\left(-\epsilon e^{-\epsilon/k} \frac{\beta}{2}\right) = \delta, \end{aligned} \quad (5.5)$$

where the first inequality follows from $(1 + x^2/2)e^{-x} \leq 1 - x + x^2$ and the last equality follows from the definition of β . Therefore, there are at least $(1 - \epsilon/k)k\nu$ good color classes with high probability.

For each good color class, we select one covered vertex and remove all other vertices. The number of removed vertices is at most $\epsilon\nu$, so at most $\epsilon\nu$ hyperedges in the independent set are removed. The remaining hyperedges form an independent set of size at least $(1 - \epsilon)\nu$. \square

Remark 5.1 The second part is a simple extension of Assadi et al. [3]. Since they only analyzed this case, $\nu = \omega(1)$ was required.

5.2 Usage of vertex sparsification Lemma

Here, we describe how to use the vertex sparsification lemma to improve the performance of Algorithms 1 and 2. For simplicity, we only describe the result for Algorithm 1, as Algorithm 2 can be handled using the same argument.

Let (V, E) be a k -uniform hypergraph with $|V| = n$ and $|E| = m$ and (E, \mathcal{I}) be an independence system. We consider the problem (5.1), where we assume the following.

1. There exists an LP-relative α -approximation algorithm.
2. The number of iterations required to guarantee $(1 - \epsilon)\alpha$ -approximation with probability at least $1 - \delta$ is bounded by $T(\log(n/\mu), \epsilon, \delta)$.

The method is shown in Algorithm 4, where $\epsilon, \delta \in (0, 1)$ are parameters for the accuracy and the probability, respectively, and $c_{\max} = \max_j c_j^+$. We first estimate the maximum size s of the independent sets such that $\alpha s \leq r \leq s$, which is computed via LP relaxation. We then apply Algorithm 3 to obtain a sparsified instance, and finally apply Algorithm 1 or 2 with an LP-relative α -approximation algorithm to obtain a solution. We now analyze the performance of this procedure.

Theorem 5.1 *Algorithm 4 finds a $(1 - \epsilon)\alpha$ -approximate solution with probability at least $1 - \delta$.*

Proof Let $r = r(E)$ be the rank of the original independence system and $\tilde{\mu}$ the optimal value of the original instance (5.1), which is a random variable determined by nature. Also, let r° be the rank of the sparsified instance, and $\tilde{\mu}^\circ$ be the optimal value of the sparsified instance, which is also a random variable.

Claim 5.1

$$\Pr(r^\circ \geq (1 - \epsilon')r) \geq 1 - \delta', \quad (5.6)$$

$$\Pr(\tilde{\mu}^\circ \geq (1 - c_{\max}\epsilon')\tilde{\mu}) \geq 1 - \delta'. \quad (5.7)$$

Algorithm 4 Speedup by vertex sparsification.

- 1: Estimate the size s of maximum independent set such that $\alpha s \leq r \leq s$.
 - 2: Sparsify the instance by Algorithm 3 with accuracy parameter $\epsilon' = \epsilon/(1 + c_{\max})$ and probability parameter $\delta' = \delta/4$.
 - 3: Run Algorithm 1 or 2 with an LP-relative α -approximation algorithm by setting $T = T(O(\log(k/p\alpha\epsilon'\delta')), \epsilon', \delta')$.
-

Proof The first inequality (5.6) immediately follows from Lemma 5.1, and we focus on the second (5.7). Fix a realization of \tilde{c} , and let $x \in \{0, 1\}^E$ be an optimal solution to (5.1) such that $I = \text{supp}(x) \in \mathcal{I}$ is minimal. By Lemma 5.1, there exists an independent set $I^\circ \subseteq I$ of size $|I^\circ| \geq (1 - \epsilon')|I|$ in the sparsified instance with probability $1 - \delta'$. Let $x^\circ \in \{0, 1\}^E$ be the vector with $\text{supp}(x^\circ) = I^\circ$, whose restriction to the sparsified hyperedge set is a feasible solution to the sparsified instance. We then have

$$\tilde{\mu}^\circ \geq \tilde{c}^\top x^\circ \geq \tilde{c}^\top x - c_{\max} \epsilon' |I| \geq \tilde{\mu} - c_{\max} \epsilon' \tilde{\mu}, \quad (5.8)$$

where the last inequality follows from the minimality of $I = \text{supp}(x)$ (for each $j \in I$, we must have $\tilde{c}_j \geq 1$, and hence $\tilde{\mu} = \tilde{c}^\top x \geq \text{supp}(x) = |I|$). \square

