Stochastic Packing Integer Programs with Few Queries

Takanori Maehara\textsuperscript{11} and Yutaro Yamaguchi\textsuperscript{12,1}

\textsuperscript{1}RIKEN Center for Advanced Intelligence Project
\textsuperscript{2}Osaka University

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.
## Contents

1 Introduction
   1.1 Problem Formulation ........................................ 1
   1.2 Our Contributions and Technique ....................... 1
   1.3 Related Work ............................................. 3
   1.4 Organization ............................................. 4

2 General Framework
   2.1 Two Strategies ........................................... 5
   2.2 Performance Analysis .................................... 6
   2.3 Proofs of Main Theorems ................................ 8

3 Constructing Witness Covers
   3.1 Totally Dual Integral Case .............................. 11
   3.2 Non-TDI Case ............................................. 11
   3.3 Exponentially Many Constraints ....................... 13

4 Applications
   4.1 Matching Problems ....................................... 13
      4.1.1 Bipartite Matching .................................. 14
      4.1.2 Non-bipartite Matching ............................ 14
      4.1.3 $k$-Hypergraph Matching .......................... 16
      4.1.4 $k$-Column Sparse Packing Integer Programming ... 18
   4.2 Matroid Problems ........................................ 19
      4.2.1 Maximum Independent Set .......................... 19
      4.2.2 Matroid Intersection ............................... 20
      4.2.3 $k$-Matroid Intersection ......................... 21
      4.2.4 Matroid Matching .................................. 22
      4.2.5 Degree Bounded Matroid .......................... 23
   4.3 Stable Set Problems ..................................... 24
      4.3.1 Stable Set in Some Perfect Graphs .............. 25
      4.3.2 Stable Set in $t$-Perfect Graphs ............... 26

5 Vertex Sparsification Lemma
   5.1 Vertex Sparsification Lemma ........................... 27
   5.2 Usage of Vertex Sparsification Lemma ............... 29
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 unified manner. Specifically, we introduce the stochastic packing integer programming problem defined as follows:

\[
\begin{align*}
\text{maximize} & \quad \tilde{c}^\top x \\
\text{subject to} & \quad Ax \leq b, \\
& \quad x \in \{0, 1\}^m,
\end{align*}
\]

(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 \(\tilde{c}_j (j = 1, 2, \ldots, m)\) 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 \(\{c^-_j, c^-_j + 1, \ldots, c^+_j\}\) 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 \((A, b, \text{and the above information on } \tilde{c})\) 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.

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

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.
Table 1.1: 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, respectively, in question. We omit $O(\cdot)$ in the iteration column. Also, all the coefficients are assumed to be $O(1)$.

For the non-adaptive algorithm, the approximation ratio is halved.

| Problem                              | Approximation Ratio | Number of Iterations $T$ |
|--------------------------------------|---------------------|--------------------------|
| Bipartite Matching                   | $1 - \epsilon$     | $\log(1/e\rho)/e\rho$   |
| Non-bipartite Matching               | $1 - \epsilon$     | $\log(n/e)/e\rho$       |
| $k$-Hypergraph Matching              | $(1 - \epsilon)/(k - 1 + 1/k)$ | $(k\log(k/e) + 1/\epsilon)/e\rho$ |
| $k$-Column Sparse PIP                | $(1 - \epsilon)/2k$ | $(k\log(k/e) + 1/\epsilon)/e\rho$ |
| Matroid (Max. Independent Set)       | $1 - \epsilon$     | $\log(m/e)/e\rho$       |
| Matroid Intersection                 | $1 - \epsilon$     | $\log(m/e)/e\rho$       |
| $k$-Matroid Intersection             | $(1 - \epsilon)/k$ | $k\log m\log(m/e)/\epsilon^3\rho$ |
| Matroid Matching                     | $(1 - \epsilon)2/3$| $\log(n/e)/e\rho$       |
| Degree Bounded Matroid               | $1 - \epsilon$     | $d\log(n/e)/\epsilon^3\rho$ |
| Stable Set in Chordal Graphs         | $1 - \epsilon$     | $\log n/e\rho$          |
| Stable Set in $t$-Perfect Graphs     | $1 - \epsilon$     | $\log n\log(n/e)/\epsilon^3\rho$ |

In the adaptive strategy\(^1\) (which is formally shown in Algorithm\(^1\) in Section 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 computes an optimal fractional solution $x$ to the optimistic LP, and rounds down each element $j$ (i.e., supposes $\tilde{c}_j$ 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.

\(^1\)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.
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 $\alpha$-approximation algorithm \(^3\) (which outputs an integral feasible solution whose objective value is at least $\alpha$ 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 $T$ iterations has no feasible solution whose objective value 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 $\mu$ (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 fruitful properties, e.g., when the constraint system $Ax \leq b, x \geq 0$ is totally dual integral.

