

Approximation-Friendly Discrepancy Rounding

Nikhil Bansal*

Viswanath Nagarajan[†]

December 2, 2016

Abstract

Rounding linear programs using techniques from discrepancy is a recent approach that has been very successful in certain settings. However this method also has some limitations when compared to approaches such as randomized and iterative rounding. We provide an extension of the discrepancy-based rounding algorithm due to Lovett-Meka that (i) combines the advantages of both randomized and iterated rounding, (ii) makes it applicable to settings with more general combinatorial structure such as matroids. As applications of this approach, we obtain new results for various classical problems such as linear system rounding, degree-bounded matroid basis and low congestion routing.

1 Introduction

A very common approach for solving discrete optimization problems is to solve some linear programming relaxation, and then round the fractional solution into an integral one, without (hopefully) incurring much loss in quality. Over the years several ingenious rounding techniques have been developed (see e.g. [Vaz01, WS11]) based on ideas from optimization, probability, geometry, algebra and various other areas. Randomized rounding and iterative rounding are two of the most commonly used methods.

Recently, discrepancy-based rounding approaches have also been very successful; a particularly notable result is for bin packing due to Rothvoss [Rot13]. Discrepancy is a well-studied area in combinatorics with several surprising results (see e.g. [Mat10]), and as observed by Lovász et al. [LSV86], has a natural connection to rounding. However, until the recent algorithmic developments [Ban10, LM12, HSS14, NT15, Rot14], most of the results in discrepancy were non-constructive and hence not directly useful for rounding. These algorithmic approaches combine probabilistic approaches like randomized rounding with linear algebraic approaches such as iterated rounding [LRS11], which makes them quite powerful.

Interestingly, given the connection between discrepancy and rounding, these discrepancy algorithms can in fact be viewed as meta-algorithms for rounding. We discuss this in §1.1 in the context of the Lovett-Meka (LM) algorithm [LM12]. This suggests the possibility of one single approach that generalizes both randomized and iterated rounding. This is our motivating goal in this paper.

While the LM algorithm is already an important step in this direction, it still has some important limitations. For example, it is designed for obtaining additive error bounds and it does not give good multiplicative error bounds (like those given by randomized rounding). This is not an issue for discrepancy applications, but crucial for many approximation algorithms. Similarly, iterated rounding can work well with exponentially sized LPs by exploiting their underlying combinatorial structure (e.g., degree-bounded spanning tree [SL07]), but the current discrepancy results [LM12, Rot14] give extremely weak bounds in such settings.

*Department of Mathematics and Computer Science, Eindhoven University of Technology.

[†]Department of Industrial and Operations Engineering, University of Michigan.

Our Results: We extend the LM algorithm to overcome the limitations stated above. In particular, we give a new variant that also gives Chernoff type multiplicative error bounds (sometimes with an additional logarithmic factor loss). We also show how to adapt the above algorithm to handle exponentially large LPs involving matroid constraints, as in iterated rounding.

This new discrepancy-based algorithm gives new results for problems such as linear system rounding with violations [BF81, LLRS01], degree-bounded matroid basis [KLS08, CVZ10], low congestion routing [KLR⁺87, LLRS01] and multi-budgeted matroid basis [GRSZ14]. These results simultaneously combine non-trivial guarantees from discrepancy, randomized rounding and iterated rounding and previously such bounds were not even known existentially.

Our results are described formally in §1.2. To place them in the proper context, we first need to describe some existing rounding approaches (§1.1). The reader familiar with the LM algorithm can directly go to §1.2.

1.1 Preliminaries

We begin by describing LM rounding [LM12], randomized rounding and iterated rounding in a similar form, and then discuss their strengths and weaknesses.

LM Rounding: Let A be a $m \times n$ matrix with 0–1 entries¹, $x \in [0, 1]^n$ a fractional vector and let $b = Ax$. Lovett and Meka [LM12] showed the following rounding result.

Theorem 1 (LM Rounding [LM12]). *Given A and x as above, For $j = 1, \dots, m$, pick any λ_j satisfying*

$$\sum_j \exp(-\lambda_j^2/4) \leq n/16. \quad (1)$$

Then there is an efficient randomized algorithm to find a solution x' such that: (i) at most $n/2$ variables of x' are fractional (strictly between 0 and 1) and, (ii) $|\langle a_j, x' - x \rangle| \leq \lambda_j \|a_j\|_2$ for each $j = 1, \dots, m$, where a_j denotes the j -th row of A .

Remark: The right hand side of (1) can be set to $(1 - \epsilon)n$ for any fixed constant $\epsilon > 0$, at the expense of $O_\epsilon(1)$ factor loss in other parameters of the theorem; see e.g. [BCKL14].

Randomized Rounding: Chernoff bounds state that if X_1, \dots, X_n are independent Bernoulli random variables, and $X = \sum_i X_i$ and $\mu = \mathbb{E}[X]$, then

$$\Pr[|X - \mu| \geq \epsilon\mu] \leq 2 \exp(-\epsilon^2\mu/4) \quad \text{for } \epsilon \leq 1.$$

Then independent randomized rounding can be viewed as the following (by using Chernoff bounds and union bound, and denoting $\lambda_j = \epsilon_j \sqrt{b_j}$).

Theorem 2 (Randomized Rounding). *For $j = 1, \dots, m$, pick any λ_j satisfying $\lambda_j \leq \sqrt{b_j}$, and*

$$\sum_j \exp(-\lambda_j^2/4) < 0.5 \quad (2)$$

Then independent randomized rounding gives a solution x' such that: (i) All variables are 0-1, and (ii) $|\langle a_j, x' - x \rangle| \leq \lambda_j \sqrt{b_j}$ for each $j = 1, \dots, m$.

¹The results below generalize to arbitrary real matrices A and vectors x in natural ways, but we consider 0-1 case for simplicity.

Iterated Rounding [LRS11]: This is based on the following linear-algebraic fact.

Theorem 3. *If $m < n$, then there is a solution $x' \in [0, 1]^n$ such that (i) x' has at least $n - m$ variables set to 0 or 1 and, (ii) $A(x' - x) = 0$ (i.e., $b = Ax'$).*

In iterated rounding applications, if $m > n$ then some cleverly chosen constraints are dropped until $m < n$ and integral variables are obtained. This is done repeatedly.

Strengths of LM rounding: Note that if we set $\lambda_j \in \{0, \infty\}$ in LM rounding, then it gives a very similar statement to Theorem 3. E.g., if we only care about some $m = n/2$ constraints then Theorem 3 gives an x' with at least $n/2$ integral variables and $a_j x = a_j x'$ for all these m constraints. Theorem 1 (and the remark below it) give the same guarantee if we set $\lambda_j = 0$ for all constraints. In general, LM rounding can be much more flexible as it allows arbitrary λ_j .

Second, LM rounding is also related to randomized rounding. Note that (2) and (1) have the same left-hand-side. However, the right-hand-side of (1) is $\Omega(n)$, while that of (2) is $O(1)$. This actually makes a huge difference. In particular, in (2) one cannot set $\lambda_j = 1$ for more than a couple of constraints (to get an $o(\sqrt{b_j})$ error bound on constraints), while in (1), one can even set $\lambda_j = 0$ for $O(n)$ constraints. In fact, almost all non-trivial results in discrepancy [Spe85, Sri97, Mat10] are based on this ability.

Weaknesses of LM rounding: First, Theorem 1 only gives a partially integral solution instead of a fully integral one as in Theorem 2.

Second, and more importantly, it only gives additive error bounds instead of multiplicative ones. In particular, note the $\lambda_j \|a_j\|_2$ vs $\lambda_j \sqrt{b_j}$ error in Theorems 1 and 2. E.g., for a constraint $\sum_i x_i = \log n$, Theorem 2 gives $\lambda \sqrt{\log n}$ error but Theorem 1 gives a much higher $\lambda \sqrt{n}$ error. So, while randomized rounding can give a good multiplicative error like $a_j x' \leq (1 \pm \epsilon_j) b_j$, LM rounding is completely insensitive to b_j .

Finally, iterated rounding works extremely well in many settings where Theorem 1 does not give anything useful. E.g., in problems involving exponentially many constraints such as the degree bounded spanning tree problem. The problem is that if m is exponentially large, then the λ_j 's in Theorem 1 need to be very large to satisfy (2).

1.2 Our Results and techniques

Our first result is the following improvement over Theorem 1.

Theorem 4. *There is a constant $K_0 > 0$ and randomized polynomial time algorithm that given any $n > K_0$, fractional solution $y \in [0, 1]^n$, $m \leq 2^n$ linear constraints $a_1, \dots, a_m \in \mathbb{R}^n$ and $\lambda_1, \dots, \lambda_m \geq 0$ with $\sum_{j=1}^m e^{-\lambda_j^2/K_0} < \frac{n}{16}$, finds a solution $y' \in [0, 1]^n$ such that:*

$$|\langle y' - y, a_j \rangle| \leq \lambda_j \cdot \sqrt{W_j(y)} + \frac{1}{n^2} \cdot \|a_j\|, \quad \forall j = 1, \dots, m \quad (3)$$

$$y'_i \in \{0, 1\}, \quad \text{for } \Omega(n) \text{ indices } i \in \{1, \dots, n\} \quad (4)$$

Here $W_j(y) := \sum_{i=1}^n a_{ji}^2 \cdot \min\{y_i, 1 - y_i\}^2$ for each $j = 1, \dots, m$.

Remarks: 1) The error $\lambda_j \sqrt{W_j(y)}$ is always smaller than $\lambda_j \|a_j\|$ in LM-rounding and $\lambda_j (\sum_{i=1}^n a_{ji}^2 \cdot y_i(1 - y_i))^{1/2}$ in randomized rounding. In fact it could even be much less if the y_i are very close to 0 or 1.

2) The term $n/16$ above can be made $(1 - \epsilon)n$ for any fixed constant $\epsilon > 0$, at the expense of worsening other constants (just as in LM rounding).