By Claim 5.1, we have $r^\circ \geq (1 - \epsilon')r$ and $\tilde{\mu}^\circ \geq (1 - c_{\max} \epsilon') \tilde{\mu}$ with probability at least $1 - 2\delta'$. Under this event, by using Algorithm 1 in Line 3 of Algorithm 4, we obtain a solution whose objective value is at least $(1 - \epsilon') \tilde{\mu}^\circ \geq (1 - (c_{\max} + 1)\epsilon') \tilde{\mu} = (1 - \epsilon) \tilde{\mu}$ with probability at least $1 - \delta'$ (and hence with probability at least $1 - 3\delta'$ in total).

The remaining issue is the number of iterations. That is, for $\beta' = \beta(k, \epsilon', \delta') = 2e^{\epsilon'/k} \log(1/\delta')/\epsilon'$ and $n^\circ = \beta' k^2 s / \delta'$, we prove

$$\log \frac{n^\circ}{\tilde{\mu}^\circ} = O\left(\log \frac{k}{p\alpha\epsilon'\delta'}\right), \quad (5.9)$$

with probability at least $1 - \delta'$, which implies that we succeed with probability at least $1 - 4\delta' = 1 - \delta$ through Algorithm 4. We make a case analysis.

Case 1: $r^\circ \geq 8 \log(1/\delta')/p$ (the rank of the sparsified instance is large). We evaluate the objective value of the independent set in the sparsified instance that corresponds to the maximum independent set in the original instance. Since each element in the sparsified independent set contributes at least 1 with probability at least p , we can apply the Chernoff bound

$$\Pr\left(\tilde{\mu}^\circ \geq \frac{pr^\circ}{2}\right) \geq \Pr\left(\sum_{i=1}^{r^\circ} X_i \geq \frac{pr^\circ}{2}\right) \geq 1 - e^{-pr^\circ/8} \geq 1 - \delta', \quad (5.10)$$

where X_i ($i = 1, \dots, r^\circ$) are i.i.d. random variables following the Bernoulli distribution with probability p . Under this event ($\tilde{\mu}^\circ \geq pr^\circ/2$), we have

$$\frac{n^\circ}{\tilde{\mu}^\circ} \leq \frac{2n^\circ}{pr^\circ} \leq \frac{4n^\circ}{pr} \leq \frac{4n^\circ}{p\alpha s} = \frac{4\beta' k^2}{p\alpha\delta'} = \frac{8e^{\epsilon'/k} \log(1/\delta')}{p\alpha\epsilon'\delta'}, \quad (5.11)$$

where the second inequality follows from $r^\circ \geq (1 - \epsilon')r \geq r/2$ (because $\epsilon' = \epsilon/(1 + c_{\max}) \leq 1/2$), and the third from $r \geq \alpha s$. Since $e^{\epsilon'/k} = O(1)$ and $\log(1/\delta') \leq 1/\delta'$, we have (5.9).

Case 2: $r^\circ \leq 8 \log(1/\delta')/p$ (the rank of the sparsified instance is small). We have

$$\frac{n^\circ}{\tilde{\mu}^\circ} \leq n^\circ \leq \frac{\beta' k^2 r}{\alpha \delta'} \leq \frac{2\beta' k^2 r^\circ}{\alpha \delta'} \leq \frac{16\beta' k^2 \log(1/\delta')}{p\alpha \delta'}, \quad (5.12)$$

where the first inequality follows from $\tilde{\mu}^\circ \geq 1$ (because it is an integer with $\tilde{\mu}^\circ \geq (1 - \epsilon' c_{\max})\tilde{\mu} > 0$), the second from $r \geq \alpha s$, and the third from $r^\circ \geq r/2$. This leads to (5.9) similarly in Case 1. \square

The sizes of witness covers of bipartite matching, k -hypergraph matching, and k -column-sparse packing integer programming depend on n/μ . Thus these are improved by using this technique.