1.3 Related Work

Our stochastic packing integer programming problem generalizes the stochastic (unweighted) matching problem \(^4\) and the stochastic (unweighted) $k$-hypergraph matching problem \(^5\), which have recently been studied in EC (Economics and Computation) community. These problems are motivated to find an optimal strategy for kidney exchange \(^6\).

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 $O(\log(1/\epsilon)/\epsilon^{2/\epsilon})$ queries per vertex. Their technique is based on the existence of disjoint short augmenting paths. Assadi et al. \(^8\) proposed adaptive and non-adaptive algorithms that respectively achieve the same approximation ratios\(^2\) 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.
with high probability, by conducting \( O(\log(1/\epsilon)/ep) \) 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 \( O(\log(n/\epsilon)/ep) \) queries per vertex are required\(^3\) which is worse than theirs.

Recently, Assadi et al. \cite{assadi2020non} 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. \cite{blum2021adaptive} proposed adaptive and non-adaptive algorithms that find \( (2-\epsilon)/k \)- and \( (4-\epsilon)/(k^2+2k) \)-approximate matchings, respectively, in expectation, by conducting \( O(s_{k,\epsilon} \log(1/\epsilon)/p^{k,\epsilon}) \) queries per vertex, where \( s_{k,\epsilon} \) is a constant depending on \( k \) and \( \epsilon \). Their technique is based on the local search method of Hurkens and Schrijver \cite{hurkens1995local}. For the adaptive case, our strategy just 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 \cite{behnezhad2020query,ren2020query,ren2020query,blum2021stochastic}, in which the queried elements must be a part of the output. Some studies \cite{ren2020query,behnezhad2020query,ren2020query} also impose additional budget constraints on the number of queries. In the stochastic probing problem \cite{blum2021stochastic,behnezhad2020query,ren2020query}, both the queried and realized elements must satisfy given constraints. Blum et al. \cite{blum2021stochastic} 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 Section 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 Section 3, we outline how to construct a small witness cover in general. In Section 4, we apply the technique to a variety of specific combinatorial problems. In Section 5, we provide a vertex sparsification lemma that can be used to improve the performance of the

\(^3\)Very recently, Behnezhad and Reyhani \cite{behnezhad2020query} 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 \( \epsilon \) and \( p \). Their analysis uses augmenting paths, like Blum et al. \cite{blum2021adaptive}.
algorithms for several problems.

2 General Framework

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

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:

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

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

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

$$
\underline{c}_j = \begin{cases} 
c_j & j \text{ has been queried,} 
c_j^- & \text{otherwise}, 
\end{cases} \quad (2.2)
$$

where recall that $c_j \in \mathbb{Z}_{+}$ denotes the realized value of $\hat{c}_j$. The optimistic and pessimistic LPs are obtained from the original stochastic problem (1.1) by replacing the objective vector $\hat{c}$ 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 2.1-3 ($Ax \leq b$ 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.

---

4The 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 Section 4.1.4 with a specific application.
Algorithm 1 Adaptive strategy.
1: for $t = 1, 2, \ldots, T$ do
2: Find an optimal solution $x$ to the optimistic LP.
3: For each $j = 1, \ldots, m$, 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.

Algorithm 2 Non-adaptive strategy.
1: for $t = 1, 2, \ldots, T$ do
2: Find an optimal solution $x$ to the optimistic LP.
3: For each $j = 1, \ldots, m$, 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, 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 Introduction, 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 Section 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.

(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 Section 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:

\[
\begin{align*}
\text{minimize} & \quad y^\top b \\
\text{subject to} & \quad y^\top A \geq c^\top, \\
& \quad y \in \mathbb{R}^n_+.
\end{align*}
\]

(2.3)

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

**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 $(\epsilon, \epsilon')$-witness cover for $\mu \in \mathbb{R}_+$ if it satisfies the following two properties.

1. For every $c \in \mathbb{Z}^m_+$, if $y^\top A < c^\top$ holds for all $y \in W$, then $y^\top A < c^\top$ holds 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 $y \in W$.

During the iterations, the constraints in the dual pessimistic LP (2.3) 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 (2.3) (by Condition 2). 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 $\mu$, which can be restricted to $\mu \geq 1$ due to Assumption 2.1-2 (see the proof for the detail).

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

$$T \geq \frac{\Delta_\epsilon}{\epsilon' \log \left( \frac{M}{\delta} \right)},$$

the pessimistic LP at Line 5 of Algorithm 1 has a $(1 - \epsilon)$-approximate solution with probability at least $1 - \delta$, where $\Delta_\epsilon = \max_j (c_j^+ - c_j^-)$ 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 (2.4), 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 $\mu$), 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 Sections 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 2.1-2, the $j$-th unit vector $\chi_j \in \mathbb{R}^m_+$ 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 $(\epsilon, \epsilon')$-witness cover $W_\mu$ of size $|W_\mu| \leq M^\mu$. We first evaluate the probability that each $y \in W_\mu$ 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 $\chi$ throughout Algorithm 1. Therefore, $y$ is feasible after $T$ iterations only if $y$ is feasible at every iteration step.
Fix \( t = 1, 2, \ldots, T \), 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 \( \overline{\sigma}, \underline{\sigma} \in \mathbb{Z}_+^m \) be the optimistic and pessimistic vectors, respectively, at Line 2 in the \( t \)-th step, and \( \overline{\mu}, \underline{\mu} \in \mathbb{R}_+^m \) the optimal values of the corresponding LPs. Note that \( \overline{\mu} \) and \( \underline{\mu} \) are respectively upper and lower bounds on \( \hat{\mu} \) at that time.