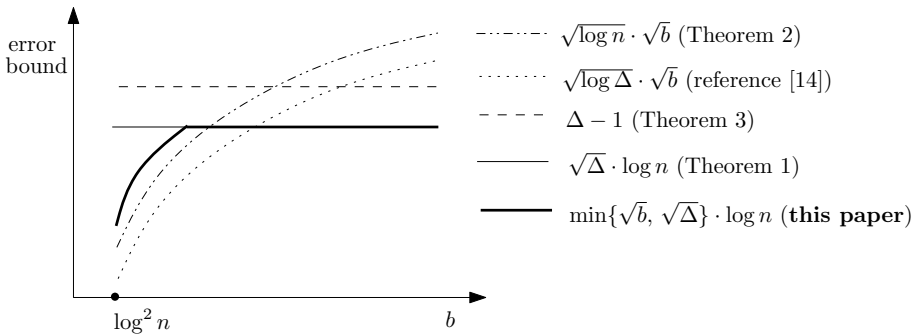


Figure 1: Additive violation bounds for linear system rounding when $\Delta \geq \log^2 n$ and $b \geq \log^2 n$.

3) The additional error term $\frac{1}{n^c} \cdot \|a_j\|$ above is negligible and can be reduced to $\frac{1}{n^c} \cdot \|a_j\|$ for any constant c , at the expense of a larger running time $n^{O(c)}$.

We note that Theorem 4 can also be obtained in a “black box” manner from LM-rounding (Theorem 1) by rescaling the polytope and using its symmetry.² However, such an approach does not work in the setting of matroid polytopes (Theorem 5). In the matroid case, we need to modify LM-rounding as outlined below.

Applications: We focus on *linear system rounding* as the prime example. Here, given matrix $A \in [0, 1]^{m \times n}$ and vector $b \in \mathbb{Z}_+^m$, the goal is to find a vector $z \in \{0, 1\}^n$ satisfying $Az = b$. As this is NP-hard, the focus has been on finding a $z \in \{0, 1\}^n$ where $Az \approx b$.

Given any fractional solution $y \in [0, 1]^n$ satisfying $Ay = b$, using Theorem 4 iteratively we can obtain an integral vector $z \in \{0, 1\}^n$ with

$$|a_j z - b_j| \leq \min \left\{ O(\sqrt{n \log(2 + m/n)}), \sqrt{L \cdot b_j} + L \right\}, \quad \forall j \in [m], \quad (5)$$

where $L = O(\log n \log m)$ and $[m] := \{1, 2, \dots, m\}$.³ Previously known algorithms could provide a bound of either $O(\sqrt{n \log(m/n)})$ for all constraints [LM12] or $O(\sqrt{\log m} \cdot \sqrt{b_j} + \log m)$ for all constraints (Theorem 2). Note that this does not imply a $\min\{\sqrt{n \log(m/n)}, \sqrt{\log m} \cdot \sqrt{b_j} + \log m\}$ violation per constraint, as in general it is not possible to combine two integral solutions and achieve the better of their violation bounds on all constraints. To the best of our knowledge, even the existence of an integral solution satisfying the bounds in (5) was not known prior to our work.

In the setting where the matrix A is “column sparse”, i.e. each variable appears in at most Δ constraints, we obtain a more refined error of

$$|a_j y - b_j| \leq \min \left\{ O(\sqrt{\Delta} \log n), \sqrt{L \cdot b_j} + L \right\}, \quad \forall j \in [m], \quad (6)$$

where $L = O(\log n \cdot \log m)$. Previous algorithms could separately achieve bounds of $\Delta - 1$ [BF81], $O(\sqrt{\Delta} \log n)$ [LM12] or $O(\sqrt{\log \Delta} \cdot \sqrt{b_j} + \log \Delta)$ [LLRS01]. For clarity, Figure 1 plots the violation bounds achieved by these different algorithms as a function of the right-hand-side b when $m = n$ (we assume $b, \Delta \geq \log^2 n$). Note again that since there are multiple constraints we can not simply combine algorithms to achieve the smaller of their violation bounds.

One can also combine the bounds in (5) and (6), and use some additional ideas from discrepancy to obtain:

$$|a_j y - b_j| \leq O(1) \cdot \min \left\{ \sqrt{j}, \sqrt{n \log(2 + \frac{m}{n})}, \sqrt{L \cdot b_j} + L, \sqrt{\Delta} \log n \right\}, \quad \forall j \in [m]. \quad (7)$$

²We thank an anonymous reviewer for pointing this out.

³For any integer $t \geq 1$, we use the notation $[t] := \{1, 2, \dots, t\}$.

Matroid Polytopes: Our main result is an extension of Theorem 4 where the fractional solution lies in a matroid polytope in addition to satisfying the linear constraints $\{a_j\}_{j=1}^m$. Recall that a matroid \mathcal{M} is a tuple (V, \mathcal{I}) where V is the groundset of elements and $\mathcal{I} \subseteq 2^V$ is a collection of independent sets satisfying the hereditary and exchange properties [Sch03]. The rank function $r : 2^V \rightarrow \mathbb{Z}$ of a matroid is defined as $r(S) = \max_{I \in \mathcal{I}, I \subseteq S} |I|$. The matroid polytope (i.e. convex hull of all independent sets) is given by the following linear inequalities:

$$P(\mathcal{M}) := \left\{ x \in \mathbb{R}^n : \sum_{i \in S} x_i \leq r(S) \ \forall S \subseteq V, \ x \geq 0 \right\}.$$

As is usual when dealing with matroids, we assume access to an “independent set oracle” for \mathcal{M} that given any subset $S \subseteq V$ returns whether/not $S \in \mathcal{I}$ in polynomial time.

Theorem 5. *There is a randomized polynomial time algorithm that given matroid \mathcal{M} , fractional solution $y \in P(\mathcal{M})$, linear constraints $\{a_j \in \mathbb{R}^n\}_{j=1}^m$ and values $\{\lambda_j\}_{j=1}^m$ satisfying the conditions in Theorem 4, finds a solution $y' \in P(\mathcal{M})$ satisfying (3)-(4).*

We note that the same result can be obtained even if we want to compute a base (maximal independent set) in the matroid: the only difference here is to add the equality $\sum_{i \in V} x_i = r(V)$ to $P(\mathcal{M})$ which corresponds to the base polytope of \mathcal{M} .

The fact that we can *exactly* preserve matroid constraints leads to a number of improvements:

Degree-bounded matroid basis (DEGMAT). Given a matroid on elements $[n] := \{1, 2, \dots, n\}$ with costs $d : [n] \rightarrow \mathbb{Z}_+$ and m “degree constraints” $\{S_j, b_j\}_{j=1}^m$ where each $S_j \subseteq [n]$ and $b_j \in \mathbb{Z}_+$, the goal is to find a minimum-cost basis I in the matroid that satisfies $|I \cap S_j| \leq b_j$ for all $j \in [m]$. Since even the feasibility problem is NP-hard, we consider *bicriteria* approximation algorithms that violate the degree bounds. We obtain an algorithm where the solution costs at most the optimal and the degree bound violation is as in (7); here Δ denotes the maximum number of sets $\{S_j\}_{j=1}^m$ containing any element.

Previous algorithms achieved approximation ratios of $(1, b + O(\sqrt{b \log n}))$ [CVZ10], based on randomized swap rounding, and $(1, b + \Delta - 1)$ [KLS08] based on iterated rounding. Again, these bounds could not be combined together as they used different algorithms. We note that in general the $(1, b + O(\sqrt{n \log(m/n)}))$ approximation is the best possible (unless $P=NP$) for this problem [CNN11, BKK⁺13].

Multi-criteria matroid basis. Given a matroid on elements $[n]$ with k different cost functions $d_i : [n] \rightarrow \mathbb{Z}_+$ (for $i = 1, \dots, k$) and budgets $\{B_i\}_{i=1}^k$, the goal is to find (if possible) a basis I with $d_i(I) \leq B_i$ for each $i \in [k]$. We obtain an algorithm that for any $\epsilon > 0$ finds in $n^{O(k^{1.5}/\epsilon)}$ time, a basis I with $d_i(I) \leq (1 + \epsilon)B_i$ for all $i \in [k]$. Previously, [GRSZ14] obtained such an algorithm with $n^{O(k^2/\epsilon)}$ running time.

Low congestion routing. Given a directed graph $G = (V, E)$ with edge capacities $b : E \rightarrow \mathbb{Z}_+$, k source-sink pairs $\{(s_i, t_i)\}_{i=1}^k$ and a length bound Δ , the goal is to find an $s_i - t_i$ path P_i of length at most Δ for each pair $i \in [k]$ such that the number N_e of paths using any edge e is at most b_e . Using an LP-based reduction [CVZ10] this can be cast as an instance of DEGMAT. So we obtain violation bounds as in (7) which implies:

$$N_e \leq b_e + \min \left\{ O(\sqrt{\Delta} \log n), O(\sqrt{b_e} \log n + \log^2 n) \right\}, \quad \forall e \in E.$$

Here $n = |V|$ is the number of vertices. Previous algorithms achieved bounds of $\Delta - 1$ [KLR⁺87] or $O(\sqrt{\log \Delta} \cdot \sqrt{b_j} + \log \Delta)$ [LLRS01] separately. We can also handle a richer set of routing requirements: given a laminar family \mathcal{L} on the k pairs, with a requirement r_T on each set $T \in \mathcal{L}$, we want to find a multiset of paths so that there are at least r_T paths between the pairs in each $T \in \mathcal{L}$. Although this is not an instance of DEGMAT, the same approach works.

Overview of techniques: Our algorithm in Theorem 4 is similar to the Lovett-Meka algorithm, and is also based on performing a Gaussian random walk at each step in a suitably chosen subspace. However, there are some crucial differences. First, instead of updating each variable by the standard Gaussian $N(0, 1)$, the variance for variable i is chosen proportional to $\min(y_i, 1 - y_i)$, i.e. proportional to how close it is to the boundary 0 or 1. This is crucial for getting the multiplicative error instead of the additive error in the constraints. However, this slows down the “progress” of variables toward reaching 0 or 1. To get around this, we add $O(\log n)$ additional constraints to define the subspace where the walk is performed: these restrict the total fractional value of variables in a particular “scale” to remain fixed. Using these we can ensure that enough variables eventually reach 0 or 1.

In order to handle the matroid constraints (Theorem 5) we need to incorporate them (although they are exponentially many) in defining the subspace where the random walk is performed. One difficulty that arises here is that we can no longer implement the random walk using “near tight” constraints as in [LM12] since we are unable to bound the dimension of near-tight matroid constraints. However, as is well known, the dimension of *exactly* tight matroid constraints is at most $n/2$ at any (strictly) fractional solution, and so we implement the random walk using exactly tight constraints. This requires us to truncate certain steps in the random walk (when we move out of the polytope), but we show that the effect of such truncations is negligible.