Corollary 5.1 *For the bipartite matching problem with $c_j = O(1)$ for all j , there is an algorithm that conducts $O(\log(1/\epsilon p)/\epsilon p)$ queries per vertex and finds $(1 - \epsilon)$ -approximate solution with probability at least $1 - \epsilon$.*

Corollary 5.2 *For the k -hypergraph matching problem with $c_j = O(1)$ for all j , there is an algorithm that conducts $O(k(\log(k/\epsilon p) + 1/\epsilon)/\epsilon p)$ queries per vertex and finds $(1 - \epsilon)/(k - 1 + 1/k)$ -approximate solution with probability at least $1 - \epsilon$.*

Corollary 5.3 *For the k -column sparse packing integer programming problem with $c_j = O(1)$, $b_i = O(1)$, and $A_{ij} = O(1)$ for all i, j , there is an algorithm that conducts $O(k(\log(k/\epsilon p) + 1/\epsilon)/\epsilon p)$ queries per vertex and finds $(1 - \epsilon)/2k$ -approximate solution with probability at least $1 - \epsilon$.*

Acknowledgements The authors thank anonymous reviewers for their careful reading and a number of valuable comments. This work was supported by JSPS KAKENHI Grant Numbers 16H06931 and 16K16011.

References

- Adamczyk, M.: Improved analysis of the greedy algorithm for stochastic matching. *Inf. Process. Lett.* **111**(15), 731–737 (2011)
- Adamczyk, M., Sviridenko, M., Ward, J.: Submodular stochastic probing on matroids. *Math. Oper. Res.* **41**(3), 1022–1038 (2016)
- Assadi, S., Khanna, S., Li, Y.: The stochastic matching problem with (very) few queries. In: *Proceedings of the 17th ACM Conference on Economics and Computation*, pp. 43–60. ACM (2016)
- Assadi, S., Khanna, S., Li, Y.: The stochastic matching problem: beating half with a non-adaptive algorithm. In: *Proceedings of the 18th ACM Conference on Economics and Computation*, pp. 99–116. ACM (2017)
- Bansal, N., Gupta, A., Li, J., Mestre, J., Nagarajan, V., Rudra, A.: When LP is the cure for your matching woes: improved bounds for stochastic matchings. *Algorithmica* **63**(4), 733–762 (2012)
- Behnezhad, S., Reyhani, N.: Almost optimal stochastic weighted matching with few queries In: *Proceedings of the 19th ACM Conference on Economics and Computation*, pp. 235–249. ACM (2018)
- Blum, A., Dickerson, J.P., Haghtalab, N., Procaccia, A.D., Sandholm, T., Sharma, A.: Ignorance is almost bliss: near-optimal stochastic matching with few queries. In: *Proceedings of the 16th ACM Conference on Economics and Computation*, pp. 325–342. ACM (2015)
- Blum, A., Gupta, A., Procaccia, A., Sharma, A.: Harnessing the power of two crossmatches. In: *Proceedings of the 14th ACM conference on Electronic Commerce*, pp. 123–140. ACM (2013)
- Chan, Y.H., Lau, L.C.: On linear and semidefinite programming relaxations for hypergraph matching. In: *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1500–1511. SIAM (2010)