Claim 2.1. For every \( \mu \in [\underline{\mu}, \overline{\mu}] \) and each \( y \in W_\mu \) with \( y^T \!A \geq \overline{\sigma}^T \), 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:

\[
\overline{\sigma}^T x \leq \overline{\sigma}^T x + y^T (b - Ax) = y^T b + (\overline{\sigma}^T - y^T A) x,
\]

where \( x \in [0, 1]^m \) is the optimal solution to the optimistic LP obtained in Line 2. Since the optimistic vector \( \overline{\sigma} \) dominates the actual vector \( \hat{\sigma} \) (irrespective of which values are realized), we have \( \overline{\sigma}^T x \geq \overline{\mu} \). Since \( y \in W_\mu \), we have \( y^T b \leq (1 - \epsilon') \mu \). Therefore, by (2.5), we see

\[
\epsilon' \mu \leq (\overline{\sigma}^T - y^T A) x \leq \overline{\sigma}^T x \leq \sum_{j: \text{violated}} (c_j^+ - \underline{\sigma}_j) x_j \leq \Delta_c \sum_{j: \text{violated}} x_j,
\]

where we say that \( j \) is violated if \( (y^T A) j < c_j \) for the realized value \( c_j \) of \( \hat{\sigma}_j \), and note that \( \underline{\sigma}_j \leq c_j^+ \geq (y^T A) j \geq \underline{\sigma}_j \), and \( c_j \geq c_j^+ - \Delta_c \) for every \( j \). Since the left-hand side of (2.6) is positive, there must exist possibly violated constraints in the support of \( x \), and if one of them, say \( \hat{\sigma}_j \), 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

\[
\prod_{j: \text{violated}} (1 - px_j) \leq \exp \left( -p \sum_{j: \text{violated}} x_j \right) \leq \exp \left( -p \frac{\epsilon' \mu}{\Delta_c} \right) . \tag{2.7}
\]

By applying Claim 2.1 to \( \hat{\mu} \) (the omniscient LP-optimal value) \( T \) times, we obtain that the probability that each \( y \in W_\hat{\mu} \) is feasible after \( T \) iterations is at most \( \exp (\epsilon' p \hat{\mu} T/\Delta_c) \).

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 \( \tilde{c}^\top x \geq \tilde{\mu} \), 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

\[
\tilde{c}_R^\top x_R^* + \tilde{c}_N^\top x_N^* = \tilde{\mu},
\]

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

\[
c_N^\top x_N^* \leq \tilde{\mu}' < \tilde{\mu}/2.
\]

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

\[
\tilde{c}_N^\top x_N^* \geq \tilde{c}_R^\top x_R^* \geq \tilde{c}_R^\top x_R^* + (\tilde{c}_N^\top - c_N^\top) x_N^* > \tilde{\mu}/2.
\]

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 a TDI system admits an integral optimal solution, i.e., it is **integral**. See, e.g., [34] for more detail.

If the system is TDI, we can construct a witness cover by enumerating all possible integral 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 \((\epsilon, \epsilon)\)-witness cover for \( \mu \geq 1 \) such that \(|W| = \exp \left( O \left( \mu \log(1 + \frac{n}{\mu}) \right) \right) \):

\[
W = \{ y \in \mathbb{Z}^n_+ : y^\top b \leq (1 - \epsilon)\mu \}.
\]  

**Proof.** It is clear that \( W \) is an \((\epsilon, \epsilon)\)-witness cover for \( \mu \), 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

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

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.

**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 \( \mu, \epsilon, \gamma \), 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 $\mu \geq 1$ such that $|W| = \exp\left(O\left(\mu \left(\gamma \log \frac{n}{\mu} + \frac{1}{\epsilon}\right)\right)\right)$:

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

Proof. By Lemma 3.2, $W$ is an $(\epsilon, \epsilon/2)$-witness cover for $\mu$. 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

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

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 $\theta$ with an objective value of at most $(1+\epsilon/2)(1-\epsilon)\mu \leq \frac{\mu}{2}$.
By enumerating the dual vectors whose entries are multiples of $\theta$, 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 $|W| = \exp \left( O \left( \frac{\mu \log m}{\epsilon^2} \log \left( 1 + \frac{n}{\mu} \right) \right) \right)$:

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

**Proof.** By Theorem 3.1, $W$ is an $(\epsilon, \epsilon/2)$-witness cover for $\mu$. 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

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

### 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

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

where \( \delta(u) = \{ e \in E : u \in e \} \). The König–Egerváry theorem \cite{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 \((\epsilon, \epsilon)\)-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.** For the bipartite matching problem, by taking \( T = \Omega(\Delta c \log(n/\epsilon) / \epsilon p) \), Algorithm 1 outputs a \((1 - \epsilon)\)-approximate solution with probability at least \( 1 - \epsilon \).