2 Matroid Partial Rounding

In this section we will prove Theorem 5 which also implies Theorem 4.

We may assume, without loss of generality, that $\max_{j=1}^m \lambda_j \leq n$. This is because setting $\mu_j = \min\{\lambda_j, n\}$ we have $\sum_{j=1}^m e^{-\mu_j^2/K_0} \leq \sum_{j=1}^m e^{-\lambda_j^2/K_0} + m \cdot e^{-n} < \frac{n}{16} + 1$ (we used the assumption $m \leq 2^n$). So we can apply Theorem 5 with μ_j s instead of λ_j s to obtain a stronger result.

Let $y \in \mathbb{R}^n$ denote the initial solution. The algorithm will start with $X_0 = y$ and update this vector over time. Let X_t denote the vector at time t for $t = 1, \dots, T$. The value of T will be defined later. Let $\ell = 3\lceil \log_2 n \rceil$. We classify the n elements into 2ℓ classes based on their initial values $y(i)$ as follows.

$$U_k := \begin{cases} \{i \in [n] : 2^{-k-1} < y(i) \leq 2^{-k}\} & \text{if } 1 \leq k \leq \ell - 1 \\ \{i \in [n] : y(i) \leq 2^{-\ell}\} & \text{if } k = \ell. \end{cases}$$

$$V_k := \begin{cases} \{i \in [n] : 2^{-k-1} < 1 - y(i) \leq 2^{-k}\} & \text{if } 1 \leq k \leq \ell - 1 \\ \{i \in [n] : 1 - y(i) \leq 2^{-\ell}\} & \text{if } k = \ell. \end{cases}$$

Note that the U_k 's partition elements of value (in y) between 0 and $\frac{1}{2}$ and the V_k 's form a symmetric partition of elements valued between $\frac{1}{2}$ and 1. This partition does not change over time, even though the value of variables might change. We define the “scale” of each element as:

$$s_i := 2^{-k}, \quad \forall i \in U_k \cup V_k, \quad \forall k \in [\ell].$$

Define $W_j(s) = \sum_{i=1}^n a_{ji}^2 \cdot s_i^2$ for each $j \in [m]$. Note that $W_j(s) \geq W_j(y)$ and

$$W_j(s) - 4 \cdot W_j(y) \leq \sum_{i=1}^n a_{ji}^2 \cdot \frac{1}{n^6} = \frac{\|a_j\|^2}{n^6}.$$

So $\sqrt{W_j(y)} \leq \sqrt{W_j(s)} \leq 2\sqrt{W_j(y)} + \frac{\|a_j\|}{n^3}$. Our algorithm will find a solution y' with $\Omega(n)$ integral variables such that:

$$|\langle y' - y, a_j \rangle| \leq \lambda_j \cdot \sqrt{W_j(s)} + \frac{1}{n^3} \cdot \|a_j\|, \quad \forall j \in [m].$$

This suffices to prove Theorem 5 as

$$\lambda_j \cdot \sqrt{W_j(s)} + \frac{1}{n^3} \cdot \|a_j\| \leq 2\lambda_j \cdot \sqrt{W_j(y)} + \left(\frac{1}{n^3} + \frac{\lambda_j}{n^3} \right) \cdot \|a_j\| \leq 2\lambda_j \cdot \sqrt{W_j(y)} + \frac{\|a_j\|}{n^2}.$$

Consider the polytope \mathcal{Q} of points $x \in \mathbb{R}^n$ satisfying the following constraints.

$$x \in P(\mathcal{M}), \tag{8}$$

$$|\langle x - y, a_j \rangle| \leq \lambda_j \cdot \sqrt{W_j(s)} + \frac{1}{n^3} \cdot \|a_j\| \quad \forall j \in [m], \tag{9}$$

$$\sum_{i \in U_k} x_i = \sum_{i \in U_k} y_i \quad \forall k \in [\ell], \tag{10}$$

$$\sum_{i \in V_k} x_i = \sum_{i \in V_k} y_i \quad \forall k \in [\ell], \tag{11}$$

$$0 \leq x_i \leq \min\{\alpha \cdot 2^{-k}, 1\} \quad \forall i \in U_k, \forall k \in [\ell], \tag{12}$$

$$0 \leq 1 - x_i \leq \min\{\alpha \cdot 2^{-k}, 1\} \quad \forall i \in V_k, \forall k \in [\ell]. \tag{13}$$

Above $\alpha = 40$ is a constant whose choice will be clear later. The algorithm will maintain the invariant that at any time $t \in [T]$, the solution X_t lies in \mathcal{Q} . In particular the constraint (8) requires that X_t stays in the matroid polytope. Constraint (9) controls the violation of the side constraints over all time steps. The last two constraints (12)-(13) enforce that variables in U_k (and symmetrically V_k) do not deviate far beyond their original scale of 2^{-k} . The constraints (10) and (11) ensure that throughout the algorithm, the total value of elements in U_k (and V_k) stay equal to their initial sum (in y). These constraints will play a crucial role in arguing that the algorithm finds a partial coloring. Note that there are only 2ℓ such constraints.

In order to deal with complexity issues, we will assume (without loss of generality, by scaling) that all entries in the constraints describing \mathcal{Q} are integers bounded by some value B . Our algorithm will then run in time polynomial in n, m and $\log_2 B$, given an independent set oracle for the matroid \mathcal{M} . Also, our algorithm will only deal with points having rational entries of small ‘‘size’’. Recall that the size of a rational number is the number of bits needed to represent it, i.e. the size of p/q (where $p, q \in \mathbb{Z}$) is $\log_2 |p| + \log_2 |q|$.

The Algorithm: Let $\gamma = n^{-6}$ and $T = K/\gamma^2$ where $K := 10\alpha^2$. The algorithm starts with solution $X_0 = y \in \mathcal{Q}$, and does the following at each time step $t = 0, 1, \dots, T$:

1. Consider the set of constraints of \mathcal{Q} that are tight at the point $x = X_t$, and define the following sets based on this.

(a) Let \mathcal{C}_t^{var} be the set of tight variable constraints among (12)-(13). This consists of:

- i. $i \in U_k$ (for any k) with $X_t(i) = 0$ or $X_t(i) = \min\{\alpha \cdot 2^{-k}, 1\}$; and
- ii. $i \in V_k$ (for any k) with $X_t(i) = 1$ or $X_t(i) = \max\{1 - \alpha \cdot 2^{-k}, 0\}$.

(b) Let \mathcal{C}_t^{side} be the set of tight side constraints from (9), i.e. those $j \in [m]$ with

$$|\langle X^t - y, a_j \rangle| = \lambda_j \cdot \sqrt{W_j(s)} + \frac{1}{n^3} \|a_j\|.$$

(c) Let \mathcal{C}_t^{part} denote the set of the 2ℓ equality constraints (10)-(11).

(d) Let \mathcal{C}_t^{rank} be a maximal linearly independent set of tight rank constraints from (8). As usual, a set of constraints is said to be linearly independent if the corresponding coefficient vectors are linearly independent. Since \mathcal{C}_t^{rank} is maximal, every tight rank constraint is a linear combination of constraints in \mathcal{C}_t^{rank} . By Claim 2, $|\mathcal{C}_t^{rank}| \leq n/2$.

2. Let \mathcal{V}_t denote the subspace orthogonal to all the constraints in \mathcal{C}_t^{var} , \mathcal{C}_t^{side} , \mathcal{C}_t^{part} and \mathcal{C}_t^{rank} . Let D be an $n \times n$ diagonal matrix with entries $d_{ii} = 1/s_i$, and let \mathcal{V}'_t be the subspace $\mathcal{V}'_t = \{Dv : v \in \mathcal{V}_t\}$. As D is invertible, $\dim(\mathcal{V}'_t) = \dim(\mathcal{V}_t)$.
3. Let $\{b_1, \dots, b_k\}$ be an almost orthonormal basis of \mathcal{V}'_t given by Fact 2. Note that all entries in these vectors are rationals of size $O(n^2 \log B)$.
4. Let G_t be a random direction defined as $G_t := \sum_{h=1}^k g_h b_h$ where the g_h are independent $\{-1, +1\}$ Bernoulli random variables.
5. Let $\overline{G}_t := D^{-1}G_t$. As $G_t \in \mathcal{V}'_t$, it must be that $G_t = Dv$ for some $v \in \mathcal{V}_t$ and thus $\overline{G}_t = D^{-1}G_t \in \mathcal{V}_t$. Note that all entries in \overline{G}_t are rationals of size $O(n^3 \log B)$.
6. Set $Y_t = X_t + \gamma \cdot \overline{G}_t$.
 - (a) If $Y_t \in \mathcal{Q}$ then $X_{t+1} \leftarrow Y_t$ and continue to the next iteration.
 - (b) Else $X_{t+1} \leftarrow$ the point in \mathcal{Q} that lies on the line segment (X_t, Y_t) and is closest to Y_t . This can be found by binary search and testing membership in the matroid polytope. By Claim 1 it follows that the number of steps in the binary search is at most $O(n \log B)$.

This completes the description of the algorithm. We actually do not need to compute the tight constraints from scratch in each iteration. We start the algorithm off with a strictly feasible solution $y \in \mathcal{Q}$ which does not have any tight constraint other than (10)-(11). Then, the only place a new constraint gets tight is Step 6b: at this point, we add the new constraint to the appropriate set among \mathcal{C}_t^{var} , \mathcal{C}_t^{side} and \mathcal{C}_t^{rank} and continue.

In order to keep the analysis clean and convey the main ideas, we will assume that the basis $\{b_1, \dots, b_k\}$ in Step 3 is exactly orthonormal. When the basis is “almost orthonormal” as given in Fact 2, the additional error incurred is negligible.

Running Time. Since the number of iterations is polynomial, we only need to show that each of the steps in any single iteration can be implemented in polynomial time. The only step that requires justification is 6b, which is shown in Claim 1. Moreover, we need to ensure that all points considered in the algorithm have rational coefficients of polynomial size. This is done by a rounding procedure (see Fact 1) that given an arbitrary point, finds a nearby rational point of size $O(n^2 \log B)$. Since the number of steps in the algorithm is polynomial, the total error incurred by such rounding steps is small.

Claim 1. *The number of binary search iterations performed in Step 6b is $O(n^4 \log B)$.*

Proof. To reduce notation let $a = X_t$, $d = \gamma \overline{G}_t$ and $Y(\mu) := a + \mu \cdot d \in \mathbb{R}^n$ for any $\mu \in \mathbb{R}$. Recall that Step 6b involves finding the maximum value of μ such that point $Y(\mu) \in \mathcal{Q}$.