10. Chen, N., Immorlica, N., Karlin, A.R., Mahdian, M., Rudra, A.: Approximating matches made in heaven. In: *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming*, pp. 266–278. Springer (2009)
11. Chitnis, R., Cormode, G., Esfandiari, H., Hajiaghayi, M., McGregor, A., Monemizadeh, M., Vorotnikova, S.: Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In: *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1326–1344. SIAM (2016)
12. Costello, K., Tetali, P., Tripathi, P.: Stochastic matching with commitment. In: *Proceedings of the 39th International Colloquium on Automata, Languages and Programming*, pp. 822–833. Springer (2012)
13. Cunningham, W.H., Marsh, A.B.: A primal algorithm for optimum matching. *Polyhedral Combinatorics* pp. 50–72 (1978)
14. Dean, B.C., Goemans, M.X., Vondrák, J.: Approximating the stochastic knapsack problem: the benefit of adaptivity. In: *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 208–217. IEEE (2004)
15. Dean, B.C., Goemans, M.X., Vondrák, J.: Adaptivity and approximation for stochastic packing problems. In: *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 395–404. SIAM (2005)
16. Dickerson, J.P., Sandholm, T.: Organ exchanges: a success story of AI in healthcare. In: *30th Conference on Artificial Intelligence Tutorial Forum* (2016)
17. Downey, R.G., Fellows, M.R.: *Parameterized Complexity*. Springer, Berlin (2012)
18. Edmonds, J.: Submodular functions, matroids, and certain polyhedra. In: *Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications*, pp. 69–87. Gordon and Breach (1970)
19. Egerváry, E.: On combinatorial properties of matrices. *Matematikai Lapok* **38**, 16–28 (1931)
20. Gijswijt, D., Pap, G.: An algorithm for weighted fractional matroid matching. *J. Comb. Theory, Ser. B* **103**(4), 509–520 (2013)
21. Gupta, A., Nagarajan, V.: A stochastic probing problem with applications. In: *Proceedings of the 16th International Conference on Integer Programming and Combinatorial Optimization*, pp. 205–216. Springer (2013)
22. Gupta, A., Nagarajan, V., Singla, S.: Adaptivity gaps for stochastic probing: submodular and XOS functions. In: *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1688–1702. SIAM (2017)
23. Hurkens, C.A.J., Schrijver, A.: On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discret. Math.* **2**(1), 68–72 (1989)
24. Király, T., Lau, L.C., Singh, M.: Degree bounded matroids and submodular flows. *Combinatorica* **32**(6), 703–720 (2012)
25. Kollipoulos, S.G., Young, N.E.: Approximation algorithms for covering/packing integer programs. *J. Comput. Syst. Sci.* **71**(4), 495–505 (2005)
26. König, D.: Graphs and matrices. *Math. Fiz. Lapok* **38**, 116–119 (1931)
27. Lee, J., Sviridenko, M., Vondrák, J.: Matroid matching: the power of local search. *SIAM J. Comput.* **42**(1), 357–379 (2013)
28. Molinaro, M., Ravi, R.: The query-commit problem. *arXiv preprint [arXiv:1110.0990](https://arxiv.org/abs/1110.0990)* (2011)
29. Panconesi, A., Srinivasan, A.: Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM J. Comput.* **26**(2), 350–368 (1997)
30. Parekh, O.: Iterative packing for demand and hypergraph matching. In: *Proceedings of the 15th International Conference on Integer Programming and Combinatorial Optimization*, pp. 349–361. Springer (2011)
31. Parekh, O., Pritchard, D.: Generalized hypergraph matching via iterated packing and local ratio. In: *Proceedings of the 12th International Workshop on Approximation and Online Algorithms*, pp. 207–223. Springer (2014)
32. Raghavan, P., Thompson, C.D.: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* **7**(4), 365–374 (1987)
33. Roth, A.E., Sönmez, T., Ünver, M.U.: Kidney exchange. *Q. J. Econ.* **119**(2), 457–488 (2004)
34. Schrijver, A.: *Combinatorial Optimization: Polyhedra and Efficiency*, Springer (2003)

35. Shepherd, F.B., Vetta, A.: The demand-matching problem. *Math. Oper. Res.* **32**(3), 563–578 (2007)
36. Turán, P.: On the theory of graphs. *Colloquium Mathematicae* **3**(1), 19–30 (1954)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.