This result can be improved by using the vertex sparsification lemma in Section 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:

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

It is known that the LP relaxation of this system is TDI/2 (totally dual half-integral) and has an integrality gap of \( 3/2 \) \cite{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:

\[
\begin{align*}
\text{maximize} \quad & \sum_{e \in E} \tilde{c}_e x_e \\
\text{subject to} \quad & \sum_{e \in \delta(u)} x_e \leq 1 \quad (u \in V), \\
& \sum_{e \in E(S)} x_e \leq \left\lceil \frac{|S|}{2} \right\rceil \quad (S \in V_{\text{odd}}), \\
& x \in \{0, 1\}^E,
\end{align*}
\]

(4.3)

where \(E(S) = \{ e \in E : e \subseteq S \} \) and \(V_{\text{odd}} = \{ S \subseteq V : |S| \text{ is odd and at least } 3 \} \).

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

\[
\begin{align*}
\text{minimize} \quad & \sum_{u \in V} y_u + \sum_{S \in V_{\text{odd}}} \left\lceil \frac{|S|}{2} \right\rceil z_S =: \tau(y, z) \\
\text{subject to} \quad & y_u + y_v + \sum_{S \in V_{\text{odd}} : \{u,v\} \subseteq S} z_S \geq \tilde{c}_e \quad (e = \{u,v\} \in E), \\
& y \in \mathbb{R}^V_+, \quad z \in \mathbb{R}^V_{\text{odd}}.
\end{align*}
\]

(4.4)

For \(\mu \geq 1\), let us define a set \(W \subseteq \mathbb{R}^V_+ \times \mathbb{R}^V_{\text{odd}}\) by

\[
W = \left\{ (y, z) \in \mathbb{Z}^V_+ \times \mathbb{Z}^V_{\text{odd}} : \tau(y, z) \leq (1 - \epsilon)\mu \right\}.
\]

(4.5)

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

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

**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

\[
\sum_{k=0}^{[\mu]} \binom{n}{k} \leq ([\mu] + 1) \left( \frac{en}{\mu} \right)^{[\mu]} = e^{\Theta(\mu \log n)}.
\]

(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 \(s_i \ (i = 1, 2, \ldots, k)\) be the size of each set contained in \(z\). Then we have

\[
s_1 + \cdots + s_k \leq 3 \left( \left\lceil \frac{s_1}{2} \right\rceil + \cdots + \left\lceil \frac{s_k}{2} \right\rceil \right) < 3\mu.
\]

(4.7)
Therefore, the number of candidates for $z$ is given by

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

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

Therefore, we obtain the following.

**Corollary 4.2.** For the non-bipartite matching problem, by taking $T = \Omega(\Delta_c \log(n/\epsilon)/\epsilon^p)$, 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 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

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

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 $\alpha$-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

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

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

**Claim 4.2.** If the optimal value is less than $\mu$, 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

\[
\sum_{u \in V} \sum_{e \in \delta(u)} x_e = k \sum_{e \in E} x_e < k\mu.
\]

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

Therefore, by Lemma 3.3, there exists an $(\epsilon, \epsilon/2)$-witness cover for each $\mu \geq 1$ of size at most $M^\mu$, 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.** For the $k$-hypergraph matching problem, by taking $T = \Omega(\Delta_c(k \log(n/\epsilon) + 1/\epsilon)/\epsilon p)$, 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 Section 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):

$$
\text{maximize } \sum_{j=1}^{m} \tilde{c}_j x_j
$$

subject to

$$
\sum_{j=1}^{m} a_{ij} x_j \leq b_i \quad (i = 1, \ldots, n),
$$

$$x \in \{0, 1\}^m,$$

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

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

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

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.** For the $k$-column sparse packing integer programing problem, by taking $T = \Omega(\Delta \cdot w(k \log(n/\epsilon) + 1/\epsilon)/\epsilon p)$, Algorithm 1 outputs a $(1 - \epsilon)/(2k)$-approximate solution with probability at least $1 - \epsilon$. 

18
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, without otherwise noted, \( m \) denotes the ground set size of the matroids in question.