By the rounding procedure (Fact 1) we know that a has rational entries of size $O(n^2 \log B)$.

We now show that the direction d has rational entries of size $O(n^3 \log B)$. This is because (i) the basis vectors $\{b_1, \dots, b_k\}$ have rational entries of size $O(n^2 \log B)$ by Fact 2, (ii) $G_t = \sum_{h=1}^k g_h \cdot b_h$ (where each $g_h = \pm 1$) has rational entries of size $O(n^3 \log B)$ and (iii) $\overline{G}_t = D^{-1}G_t$ where D^{-1} is a diagonal matrix with rational entries of size $O(n \log B)$.

Next, observe that for any constraint $\langle a', x \rangle \leq \beta$ in \mathcal{Q} , the point of intersection of the hyperplane $\langle a', x \rangle = \beta$ with line $\{Y(\mu) : \mu \in \mathbb{R}\}$ is $\mu = \frac{\beta - \langle a', a \rangle}{\langle a', d \rangle}$ which is a rational of size at most $\sigma = O(n^4 \log B)$ as a', a, d, β all have rational entries of size $O(n^3 \log B)$. Let $\epsilon = 2^{-2\sigma}$ be a value such that the difference between any two distinct rationals of size at most σ is more than ϵ .

In Step 6b, we start the binary search with the interval $[0, 1]$ for μ where $Y(0) \in \mathcal{Q}$ and $Y(1) \notin \mathcal{Q}$. We perform this binary search until the interval width falls below ϵ , which requires

$\log_2 \frac{1}{\epsilon} = O(n^4 \log B)$ iterations. At the end, we have two values $\mu_0 < \mu_1$ with $\mu_1 - \mu_0 < \epsilon$ such that $Y(\mu_0) \in \mathcal{Q}$ and $Y(\mu_1) \notin \mathcal{Q}$. Moreover, we obtain a constraint $\langle a', x \rangle \leq \beta$ in \mathcal{Q} that is not satisfied by $Y(\mu_1)$. We set μ' to be the (unique) value such that $Y(\mu')$ satisfies this constraint at equality, and set $X_{t+1} = Y(\mu')$. Note that $\mu_0 \leq \mu' < \mu_1$. To see that $Y(\mu') \in \mathcal{Q}$, suppose (for contradiction) that some constraint in \mathcal{Q} is not satisfied at $Y(\mu')$; then the point of intersection of line $\{Y(\mu) : \mu \in \mathbb{R}\}$ with this constraint must be at $\mu \in [\mu_0, \mu']$ which (by the choice of ϵ) can not be a rational of size at most σ — a contradiction. \square

Analysis. The analysis involves proving the following main lemma.

Lemma 1. *With constant probability, the final solution X_T has $|\mathcal{C}_T^{var}| \geq \frac{n}{20}$.*

We first show how this implies Theorem 5.

Proof of Theorem 5 from Lemma 1: The algorithm outputs the solution $y' := X_T$. By design the algorithm ensures that $X_T \in \mathcal{Q}$, and thus $X_T \in P(\mathcal{M})$ and it satisfies the error bounds (9) on the side constraints. It remains to show that $\Omega(n)$ variables in X_T must be integer valued whenever $|\mathcal{C}_T^{var}| \geq \frac{n}{20}$. For each $k \in [\ell]$ define $u_k := |\{i \in U_k : X_T(i) = \alpha \cdot 2^{-k}\}|$ and $v_k := |\{i \in V_k : X_T(i) = 1 - \alpha \cdot 2^{-k}\}|$. By the equality constraints (10) for U_k , it follows that

$$u_k \cdot \alpha \cdot 2^{-k} \leq \sum_{i \in U_k} X_T(i) = X_T(U_k) = y(U_k) \leq |U_k| \cdot 2^{-k}.$$

This gives that $u_k \leq \frac{1}{\alpha} |U_k|$. Similarly, $v_k \leq \frac{1}{\alpha} |V_k|$. This implies that $\sum_{k=1}^{\ell} (u_k + v_k) \leq n/\alpha$. As the tight variables in \mathcal{C}_t^{var} have values either 0 or 1 or $\alpha \cdot 2^{-k}$ or $1 - \alpha \cdot 2^{-k}$, it follows that the number of $\{0, 1\}$ variables is at least

$$|\mathcal{C}_t^{var}| - \sum_{k=1}^{\ell} (u_k + v_k) \geq \left(|\mathcal{C}_t^{var}| - \frac{n}{\alpha} \right) \geq \left(\frac{1}{20} - \frac{1}{\alpha} \right) n$$

which is at least $n/40$ by choosing $\alpha = 40$. \square

In the rest of this section we prove Lemma 1.

Claim 2. *Given any $x \in P(\mathcal{M})$ with $0 < x < 1$, the maximum number of tight linearly independent rank constraints is $n/2$.*

Proof. Recall that a tight constraint in $P(\mathcal{M})$ is any subset $T \subseteq V$ with $\sum_{i \in T} x_i = r(T)$. The claim follows from the known property (see eg. [Sch03]) that for any $x \in P(\mathcal{M})$ there is a linearly independent collection \mathcal{C} of tight constraints such that (i) \mathcal{C} spans all tight constraints and (ii) \mathcal{C} forms a chain family. Since all right-hand-sides are integer and each variable is strictly between 0 and 1, it follows that $|\mathcal{C}| \leq \frac{n}{2}$. \square

Claim 3. *The truncation Step 6b occurs at most n times.*

Proof. We will show that whenever Step 6b occurs (i.e. the random move gets truncated) the dimension $\dim(\mathcal{V}_{t+1})$ decreases by at least 1, i.e. $\dim(\mathcal{V}_{t+1}) \leq \dim(\mathcal{V}_t) - 1$. As the maximum dimension is n this would imply the claim.

Let \mathcal{E}_t denote the subspace spanned by all the tight constraints of $X_t \in \mathcal{Q}$; Recall that $\mathcal{V}_t = \mathcal{E}_t^\perp$ is the subspace orthogonal to \mathcal{E}_t , and thus $\dim(\mathcal{E}_t) = n - \dim(\mathcal{V}_t)$. We also have $\mathcal{E}_0 \subseteq \mathcal{E}_1 \subseteq \dots \subseteq \mathcal{E}_T$. Suppose that Step 6b occurs in iteration t . Then we have $X_t \in \mathcal{Q}$, $Y_t \notin \mathcal{Q}$ and $Y_t - X_t \in \mathcal{V}_t$.

Moreover $X_{t+1} = X_t + \epsilon(Y_t - X_t) \in \mathcal{Q}$ where $\epsilon \in [0, 1)$ is such that $X_t + \epsilon'(Y_t - X_t) \notin \mathcal{Q}$ for all $\epsilon' > \epsilon$. So there is some constraint $\langle a', x \rangle \leq \beta$ in \mathcal{Q} with:

$$\langle a', X_t \rangle \leq \beta, \quad \langle a', X_{t+1} \rangle = \beta \quad \text{and} \quad \langle a', Y_t \rangle > \beta.$$

Since this constraint satisfies $\langle a', Y_t - X_t \rangle > 0$ and $Y_t - X_t \in \mathcal{V}_t$, we have $a' \notin \mathcal{E}_t$. As a' is added to \mathcal{E}_{t+1} , we have $\dim(\mathcal{E}_{t+1}) \geq 1 + \dim(\mathcal{E}_t)$. This proves the desired property and the claim. \square

The statements of the following two lemmas are similar to those in [LM12], but the proofs require additional work since our random walk is different. The first lemma shows that the expected number of tight side constraints at the end of the algorithm is not too high, and the second lemma shows that the expected number of tight variable constraints is large.

Lemma 2. $\mathbb{E}[|\mathcal{C}_T^{\text{side}}|] < \frac{n}{4}$.

Proof. Note that $X_T - y = \gamma \sum_{t=0}^T \bar{G}_t + \sum_{q=1}^n \Delta_{t(q)}$ where Δ s correspond to the truncation incurred during the iterations $t = t(1), \dots, t(n)$ for which Step 6b applies (by Claim 3 there are at most n such iterations). Moreover for each q , $\Delta_{t(q)} = \delta \cdot \bar{G}_{t(q)}$ for some δ with $0 < |\delta| < \gamma$.

If $j \in \mathcal{C}_T^{\text{side}}$, then $|\langle X_T - y, a_j \rangle| = \lambda_j \sqrt{W_j(s)} + \frac{1}{n^3} \cdot \|a_j\|$. We have

$$|\langle X_T - y, a_j \rangle| \leq |\gamma \sum_{t=0}^T \langle \bar{G}_t, a_j \rangle| + \sum_{q=1}^n \gamma |\langle \bar{G}_{a(q)}, a_j \rangle| \leq |\gamma \sum_{t=0}^T \langle \bar{G}_t, a_j \rangle| + n\gamma \cdot \max_{t=0}^T |\langle \bar{G}_t, a_j \rangle|.$$

Note that at any iteration t ,

$$|\langle \bar{G}_t, a_j \rangle| = |\langle D^{-1}G_t, a_j \rangle| \leq |\langle G_t, a_j \rangle| \leq \sum_{h=1}^k |\langle b_h, a_j \rangle| \leq n\|a_j\|.$$

The first inequality above uses that D^{-1} is a diagonal matrix with entries at most one, the second inequality is by definition of G_t where $\{b_h\}$ is an orthonormal basis of \mathcal{V}'_t , and the last inequality uses that each b_h is a unit vector. As $\gamma = n^{-6}$, we have $n\gamma \cdot \max_{t=0}^T |\langle \bar{G}_t, a_j \rangle| \leq \|a_j\|/n^4$. So it follows that if $j \in \mathcal{C}_T^{\text{side}}$, then we must have:

$$|\gamma \sum_{t=0}^T \langle \bar{G}_t, a_j \rangle| \geq \lambda_j \sqrt{W_j(s)}.$$

In order to bound the probability of this event, we consider the sequence $\{Z_t\}$ where $Z_t = \langle \bar{G}_t, a_j \rangle$, and note the following useful facts.