4.2.1 Maximum Independent Set

Let \( M = (E, \mathcal{I}) \) be a matroid on a finite set \( E \), and let \( r: 2^E \to \mathbb{Z}_+ \) 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 \( \mathcal{F}_M \) denote the family of flats in \( M \). 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:

\[
\begin{align*}
\text{maximize} & \quad \sum_{e \in E} \tilde{c}_e x_e \\
\text{subject to} & \quad \sum_{e \in S} x_e \leq r(S) \quad (S \subseteq E), \\
& \quad x \in \{0, 1\}^E.
\end{align*}
\]

(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

\[
\begin{align*}
\text{minimize} & \quad \sum_{S \subseteq E} r(S) y_S \\
\text{subject to} & \quad \sum_{S \subseteq E \setminus e : e \in S} y_S \geq \tilde{c}_e \quad (e \in E), \\
& \quad y \in \mathbb{R}_+^{2^E}.
\end{align*}
\]

(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

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

(4.16)

It is clear that \( W \) is an \((\epsilon, \epsilon)\)-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. Let \(r_1, \ldots, r_k\) be the sizes of each set in \(y\). Then we have \(r_1 + \cdots + r_k < \mu\). 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 \(m^r\). Therefore, the number of dual candidates for \(y\) is given by

\[
\sum_{k=0}^{\lfloor \mu \rfloor} \sum_{r_1, \ldots, r_k \leq \mu} \left( \begin{array}{c} m \\ r_1 \end{array} \right) \cdots \left( \begin{array}{c} m \\ r_k \end{array} \right) \leq e^{O(\mu \log m)}. \tag{4.17}
\]

Therefore, we obtain the following.

Corollary 4.5. For the maximum independent set problem, by taking \(T = \Omega(\Delta_c \log(m/\epsilon)/\epsilon p)\), 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 \(r_j : 2^E \to \mathbb{Z}_+\), and \(\tilde{c} \in \mathbb{Z}_+^E\) be a stochastic weight. The weighted matroid intersection problem can be represented as

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

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. For the matroid intersection problem, by taking \(T = \Omega(\Delta_c \log(m/\epsilon)/\epsilon p)\), Algorithm 1 outputs a \((1-\epsilon)\)-approximate solution with probability at least \(1-\epsilon\).\footnote{Note that the bipartite matching problem is a special case of the matroid intersection problem, and Corollary 5.1 is obtained from a naive application of this result. Using the vertex sparsification lemma shown in Section 5, a stronger result is obtained for bipartite matching (Corollary 5.1).}
4.2.3 k-Matroid Intersection

Let $M_j = (E, I_j)$ ($j = 1, 2, \ldots, k$) 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

$$\begin{align*}
\text{maximize} & \quad \sum_{e \in E} \tilde{c}_e x_e \\
\text{subject to} & \quad \sum_{e \in S} x_e \leq r_j(S) \quad (S \subseteq E, \ j \in \{1, 2, \ldots, k\}),
\end{align*}$$

(4.19)

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 $W \subseteq (\mathbb{R}_+^{2^V})^k$ by

$$W = \left\{ (y^1, \ldots, y^k) \in (\theta\mathbb{Z}_+^V)^k : \sum_{j,S} r_j(S)y^j_S \leq \left(1 - \frac{\epsilon}{2}\right) \mu, \supp(y^j) \subseteq F_{M_j} \ (\forall j = 1, \ldots, k) \right\}.$$

(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. As same as the matroid case, we regard $y^j$ as a multiset in which each set $S$ contributes $\theta$. Let $r_1, \ldots, r_k$ be the size of each set in $y^j$. Then we have $r_1 + \cdots + r_k < \mu/\theta$. By the same argument as for the matroid case, 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 $\theta = \Theta(\epsilon^2/\log m)$).

Therefore, we obtain the following.

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

21
4.2.4 Matroid Matching

The matroid matching problem is a common generalization of the matching problem and the matroid intersection problem. Let \((V, E)\) be a graph with \(|V| = n\), \(M = (V, \mathcal{I})\) be a matroid whose rank function is \(r: 2^V \to \mathbb{Z}_+\), and \(\tilde{c} \in \mathbb{Z}_+^E\) be a stochastic edge weight. The task is to find a maximum-weight subset of edges such that the endpoints form an independent set of the matroid \(M\). Vande Vate [37] proposed the following LP formulation:

\[
\begin{align*}
\text{maximize} & \quad \sum_{e \in E} \tilde{c}_e x_e \\
\text{subject to} & \quad \sum_{e \in E} a(S, e)x_e \leq r(S) \quad (S \subseteq V), \\
& \quad x \in \{0, 1\}^E,
\end{align*}
\]

where \(a(S, e) = |S \cap e|\). Gijswijt and Pap [20] showed that the system is TDI/2, and proposed a polynomial-time algorithm for obtaining a fractional optimal solution. A \((2/3)\)-approximate integral solution can be obtained by rounding this solution by the procedure of Lee, Sviridenko, and Vondrák [27]. The expected number of revealed edges is \(O(T)\) per vertex since \(\sum_{e \in \delta(u)} x_e = \sum_{e \in E} a({\{u\}}, e) x_e \leq r({\{u\}}) \leq 1\).

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

\[
\begin{align*}
\text{minimize} & \quad \sum_{S \subseteq V} r(S)y_S \\
\text{subject to} & \quad \sum_{S \subseteq V: e \in S} a(S, e)y_S \geq \tilde{c}_e \quad (e \in E),
\end{align*}
\]

Since \(a(S, e)\) is monotonically nondecreasing in \(S\), we can restrict the support of \(y\) to the set of flats. We define a witness cover by

\[
W = \left\{ y \in \frac{1}{2}\mathbb{Z}_+^V : \sum_{S \subseteq V} r(S)y_S \leq (1 - \epsilon)\mu, \supp(y) \subseteq \mathcal{F}_M \right\}.
\]

By the same argument as for the matroid maximization problem, we obtain the following.

**Corollary 4.8.** For the matroid matching problem, by taking \(T = \Omega(\Delta_c \log(n/\epsilon)/\epsilon p)\), Algorithm 1 outputs a \((2 - \epsilon)/3\)-approximate solution with probability at least \(1 - \epsilon\).

\[\text{\footnotesize \footnote{Lee, Sviridenko and Vondrák [27] proposed a rounding procedure for a slightly different formulation. However, since the solution to the present formulation can be easily converted to the Lee et al.’s formulation, we can use the rounding procedure.}}\]
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 \(M = (V, \mathcal{I})\) 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

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

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

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

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

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

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 \(z\):

\[
\text{minimize } \sum_{S \subseteq V} r(S)z_S \\
\text{subject to } \sum_{S \subseteq V: u \in S} z_S \geq \bar{c}_u - \sum_{e \in \delta(u)} y'_e \quad (u \in V).
\]

(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 \((1 - \epsilon/2)\mu\).

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 \(W \subseteq \mathbb{R}^E_+ \times \mathbb{R}^{2^V}_+\) by

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

(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.

Therefore, we obtain the following.

Corollary 4.9. For the degree bounded matroid problem with maximum degree \(d\), by taking \(T = \Omega(\Delta d \log(n/\epsilon)/\epsilon^2 p)\), 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 $\alpha$ (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 $\tilde{c}: V \to \mathbb{Z}_+$ be a stochastic vertex weight. The maximum stable set problem can be represented as

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

(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:

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

(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

$$\begin{align*}
\text{minimize} & \quad \sum_{C \in \mathcal{C}} y_C \\
\text{subject to} & \quad \sum_{C \in \mathcal{C}: u \in C} y_C \geq c_u \quad (u \in V),
\end{align*}$$

(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 $O(k \log(n/e)/\epsilon^p)$. 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:

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

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\(^7\), 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

\[
\begin{align*}
\text{minimize} & \quad \sum_{e \in E} y_e + \sum_{C \in \mathcal{C}} \left\lfloor \frac{|C|}{2} \right\rfloor \tilde{z}_C \\
\text{subject to} & \quad \sum_{e \in \delta(u)} y_e + \sum_{C \in \mathcal{C} : u \in C} z_C \geq \tilde{c}_u \quad (u \in V),
\end{align*}
\]

where \(\mathcal{C}\) is the set of odd cycles. We regard \(z_C\) as a multiset in which each odd cycle \(C\) contributes \(\theta\). Let \(c_1, \ldots, c_k\) be the sizes of each odd cycles. We then have \(c_1 + \cdots + c_k < \mu/\theta\). We define the witness cover by

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

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. For the t-stable set problem, by taking \(T = \Omega(\Delta_{\epsilon} \log n \log(n/\epsilon)/\epsilon^3 p)\), Algorithm 1 outputs a \((1 - \epsilon)\)-approximate solution with probability at least \(1 - \epsilon\).\(^8\)

\(^7\)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.

\(^8\)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.
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 \( \mu \) while approximately preserving any matchings of size \( \nu = \omega(1) \) with high probability. This procedure is very useful as a preprocessing step 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 \( \nu = \omega(1) \). 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 is aimed to for reducing space complexity of packing problems in streaming setting, but these conduct different analysis and provide different guarantees.

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 \( X \subseteq Y \in \mathcal{I} \Rightarrow X \in \mathcal{I} \)), whose rank function \( r: 2^E \to \mathbb{Z}_+ \) 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 (4.1) in this section:

\[
\begin{align*}
\text{maximize} & \quad \sum_{e \in E} \tilde{c}_e x_e \\
\text{subject to} & \quad \sum_{e \in S} x_e \leq r(S) \quad (S \subseteq E), \\
& \quad x \in \{0, 1\}^E. 
\end{align*}
\]

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 Section 4.1) as follows: let \( V = \{1, \ldots, n\} \) and \( E = \{1, \ldots, m\} \) 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 \( \{1, \ldots, n^\circ\} \) to each vertex, where \( n^\circ = \beta(k, \epsilon, \delta)k^2 s/\delta \) with \( \beta(k, \epsilon, \delta) = 2e^{c/k} \log(1/\delta)/\epsilon \). 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.
Algorithm 3 Vertex sparsification.

1: Assign a random color in \(\{1, \ldots, \beta(k, \epsilon, \delta)k^2s\}\) to each vertex, where \(\beta(k, \epsilon, \delta) = \frac{2e^\epsilon/k \log(1/\delta)}{\epsilon}\).