Observation 1. *The sequence $\{Z_t\}$ forms a martingale satisfying:*

1. $\mathbb{E}[Z_t | Z_{t-1}, \dots, Z_0] = 0$ for all t .
2. $|Z_t| \leq n\|a_j\|$ whp for all t .
3. $\mathbb{E}[Z_t^2 | Z_{t-1}, \dots, Z_0] \leq \sum_{i=1}^n s_i^2 \cdot a_{j_i}^2 = W_j(s)$ for all t .

Proof. As $\bar{G}_t = \sum_{h=1}^k g_h \cdot b_h$ where each $\mathbb{E}[g_h] = 0$, we have $\mathbb{E}[\bar{G}_t | \bar{G}_0, \dots, \bar{G}_{t-1}] = \mathbf{0}$. Note that \bar{G}_t is not independent of $\bar{G}_0, \dots, \bar{G}_{t-1}$, as these choices determine the subspace where \bar{G}_t lies. So $\{Z_t\}$ forms a martingale sequence with the first property.

For the remaining two properties, we fix $j \in [m]$ and t and condition on Z_0, \dots, Z_{t-1} . To reduce notation we drop all subscripts: so $a = a_j$, $G = G_t$, $\mathcal{V}' = \mathcal{V}'_t$ and $Z = Z_t$.

Let $\{b_r\}$ denote an orthonormal basis for the linear subspace \mathcal{V}' . Then $G = \sum_r g_r \cdot b_r$ where each g_r is iid ± 1 with probability half. As $\bar{G} = D^{-1}G$, we have $Z = \langle \bar{G}, a \rangle = \sum_r \langle D^{-1}b_r, a \rangle g_r = \sum_r \langle D^{-1}a, b_r \rangle g_r$. So, we can bound

$$|Z| \leq \sum_r |\langle D^{-1}a, b_r \rangle| \cdot |g_r| \leq \|D^{-1}a\| \sum_r |g_r| \leq n\|a\|.$$

The first inequality follows from the triangle inequality, the second by Cauchy-Schwartz and as b_r is a unit-vector, and the third follows as D^{-1} is a diagonal matrix with entries at most one. This proves property 2.

Finally, $\mathbb{E}[Z^2] = \sum_r \langle D^{-1}a, b_r \rangle^2 \mathbb{E}[g_r^2] = \sum_r \langle D^{-1}a, b_r \rangle^2 \leq \|D^{-1}a\|^2$, where the last step follows as $\{b_r\}$ is an orthonormal basis for a subspace of \mathbb{R}^n . This proves property 3. \square

Using a martingale concentration inequality, we obtain:

Claim 4. $\Pr \left[|\gamma \sum_{t=0}^T \langle \bar{G}_t, a_j \rangle| \geq \lambda_j \sqrt{W_j(s)} \right] = \Pr \left[\left| \sum_{t=0}^T Z_t \right| \geq \frac{\lambda_j}{\gamma} \sqrt{W_j(s)} \right] \leq 2 \cdot \exp(-\lambda_j^2/3K).$

Proof. The first equality is by definition of the Z_t s. We now use the following concentration inequality:

Theorem 6 (Freedman [Fre75] (Theorem 1.6)). *Consider a real-valued martingale sequence $\{Z_t\}_{t \geq 0}$ such that $Z_0 = 0$, $\mathbb{E}[Z_t | Z_{t-1}, \dots, Z_0] = 0$ for all t , and $|Z_t| \leq M$ almost surely for all t . Let $W_t = \sum_{j=0}^t \mathbb{E}[Z_j^2 | Z_{j-1}, Z_{j-2}, \dots, Z_0]$ for all $t \geq 1$. Then for all $\ell \geq 0$ and $\sigma^2 > 0$, and any stopping time τ we have*

$$\Pr \left[\left| \sum_{j=0}^{\tau} Z_j \right| \geq \ell \text{ and } W_{\tau} \leq \sigma^2 \right] \leq 2 \exp \left(-\frac{\ell^2/2}{\sigma^2 + M\ell/3} \right)$$

We apply this with $M = n\|a_j\|$, $\ell = \frac{\lambda_j}{\gamma} \sqrt{W_j(s)}$, $\sigma^2 = T \cdot W_j(s)$ and $\tau = T$. Note that

$$\frac{\ell^2}{2\sigma^2 + \frac{2}{3}M\ell} = \frac{\lambda_j^2}{2\gamma^2 T + \frac{2}{3}\gamma n \|a_j\| \lambda_j / \sqrt{W_j(s)}} \geq \frac{\lambda_j^2}{2\gamma^2 T + 1},$$

where the last inequality uses $W_j(s) \geq \|a_j\|^2/n^6$, $\lambda_j \leq n$ and $\gamma = n^{-6}$. Thus

$$\Pr \left[\left| \gamma \sum_{t=0}^T \langle \bar{G}_t, a_j \rangle \right| \geq \lambda_j \sqrt{W_j(s)} \right] \leq 2 \exp \left(\frac{-\lambda_j^2}{2\gamma^2 T + 1} \right) \leq 2 \cdot \exp(-\lambda_j^2/3K).$$

The last inequality uses $T = K/\gamma^2$ and $K \geq 1$. This completes the proof of the claim. \square

By the above claim, we have $\mathbb{E}[|\mathcal{C}_T^{side}|] < 2 \sum_{j=1}^m \exp(-\lambda_j^2/(30\alpha^2)) < 0.25n$. This completes the proof of Lemma 2. \square

We now prove that in expectation, at least $0.1n$ variables become tight at the end of the algorithm. This immediately implies Lemma 1.

Lemma 3. $\mathbb{E}[|\mathcal{C}_T^{var}|] \geq 0.1n$.

Proof. Define the following potential function, which will measure the progress of the algorithm toward the variables becoming tight.

$$\Phi(x) := \sum_{k=1}^{\ell} 2^{2k} \cdot \left(\sum_{i \in U_k} x(i)^2 + \sum_{i \in V_k} (1 - x(i))^2 \right), \quad \forall x \in \mathcal{Q}.$$

Note that since $X_T \in \mathcal{Q}$, we have $X_T(i) \leq \alpha \cdot 2^{-k}$ for $i \in U_k$ and $1 - X_T(i) \leq \alpha \cdot 2^{-k}$ for $i \in V_k$. So $\Phi(X_T) \leq \alpha^2 \cdot n$. We also define the “incremental function” for any $x \in \mathcal{Q}$ and $g \in \mathbb{R}^n$, $f(x, g) := \Phi(x + \gamma D^{-1}g) - \Phi(x)$. Recall that D^{-1} is the $n \times n$ diagonal matrix with entries (s_1, \dots, s_n) where $s_i = 2^{-k}$ for $i \in U_k \cup V_k$. So

$$\begin{aligned} f(x, g) &= \gamma^2 \sum_{i=1}^n g(i)^2 + 2 \sum_{k=1}^{\ell} 2^{2k} \cdot \left(\sum_{i \in U_k} x(i) \gamma s_i \cdot g(i) - \sum_{i \in V_k} (1 - x(i)) \gamma s_i \cdot g(i) \right) \\ &= \gamma^2 \sum_{i=1}^n g(i)^2 + 2\gamma \sum_{k=1}^{\ell} \left(\sum_{i \in U_k} \frac{x(i)g(i)}{s_i} - \sum_{i \in V_k} \frac{(1 - x(i))g(i)}{s_i} \right) \end{aligned}$$

Suppose the algorithm was modified to never have the truncation step 6b, then in any iteration t , the increase $\Phi(Y_t) - \Phi(X_t) = f(X_t, G_t)$ where G_t is the random direction chosen in \mathcal{V}'_t . The following is by simple calculation.

$$\begin{aligned} f(X_t, G_t) - f(X_t, \delta G_t) &= \gamma^2(1 - \delta^2) \|G_t\|_2^2 + 2\gamma(1 - \delta) \sum_{k=1}^{\ell} \left(\sum_{i \in U_k} \frac{X_t(i)}{s_i} \cdot G_t(i) - \sum_{i \in V_k} \frac{1 - X_t(i)}{s_i} \cdot G_t(i) \right) \\ &\leq \gamma^2(1 - \delta^2) \|G_t\|_2^2 + 2\alpha\gamma(1 - \delta) \sum_{i=1}^n |G_t(i)| \leq \gamma^2 \|G_t\|_2^2 + 2\gamma\alpha \|G_t\|_1 \\ &\leq \gamma^2 n + 2\gamma\alpha n^{3/2} \leq \frac{1}{n} \end{aligned} \tag{14}$$

The first inequality in (14) uses the fact that G_t is the sum of orthogonal unit vectors, and the second inequality uses $\gamma = n^{-6}$ and $\alpha = O(1)$.

This implies that

$$\begin{aligned} \Phi(X_T) - \Phi(X_0) &= \sum_{t=0}^T f(X_t, \delta_t G_t) \geq \sum_{t=0}^T f(X_t, G_t) - \frac{1}{n} \sum_{t=0}^T \mathbf{1}[\text{step 6b occurs in iteration } t] \\ &\geq \sum_{t=0}^T f(X_t, G_t) - 1 \quad (\text{by Claim 3}) \end{aligned} \tag{15}$$

Claim 5. $\mathbb{E}[\Phi(X_T)] - \Phi(y) \geq \gamma^2 T \cdot \mathbb{E}[\dim(\mathcal{V}_T)] - 1$.