2: Return all colorful hyperedges.

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 \(I^* \subseteq I\) of size at least \((1 - \epsilon)|I|\) in the sparsified instance with probability at least 1 − δ.

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

Case 1: \(\nu \leq \beta\) (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 \(k\nu\), the probability that this has occurred is at least

\[
\frac{n^\nu(n^\nu - 1) \cdots (n^\nu - k\nu + 1)}{n^\nu k^\nu} \geq \exp\left(\frac{-k^2\nu^2}{n^\nu}\right) \geq \exp\left(-\frac{\delta\nu^2}{\beta s}\right) \geq e^{-\delta} \geq 1 - \delta. \tag{5.2}
\]

Here, the first inequality follows from the falling factorial approximation (Lemma 5.2 below), and the second inequality follows from \(\nu \leq r \leq s\) and \(\nu \leq \beta\).

Lemma 5.2 (Falling Factorial Approximation).

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

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

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

Case 2: \(\nu \geq \beta\) (the rank of \(I\) is large). We further reduce the number of colors by mapping each color class to \(\{1, \ldots, k^2\nu/\epsilon\}\). (Note that \(k^2\nu/\epsilon \leq n^\nu\) 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 \(\Pr(X_c = 1) = (1 - \epsilon/k^2\nu)^{k\nu} \leq e^{-\epsilon/k}\). Therefore \(\mathbb{E}[\sum_c X_c] \leq e^{-\epsilon/k}k^2\nu/\epsilon\). Since \(X_c\) are negatively correlated random variables, we can apply
the Chernoff bound [29]:

\[
\Pr \left( \sum_c X_c \geq \frac{k^2 \nu}{\epsilon} - \left( 1 - \frac{\epsilon}{k} \right) k \nu \right) = \Pr \left( \sum_c X_c \geq \left( 1 - \frac{\epsilon}{k} + \frac{\epsilon^2}{k^2} \right) \frac{k^2 \nu}{\epsilon} \right)
\leq \Pr \left( \sum_c X_c \geq \left( 1 + \frac{\epsilon^2}{2k^2} \right) e^{-\epsilon/k} \frac{k^2 \nu}{\epsilon} \right)
\leq \exp \left( -\epsilon e^{-\epsilon/k} \frac{\nu}{2} \right) \leq \exp \left( -\epsilon e^{-\epsilon/k} \frac{\beta}{2} \right) = \delta, \quad (5.5)
\]

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 \(\beta\). 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 \((1 - \epsilon) \nu\).

**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 \((E, I)\) be an independence system. We consider the problem (5.1), where we assume the following.

1. There exists an LP-relative \(\alpha\)-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 \(\epsilon, \delta \in (0, 1)\) be 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 \(\alpha\)-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 \(1 - \delta\).
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 $\epsilon' = \epsilon/(1 + c_{\text{max}})$ and probability parameter $\delta' = \delta/4$.
3: Run Algorithm 1 or 2 with an LP-relative $\alpha$-approximation algorithm by setting $T = T(O(\log(k / p \alpha \epsilon'))$, $\epsilon'$, $\delta'$).

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

Claim 5.1.

$$\Pr(r^\circ \geq (1 - \epsilon')r) \geq 1 - \delta', \quad (5.6)$$
$$\Pr(\hat{\mu}^\circ \geq (1 - c_{\text{max}} \epsilon')\hat{\mu}) \geq 1 - \delta'. \quad (5.7)$$

Proof. The first inequality (5.6) immediately follows from Lemma 5.1, and we focus on the second (5.7). Fix a realization of $\hat{c}$, 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

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

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

By Claim 5.1 we have $r^\circ \geq (1 - \epsilon')r$ and $\hat{\mu}^\circ \geq (1 - c_{\text{max}} \epsilon')\hat{\mu}$ 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 $(1 - \epsilon')\hat{\mu}^\circ \geq (1 - (c_{\text{max}} + 1)\epsilon')\hat{\mu} = (1 - \epsilon)\hat{\mu}$ 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

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

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. $r^o \geq 8 \log(1/\delta')/p$ (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

$$\Pr\left(\hat{\mu}^o \geq \frac{pr^o}{2}\right) \geq \Pr\left(\sum_{i=1}^{r^o} X_i \geq \frac{pr^o}{2}\right) \geq 1 - e^{-pr^o/8} \geq 1 - \delta'$$

(5.10)

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

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

(5.11)

where the second inequality follows from $r^o \geq (1 - \epsilon')r \geq r/2$ (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. $r^o \leq 8 \log(1/\delta')/p$ (the rank of the sparsified instance is small). We have

$$n^o \mu^o \leq \frac{2n^o}{pr^o} \leq \frac{4n^o}{pr} \leq \frac{4n^o}{pas} = \frac{16\beta'k^2 \log(1/\delta')}{p\alpha\delta'},$$

(5.12)

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

The sizes of witness covers of bipartite matching, $k$-hypergraph matching, and $k$-column-sparse packing integer programming depend on $n/\mu$. 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 $O(\log(1/\epsilon)p)/ep$ 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 $O(k(\log(k/ep) + 1/\epsilon)/ep)$ 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 $O(k(\log(k/ep) + 1/\epsilon)/ep)$ queries per vertex and finds $(1 - \epsilon)/2k$-approximate solution with probability at least $1 - \epsilon$. 

31
Acknowledgment

The authors thank anonymous reviewers for their valuable comments. This work was supported by JSPS KAKENHI Grant Numbers 16H06931 and 16K16011.

References

[1] Marek Adamczyk. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters*, 111(15):731–737, 2011.

[2] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Mathematics of Operations Research*, 41(3):1022–1038, 2016.

[3] Sepehr Assadi, Sanjeev Khanna, and Yang Li. The stochastic matching problem with (very) few queries. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 43–60. ACM, 2016.

[4] Sepehr Assadi, Sanjeev Khanna, and Yang Li. The stochastic matching problem: Beating half with a non-adaptive algorithm. *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 99–116, 2017.

[5] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[6] Soheil Behnezhad and Nima Reyhani. Almost optimal stochastic weighted matching with few queries. *arXiv preprint arXiv:1710.10592*, 2017.

[7] Avrim Blum, John P Dickerson, Nika Haghtalab, Ariel D Procaccia, Tuomas Sandholm, and Ankit Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Proceedings of the 2015 ACM Conference on Economics and Computation*, pages 325–342. ACM, 2015.

[8] Avrim Blum, Anupam Gupta, Ariel Procaccia, and Ankit Sharma. Harnessing the power of two crossmatches. In *Proceedings of the 2013 ACM conference on Electronic Commerce*, pages 123–140. ACM, 2013.

[9] Yuk Hei Chan and Lap Chi Lau. On linear and semidefinite programming relaxations for hypergraph matching. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1500–1511, 2010.

[10] Ning Chen, Nicole Immorlica, Anna R Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *Proceedings of the 2009 International Colloquium on Automata, Languages, and Programming*, pages 266–278. Springer, 2009.
[11] Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1326–1344. Society for Industrial and Applied Mathematics, 2016.

[12] Kevin Costello, Prasad Tetali, and Pushkar Tripathi. Stochastic matching with commitment. *Automata, Languages, and Programming*, pages 822–833, 2012.

[13] William H Cunningham and AB Marsh. A primal algorithm for optimum matching. *Polyhedral Combinatorics*, pages 50–72, 1978.

[14] Brian C Dean, Michel X Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 395–404, 2005.

[15] Brian C Dean, Michel X Goemans, and J Vondrak. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217, 2004.

[16] John P Dickerson and Tuomas Sandholm. Organ exchanges: A success story of ai in healthcare. 2016.

[17] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[18] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. *Edited by G. Goos, J. Hartmanis, and J. van Leeuwen*, 11, 1970.

[19] Eugene Egerváry. *On combinatorial properties of matrices*. 1931.

[20] Dion Gijswijt and Gyula Pap. An algorithm for weighted fractional matroid matching. *Journal of Combinatorial Theory, Series B*, 103(4):509–520, 2013.

[21] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Proceedings of the 2013 International Conference on Integer Programming and Combinatorial Optimization*, pages 205–216, 2013.

[22] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and xos functions. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1688–1702. SIAM, 2017.

[23] Cor A. J. Hurkens and Alexander Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.
[24] Tamás Király, Lap Chi Lau, and Mohit Singh. Degree bounded matroids and submodular flows. *Combinatorica*, 32(6):703–720, 2012.

[25] Stavros G Kolliopoulos and Neal E Young. Approximation algorithms for covering/packing integer programs. *Journal of Computer and System Sciences*, 71(4):495–505, 2005.

[26] D König. Graphs and matrices. *Math. Fiz. Lapok*, 38:116–119, 1931.

[27] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Matroid matching: the power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013.

[28] Marco Molinaro and R Ravi. The query-commit problem. *arXiv preprint arXiv:1110.0990*, 2011.

[29] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.

[30] Ojas Parekh. Iterative packing for demand and hypergraph matching. In *Proceedings of the 2011 International Conference on Integer Programming and Combinatorial Optimization*, pages 349–361, 2011.

[31] Ojas Parekh and David Pritchard. Generalized hypergraph matching via iterated packing and local ratio. In *International Workshop on Approximation and Online Algorithms*, pages 207–223. Springer, 2014.

[32] Prabhakar Raghavan and Clark D Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

[33] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.

[34] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. 2003.

[35] F Bruce Shepherd and Adrian Vetta. The demand-matching problem. *Mathematics of Operations Research*, 32(3):563–578, 2007.

[36] Paul Turán. On the theory of graphs. In *Colloquium Mathematicae*, volume 3, pages 19–30. Institute of Mathematics Polish Academy of Sciences, 1954.

[37] John H Vande Vate. Fractional matroid matchings. *Journal of Combinatorial Theory, Series B*, 55(1):133–145, 1992.