Proof. From (15) we have:

$$\mathbb{E}[\Phi(X_T)] - \Phi(X_0) \geq \sum_{t=0}^T \mathbb{E}[f(X_t, G_t)] - 1. \tag{16}$$

In any iteration t , as $G_t = \sum_{h=1}^k g_h b_h$ where $\{b_h\}$ is an orthonormal basis for \mathcal{V}'_t and $g_h = \pm 1$,

$$\mathbb{E}[f(X_t, G_t)] = \gamma^2 \sum_{i=1}^n \mathbb{E}[G_t(i)^2] = \gamma^2 \sum_{h=1}^k \|b_h\|^2 = \gamma^2 k = \gamma^2 \mathbb{E}[\dim(\mathcal{V}'_t)] = \gamma^2 \mathbb{E}[\dim(\mathcal{V}_t)].$$

Moreover, because $\mathcal{V}_0 \supseteq \mathcal{V}_1 \supseteq \dots \mathcal{V}_T$, we have $\mathbb{E}[\dim(\mathcal{V}_t)] \geq \mathbb{E}[\dim(\mathcal{V}_T)]$. So

$$\sum_{t=0}^T \mathbb{E}[f(X_t, G_t)] \geq \gamma^2 T \cdot \mathbb{E}[\dim(\mathcal{V}_T)]. \quad (17)$$

Combining (16) and (17), we complete the proof of Claim 5. \square

By Claim 2 and the fact that $|\mathcal{C}_T^{part}| = 2\ell$, we have

$$\begin{aligned} \dim(\mathcal{V}_T) &\geq n - \dim(\mathcal{C}_T^{var}) - \dim(\mathcal{C}_T^{side}) - \dim(\mathcal{C}_T^{rank}) - \dim(\mathcal{C}_T^{part}) \\ &\geq \frac{n}{2} - 2\ell - \dim(\mathcal{C}_T^{var}) - \dim(\mathcal{C}_T^{side}) \end{aligned}$$

Taking expectations and by Claim 2, this gives

$$\mathbb{E}[\dim(\mathcal{V}_T)] \geq \frac{n}{4} - 2\ell - \dim(\mathcal{C}_T^{var}) \quad (18)$$

Using $\Phi(X_T) \leq \alpha^2 n$ and Claim 5, we obtain:

$$\alpha^2 n \geq \mathbb{E}[\Phi_T] \geq \gamma^2 T \cdot \left(\frac{n}{4} - 2\ell - \mathbb{E}[\dim(\mathcal{C}_T^{var})] \right) - 1.$$

Rearranging and using $T = K/\gamma^2$, $K = 10\alpha^2$ and $\ell = \log n$ gives that

$$\mathbb{E}[\dim(\mathcal{C}_T^{var})] \geq \frac{n}{4} - \frac{\alpha^2 n}{K} - 2\ell - \frac{1}{K} \geq 0.1n,$$

where we used $K = 10\alpha^2$, $\alpha = 40$ and $\ell = O(\log n)$. This completes the proof of Lemma 3. \square

3 Applications

3.1 Linear System Rounding with Violations

Consider a 0–1 integer program on n variables where each constraint $j \in [m]$ corresponds to some subset $S_j \subseteq [n]$ of the variables having total value $b_j \in \mathbb{Z}_+$. That is,

$$P = \left\{ x \in \{0, 1\}^n \quad : \quad \sum_{i \in S_j} x_i = b_j, \quad \forall j \in [m] \right\}.$$

Theorem 7. *There is a randomized polynomial time algorithm that given any fractional solution satisfying the constraints in P , finds an integer solution $x \in \{0, 1\}^n$ where for each $j \in [m]$,*

$$|x(S_j) - b_j| \leq O(1) \cdot \min \left\{ \sqrt{j}, \sqrt{n \log(m/n)}, \sqrt{\log m \log n \cdot b_j} + \log m \log n, \sqrt{\Delta \log n} \right\}.$$

Above $\Delta = \max_{i=1}^n |\{j \in [m] : i \in S_j\}|$ is the maximum number of constraints that any variable appears in.

Proof. Let $y \in [0, 1]^n$ be a fractional solution with $\sum_{i \in S_j} y_i = b_j$ for all $j \in [m]$. The algorithm in Theorem 7 uses Theorem 4 iteratively to obtain the integral solution x .

In each iteration, we start with a fractional solution y' with $f \leq n$ fractional variables and set the parameters λ_j suitably so that $\sum_{j=1}^m e^{-\lambda_j^2/K_0} \leq \frac{f}{16}$. That is, the condition in Theorem 4 is satisfied. Note that $W_j(y') = \sum_{i \in S_j} (y'_i)^2 \leq y'(S_j)$ and $W_j(y') \leq f$. Now, by applying Theorem 4, we would obtain a new fractional solution y'' such that:

- For each $j \in [m]$, $|y''(S_j) - y'(S_j)| \leq \lambda_j \sqrt{W_j(y')} + \frac{1}{n} \leq O(\lambda_j) \cdot \sqrt{f}$.
- The number of fractional variables in y'' is at most $\frac{f}{K}$ for some constant $K > 1$.

Therefore, after $\frac{\log n}{\log K} = O(\log n)$ iterations we obtain a solution with $O(1)$ fractional variables. Setting these fractional variables arbitrarily to 0–1 values, we obtain an integral solution x .

Let us partition the constraints into sets M_1, M_2, M_3 and M_4 based on which of the four terms in Theorem 7 is minimum. That is, $M_1 \subseteq [m]$ consists of constraints $j \in [m]$ where \sqrt{j} is smaller than the other three terms; M_2, M_3, M_4 are defined similarly. Below we show how to set the parameters λ_j and bound the constraint violations for these parts separately.

Error bound of $\min\{\sqrt{j}, \sqrt{n \log(m/n)}\}$ for $j \in M_1 \cup M_2$. In any iteration with $f \leq n$ fractional variables, we set the parameters λ_j s in Theorem 4 as follows:

$$\lambda_j = \begin{cases} 0 & \text{if } j < c_1 f \\ \sqrt{c_2 \log \frac{j}{c_1 f}} & \text{if } j \geq c_1 f \end{cases}$$

Here c_1 and c_2 are constants that will be fixed later. Note that

$$\sum_{j \in M_1 \cup M_2} e^{-\lambda_j^2/K_0} \leq c_1 f + \sum_{j \geq c_1 f} e^{-\frac{c_2}{K_0} \log \frac{j}{c_1 f}} \leq c_1 f + \sum_{i \geq 0} 2^i c_1 f \cdot e^{-i c_2/K_0} \leq c_1 f + c_1 f \sum_{i \geq 0} 2^{-i} \leq 3c_1 f,$$

which is at most $f/48$ for $c_1 < 1/150$. The second inequality above is obtained by bucketing the j s into intervals of the form $[2^i \cdot c_1 f, 2^{i+1} \cdot c_1 f]$. The third inequality uses $c_2 \geq 2K_0$.

We now bound the error incurred.

1. Consider first a constraint $j \leq n$. Note that λ_j stays zero until the number of fractional variables f drops below j/c_1 . So we can bound $|x(S_j) - b_j|$ by:

$$\sum_{i \geq 0} \sqrt{c_2 \frac{j}{c_1 K^i} \cdot \log K^i} \leq O(\sqrt{j}) \sum_{i \geq 0} \sqrt{i} K^{-i/2} = O(\sqrt{j}),$$

where i indexes the iterations of the algorithm after f drops below j/c_1 for the first time.

2. Now consider a constraint $j > n$. Similarly, we bound $|x(S_j) - b_j|$ by:

$$\sum_{i \geq 0} \sqrt{c_2 \frac{n}{K^i} \cdot \log \left(\frac{j}{c_1 n} K^i \right)} \leq O(\sqrt{n \log(j/n)}) \sum_{i \geq 0} \sqrt{i} K^{-i/2} = O(\sqrt{n \log(j/n)}).$$

Here i indexes the number of iterations of the algorithm from its start.

Error bound of $\sqrt{L \cdot b_j} + L$ for $j \in M_3$, where $L = \Theta(\log m \log n)$. Note that the additive term in this expression is at least L . If any $b_j < L$ then we increase it to L (and add dummy elements to S_j and ensure $y(S_j) = L$); this only affects the error term by a constant factor as $L \leq \sqrt{L \cdot b_j} + L \leq 2L$. So in the following we assume that $\min_j b_j \geq L$.

Here we set $\lambda_j = \infty$ in all iterations, which satisfies $\sum_{j \in M_3} e^{-\lambda_j^2/K_0} = 0$.

The analysis of the error incurred is similar to that in Lemma 2 and we only sketch the details; the main difference is that we analyze the deviation in a combined manner over all $O(\log n)$ iterations. Fix any constraint $j \in [m]$. If we ignore the error due to the truncation steps over all iterations⁴ then we can write $|x(S_j) - b_j| = |\sum_{t=0}^P \gamma Z_t|$ where $\gamma = n^{-6}$ and $Z_t = \langle \bar{G}_t, \mathbf{1}_{S_j} \rangle$; recall

⁴This can be bounded by $o(1)$ exactly as in Event 2 of Lemma 2.

that each $\bar{G}_t = D^{-1}G_t$ for random direction G_t as in Step 4 of the algorithm in Section 2. Here $P = O(\log n/\gamma^2)$ since there are $O(\log n)$ iterations and $O(1/\gamma^2)$ steps in each iteration. We will use the concentration inequality in Theorem 6 with martingale $\{Z_t\}_{t \geq 0}$ and stopping time τ being the first time t' where $|\sum_{t=0}^{t'} Z_t| > \frac{1}{\gamma} \sqrt{Lb_j}$. Then it follows that at any step t' before stopping, the current solution y' satisfies $y'(S_j) - y(S_j) = \gamma \sum_{t=0}^{t'} Z_t \leq \sqrt{Lb_j} \leq b_j$ (using the assumption $b_j \geq L$), i.e. $y'(S_j) \leq 2b_j$. Now we can bound $W_\tau \leq P \cdot O(b_j) = O(\log n/\gamma^2) \cdot b_j$. Using Theorem 6 with $\ell = \sqrt{Lb_j}/\gamma$, we obtain:

$$\Pr \left[\left| \gamma \sum_{t=0}^{\tau} Z_t \right| \geq \sqrt{Lb_j} \right] \leq 2 \exp \left(\frac{-Lb_j}{O(\log n)b_j} \right) \leq \frac{1}{m^2},$$

by choosing a large enough constant in $L = O(\log m \log n)$. It follows that with probability at least $1 - m^{-2}$, we have $\tau = P$ and $|x(S_j) - b_j| = |\sum_{t=0}^P \gamma Z_t| \leq \sqrt{L \cdot b_j}$. Finally, taking a union bound over $|M_3| \leq m$ such events, we obtain that with high probability, $|x(S_j) - b_j| \leq \sqrt{L \cdot b_j}$ for all $j \in M_3$.

Error bound of $\sqrt{\Delta} \log n$ for $j \in M_4$. Here we set $\lambda_j = \sqrt{K_1 \Delta} / \sqrt{|S_j|}$ in all iterations, where K_1 is a constant to be fixed later. We first bound $\sum_{j \in M_4} e^{-\lambda_j^2/K_0}$. Note that when restricted to the f fractional variables in any iteration, $\sum_{j=1}^m |S_j| \leq \Delta f$ since each variable appears in at most Δ constraints. So the number of constraints with $|S_j| > 64\Delta$ is at most $\frac{f}{64}$. For $h \geq 0$, the number of constraints with $|S_j| \in [2^{-h-1}64\Delta, 2^{-h}64\Delta)$ is at most $2^{h+1} \frac{f}{64}$. So,

$$\sum_{j \in M_4} e^{-\lambda_j^2/K_0} \leq \frac{f}{64} + \sum_{h=0}^{\infty} 2^{h+1} \frac{f}{64} \exp \left(\frac{-K_1 \Delta}{2^{-h}64\Delta \cdot K_0} \right) \leq \frac{f}{64} + \frac{f}{64} \sum_{h=0}^{\infty} 2^{h+1} e^{-2^{h+2}} \leq \frac{f}{48}.$$

The second inequality is by choosing large enough constant K_1 .

We now bound the error incurred for any constraint $j \in M_4$. The error in a single iteration is at most $O(\sqrt{\Delta}) + \frac{1}{n}$. So the overall error $|x(S_j) - b_j| = O(\sqrt{\Delta} \log n)$.

Overall iteration. By setting the λ_j parameters for the different parts M_1, M_2, M_3, M_4 as above, it follows that in any iteration with f fractional variables, we have $\sum_{j=1}^m e^{-\lambda_j^2/K_0} \leq \frac{f}{24}$ which satisfies the condition in Theorem 4. \square

Remark: The above result also extends to the following ‘‘group sparse’’ setting. Suppose the constraints in M_4 are further partitioned into g groups $\{G_k\}_{k=1}^g$ where the column sparsity restricted to constraints in each group G_k is Δ_k . Then we obtain an integral solution with $|x(S_j) - b_j| = O(\sqrt{g \cdot \Delta_k} \log n)$ for all $j \in G_k$. The only modification required in the above proof is to set $\lambda_j = \sqrt{K_1 \cdot g \cdot \Delta_k} / \sqrt{|S_j|}$ for $j \in G_k$.

3.2 Minimum Cost Degree Bounded Matroid Basis

The input to the *minimum cost degree bounded matroid* problem (DEGMAT) is a matroid defined on elements $V = [n]$ with costs $d : V \rightarrow \mathbb{Z}_+$ and m ‘‘degree constraints’’ $\{S_j, b_j\}_{j=1}^m$ where each $S_j \subseteq [n]$ and $b_j \in \mathbb{Z}_+$. The objective is to find a minimum-cost base I in the matroid that obeys all the degree bounds, i.e. $|I \cap S_j| \leq b_j$ for all $j \in [m]$. Here we make a minor technical assumption that all costs are polynomially bounded integers.

An algorithm for DEGMAT is said to be an $(\alpha, \beta \cdot b + \gamma)$ -bicriteria approximation algorithm if for any instance, it finds a base I satisfying $|I \cap S_j| \leq \beta \cdot b_j + \gamma$ for all $j \in [m]$ and having cost at most α times the optimum (which satisfies all degree bounds).

Theorem 8. *There is a randomized algorithm for DEGMAT, that on any instance, finds a base I^* of cost at most the optimum where for each $j \in [m]$:*

$$|I^* \cap S_j| \leq O(1) \cdot \min \left\{ \sqrt{j}, \sqrt{n \log(m/n)}, \sqrt{\log m \log n \cdot b_j} + \log m \log n, \sqrt{\Delta \log n} \right\}.$$

Proof. Let $y \in [0, 1]^n$ be an optimal solution to the natural LP relaxation of DEGMAT. We now describe the rounding algorithm: this is based on iterative applications of Theorem 5. First, we incorporate the cost as a special degree constraint $v_0 = d$ indexed zero. We will require zero violation in the cost during each iteration, i.e. $\lambda_0 = 0$ always. We partition the degree constraints $[m]$ as in Theorem 7: recall the definitions of M_1, M_2, M_3, M_4 , and the setting of their λ_j parameters in each iteration.

In each iteration, we start with a fractional solution y' with $f \leq n$ fractional variables. Using the same calculations as Theorem 7, we have $\sum_{j=0}^m e^{-\lambda_j^2/K_0} \leq 1 + \frac{f}{24} \leq \frac{f}{16}$ assuming $f \geq 48$. For now assume $f \geq \max\{K_0, 48\}$; applying Theorem 5, we obtain a new fractional solution y'' that has:

- $|\langle v_0, y'' - y' \rangle| \leq \|d\|/n^{O(1)} \leq \frac{1}{n}$.
- For each $j \in [m]$, $|y''(S_j) - y'(S_j)| \leq \lambda_j \sqrt{W_j(y')} + \frac{1}{n}$.
- The number of fractional variables in y'' is at most $\frac{f}{K'}$ for some constant $K' > 1$.

The first condition uses the fact that the error term $\|a_j\|/n^2$ in Theorem 5 can be reduced to $\|a_j\|/n^c$ for any constant c , and that $\|d\| \leq \text{poly}(n)$ as we assumed all costs to be polynomially bounded.

We repeat these iterations as long as $f \geq \max\{K_0, 48\}$: this takes $T \leq \frac{\log n}{\log K'} = O(\log n)$ iterations. The violation in the cost (i.e. constraint $j = 0$) is at most $\frac{T}{n} < 1$. For any degree constraint $j \in [m]$, the violation is exactly as in Theorem 7.

At the end of the above iterations, we are left with an almost integral solution x : it has $O(1)$ fractional variables. Notice that x lies in the matroid base polytope: so it can be expressed as a convex combination of (integral) matroid bases. We output the minimum cost base I^* in this convex decomposition of x . Note that the cost of solution I^* is at most that of x which is less than $\langle d, y \rangle + 1$. Moreover, I^* agrees with x on all integral variables of x : so the worst case additional violation of any degree constraint is just $O(1)$. \square

We state two special cases of this result, which improve on prior work.

Corollary 1. *There are randomized bicriteria approximation algorithms for DEGMAT with ratios $(1, b + O(\sqrt{n \log(m/n)}))$ and $(1, O(\sqrt{\Delta \log n}))$.*

Previously, [CVZ10] obtained a $(1, b + O(\sqrt{n \log(m)}))$ bicriteria approximation and [KLS08] obtained a $(1, \Delta - 1)$ bicriteria approximation for DEGMAT.

3.3 Multi-criteria Matroid Basis

The input to the *multi-criteria matroid basis* is a matroid \mathcal{M} defined on elements $V = [n]$ with k different cost functions $d_j : [n] \rightarrow \mathbb{Z}_+$ (for $j = 1, \dots, k$) and budgets $\{B_j\}_{j=1}^k$. The goal is to find (if possible) a basis I with $d_j(I) \leq B_j$ for each $j \in [k]$. We obtain:

Theorem 9. *There is a randomized algorithm for multi-criteria matroid basis, that given any $\epsilon > 0$ finds in $n^{O(k^{1.5}/\epsilon)}$ time, a basis I with $d_j(I) \leq (1 + \epsilon)B_j$ for all $j \in [k]$.*

Previously, [GRSZ14] obtained a deterministic algorithm for MCM that required $n^{O(k^2/\epsilon)}$ time. One could also use the algorithm of [CVZ10] to obtain a randomized PTAS for MCM, but this approach requires at least $n^{\Omega(k/\epsilon^2)}$ time. Our running time is better when $\epsilon < 1/\sqrt{k}$.

We now describe the algorithm in Theorem 9. An element e is said to be *heavy* if its j^{th} cost $d_j(e) > \frac{\epsilon}{\sqrt{k}}B_j$ for any $j \in [k]$. Note that the optimal solution contains at most $\frac{k^{1.5}}{\epsilon}$ heavy elements. The algorithm first guesses by enumeration all heavy elements in the optimal solution. Let \mathcal{M}' denote the matroid obtained by contracting these heavy elements. Let B'_j denote the residual budget for each $j \in [k]$. The algorithm now solves the natural LP relaxation:

$$x \in P(\mathcal{M}'), \quad \langle d_j, x \rangle \leq B'_j, \quad \forall j \in [k].$$

The rounding algorithm is an iterative application of Theorem 5: the number of fractional variables decreases by a factor of $K > 1$ in each iteration.

As long as the number of fractional variables $n' < 16k$, we use $\lambda_j = 0$ for all $j \in [k]$; note that this satisfies the condition $\sum_{j=1}^k e^{-\lambda_j^2/K_0} \leq n'/16$. Note that there is no loss in any of the budget constraints in this first phase of the rounding.

Once $n' \leq N := 16k$, we choose each $\lambda_j = \sqrt{K_0 \log(N/n')}$ which satisfies the condition on λ s. The loss in the j^{th} budget constraint in such an iteration is at most $\lambda_j \sqrt{n'} \cdot d_j^{\max}$ where $d_j^{\max} \leq \frac{\epsilon}{\sqrt{k}}B_j$ is the maximum cost of any element. So the increase in the j^{th} budget constraint over all iterations is at most:

$$d_j^{\max} \cdot \sum_{i=0}^{t-1} \sqrt{K_0 \frac{N}{K^i} \log(K^i)} \leq O(\sqrt{N}) \cdot d_j^{\max} = O(\epsilon)B_j.$$

Above i indexes iterations in the second phase of rounding.

3.4 Low Congestion Routing on Short Paths

The routing on short paths (RSP) problem is defined on an n -vertex directed graph $G = (V, E)$ with edge capacities $b : E \rightarrow \mathbb{Z}_+$. There are k source-sink pairs $\{(s_i, t_i)\}_{i=1}^k$ and a length bound Δ . The goal in RSP is to find an $s_i - t_i$ path P_i of length at most Δ for each pair $i \in [k]$ such that the number of paths using any edge e is at most b_e .

The decision problem of determining whether there exist such paths is NP-complete. Hence we focus on bicriteria approximation algorithms, where we attempt to find paths P_i s that violate the edge capacities by a small amount. As noted in [CVZ10], we can use any LP-based algorithm for DEGMAT to obtain one for RSP: for completeness we describe this briefly below.

Let \mathcal{P}_i denote the set of all $s_i - t_i$ paths of length at most Δ . Consider the following LP relaxation for RSP.

$$\begin{aligned} \sum_{P \in \mathcal{P}_i} x_{i,P} &\geq 1, & \forall i \in [k] \\ \sum_{i=1}^k \sum_{P \in \mathcal{P}_i: e \in P} x_{i,P} &\leq b_e, & \forall e \in E \\ x &\geq 0. \end{aligned}$$

Although this LP has an exponential number of variables, it can be solved in polynomial time by an equivalent polynomial-size formulation using a “time-expanded network”.

Given any feasible instance of RSP, we obtain a fractional solution to the above LP. Moreover, the number of non-zero variables $x_{i,P}$ is at most $k + |E| = \text{poly}(n)$. Let \mathcal{P}'_i denote the set of $s_i - t_i$ paths with non-zero value in this fractional solution. Consider now an instance of DEGMAT on

groundset $U = \cup_{i=1}^k \mathcal{P}'_i$ where the matroid is a *partition matroid* that requires one element from each \mathcal{P}'_i . The degree constraints correspond to edges $e \in E$, i.e. $S_e = \{P \in U : e \in P\}$. The goal is to find a base I in the partition matroid such that $|S_e \cap I| \leq b_e$ for all $e \in E$. Note that the column sparsity of the degree constraints is Δ since each path in U has length at most Δ . Moreover $\{x_{i,P} : P \in \mathcal{P}'_i, i \in [k]\}$ is a feasible fractional solution to the LP relaxation of this DEGMAT instance. So we obtain:

Corollary 2. *There is an algorithm that given any feasible instance of RSP, computes an $s_i - t_i$ path of length at most Δ for each $i \in [k]$ where the number of paths using any edge e is at most $b_e + \min \left\{ O(\sqrt{\Delta} \log n), O(\sqrt{b_e} \log n + \log^2 n) \right\}$.*

Multipath routing with laminar requirements Our techniques can also handle a richer set of requirements in the RSP problem. In addition to the graph G , pairs $\{(s_i, t_i)\}_{i=1}^k$ and length bound Δ , there is a laminar family \mathcal{L} defined on the pairs $[k]$ with an integer requirement r_T on each set $T \in \mathcal{L}$. The goal in the *laminar RSP* problem is to find a multiset of $s_i - t_i$ paths (for $i \in [k]$) such that:

1. each path has length at most Δ ,
2. for each $T \in \mathcal{L}$, there are at least r_T paths between pairs of T , and
3. the number of paths using any edge e is at most b_e .

Consider the following LP relaxation for this problem.

$$\begin{aligned} \sum_{i \in T} \sum_{P \in \mathcal{P}'_i} x_{i,P} &\geq r_T, & \forall T \in \mathcal{L} \\ \sum_{i=1}^k \sum_{P \in \mathcal{P}'_i: e \in P} x_{i,P} &\leq b_e, & \forall e \in E \\ x &\geq 0. \end{aligned}$$

This LP can again be solved using an equivalent polynomial-sized LP. Let \mathcal{P}'_i denote the set of $s_i - t_i$ paths with non-zero value in this fractional solution, and define groundset $U = \cup_{i=1}^k \mathcal{P}'_i$. As before, we also define “degree constraints” corresponding to edges $e \in E$, i.e. at most b_e elements can be chosen from $S_e = \{P \in U : e \in P\}$. Unlike the usual RSP problem we can not directly cast these laminar requirements as a matroid constraint, but a slight modification of the DEGMAT algorithm works.

The main idea is that the partial rounding result (Theorem 5) also holds if we want to exactly preserve any laminar family \mathcal{L} of constraints (instead of a matroid). Note that a laminar family on $|U|$ elements might have $2|U|$ sets. However, it is easy to see that the number of *tight* constraints of \mathcal{L} at any strictly fractional solution is at most $|U|/2$. Using this observation in place of Claim 2, we obtain the partial rounding result also for laminar constraints.

Finally using this partial rounding as in Theorem 8, we obtain:

Theorem 10. *There is an algorithm that given any feasible instance of laminar RSP, computes a multiset \mathcal{Q} of $s_i - t_i$ paths such that:*

1. each path in \mathcal{Q} has length at most Δ ,
2. for each $T \in \mathcal{L}$, there are at least r_T paths in \mathcal{Q} between pairs of T , and
3. the number of paths in \mathcal{Q} using any edge e is at most:

$$b_e + \min \left\{ O(\sqrt{\Delta} \log n), O(\sqrt{b_e} \log n + \log^2 n) \right\}.$$

References

- [Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *Foundations of Computer Science (FOCS)*, pages 3–10, 2010.
- [BCKL14] Nikhil Bansal, Moses Charikar, Ravishankar Krishnaswamy, and Shi Li. Better algorithms and hardness for broadcast scheduling via a discrepancy approach. In *SODA*, pages 55–71, 2014.
- [BF81] J. Beck and T. Fiala. Integer-making theorems. *Discrete Applied Mathematics*, 3:1–8, 1981.
- [BKK⁺13] Nikhil Bansal, Rohit Khandekar, Jochen Könemann, Viswanath Nagarajan, and Britta Peis. On generalizations of network design problems with degree bounds. *Math. Program.*, 141(1-2):479–506, 2013.
- [CNN11] Moses Charikar, Alantha Newman, and Aleksandar Nikolov. Tight hardness results for minimizing discrepancy. In *SODA*, pages 1607–1614, 2011.
- [CVZ10] Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*, pages 575–584, 2010.
- [Fre75] David A. Freedman. On tail probabilities for martingales. *Annals of Probability*, 3:100–118, 1975.
- [GRSZ14] Fabrizio Grandoni, R. Ravi, Mohit Singh, and Rico Zenklusen. New approaches to multi-objective optimization. *Math. Program.*, 146(1-2):525–554, 2014.
- [HSS14] Nicholas J. A. Harvey, Roy Schwartz, and Mohit Singh. Discrepancy without partial colorings. In *APPROX/RANDOM 2014*, pages 258–273, 2014.
- [KLR⁺87] Richard M. Karp, Frank Thomson Leighton, Ronald L. Rivest, Clark D. Thompson, Umesh V. Vazirani, and Vijay V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.
- [KLS08] Tamás Király, Lap Chi Lau, and Mohit Singh. Degree bounded matroids and submodular flows. In *IPCO*, pages 259–272, 2008.
- [LLRS01] Frank Thomson Leighton, Chi-Jen Lu, Satish Rao, and Aravind Srinivasan. New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM J. Comput.*, 31(2):626–641, 2001.
- [LM12] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. In *FOCS*, pages 61–67, 2012.
- [LRS11] Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.
- [LSV86] L. Lovasz, J. Spencer, and K. Vesztergombi. Discrepancy of set-systems and matrices. *European J. Combin.*, 7:151–160, 1986.
- [Mat10] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Springer, 2010.
- [NT15] Aleksandar Nikolov and Kunal Talwar. Approximating hereditary discrepancy via small width ellipsoids. In *Symposium on Discrete Algorithms, SODA*, pages 324–336, 2015.

- [Rot13] Thomas Rothvoss. Approximating bin packing within $o(\log \text{OPT} * \log \log \text{OPT})$ bins. In *FOCS*, pages 20–29, 2013.
- [Rot14] Thomas Rothvoss. Constructive discrepancy minimization for convex sets. In *IEEE Symposium on Foundations of Computer Science, FOCS*, pages 140–145, 2014.
- [Sch03] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.
- [SL07] Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC*, pages 661–670, 2007.
- [Spe85] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985.
- [Sri97] Aravind Srinivasan. Improving the discrepancy bound for sparse matrices: Better approximations for sparse lattice approximation problems. In *Symposium on Discrete Algorithms (SODA)*, pages 692–701, 1997.
- [Vaz01] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- [WS11] David Williamson and David Shmoys. *The design of Approximation Algorithms*. Cambridge University Press, 2011.

A Useful Linear Programming Facts

Fact 1. Consider any polyhedron given by $P = \{x : Ax \leq b\}$ where all entries in A, b are integers of size at most $\log_2 B$. Then there is a polynomial (in $\log B$ and size of A) time algorithm that given any point $u \in P$, finds another point $v^* \in P$ where (i) $\|u - v^*\|_1 \leq \frac{1}{n^{\frac{1}{7B}}}$ and (ii) all entries in v^* are rationals of size $O(n^2 \log B)$.

Proof. Let $L := 2n^8 B$ and u' denote the point with coordinates $u'_i = \frac{1}{L} \lfloor L \cdot u_i \rfloor$ for all $i \in [n]$. We now write a linear program that computes the point $v \in P$ with minimum ℓ_1 distance from u' .

$$\begin{aligned}
 \min \quad & \sum_{i=1}^n d_i \\
 \text{s.t.} \quad & Av \leq b \\
 & |Lv_i - Lu'_i| \leq Ld_i \quad \forall i \in [n] \\
 & \sum_{i=1}^n d_i \leq 1 \\
 & \mathbf{d}, \mathbf{v} \in \mathbb{R}^n.
 \end{aligned}$$

Note that the feasible region of this LP is a polytope (bounded polyhedron) due to the last two constraints. So there is an optimal extreme point solution v^* that can be found in polynomial time. Since all constraint coefficients in this LP are integers bounded by L , the entries in v^* must be rationals bounded by $(2nL)^{2n}$. Finally, $u \in P$ corresponds to a feasible solution to this LP with $v = u$, $d_i = |v_i - u_i|$ (for $i \in [n]$) and objective $\|u - u'\|_1 \leq \frac{n}{L}$. It now follows that $\|v^* - u'\|_1 \leq \frac{n}{L}$ and so $\|v^* - u\|_1 \leq \|v^* - u'\|_1 + \|u' - u\|_1 \leq \frac{2n}{L}$. \square

Fact 2. Consider any linear subspace given by $\{x : Ax = 0\}$ where all entries in A are integers of size at most $\log_2 B$. Then there is a polynomial (in $\log B$ and size of A) time algorithm that computes a basis $\{b_j\}_{j=1}^k$ of this subspace where (i) all entries are rationals of size $O(n^2 \log B)$, (ii) $|\langle b_j, b_j \rangle - 1| \leq \frac{1}{n^4 B}$ for all $j \in [k]$, and (iii) $|\langle b_j, b_\ell \rangle| \leq \frac{1}{n^4 B}$ for all $j \neq \ell, j, \ell \in [k]$.

Proof. We can obtain an orthonormal basis $\{b'_j\}_{j=1}^k$ of this subspace using Gaussian elimination and Gram-Schmidt orthogonalization. This clearly satisfies the last two conditions. But some more work is needed since we require the entries in the basis vectors to be bounded integers.

To ensure this, we modify each vector b'_j into b_j separately by applying Fact 1 with polyhedron $P = \{x : Ax = 0\}$, $u = b'_j$, and then set $b_j = v^*$. Now the last condition follows from Fact 1(ii). The first and second conditions follow from Fact 1(i) since $\{b'_j\}_{j=1}^k$ is orthonormal. \square