Concentration Inequalities for Nonlinear Matroid Intersection*

Konstantin Makarychev
Microsoft Research

Warren Schudy
IBM Research

Maxim Sviridenko†
University of Warwick

Abstract

In this work we propose new randomized rounding algorithms for matroid intersection and matroid base polytopes. We prove concentration inequalities for polynomial objective functions and constraints that has numerous applications and can be used in approximation algorithms for Minimum Quadratic Spanning Tree, Unrelated Parallel Machines Scheduling and scheduling with time windows and nonlinear objectives. We also show applications related to Constraint Satisfaction and dense polynomial optimization.

1 Introduction

Randomized rounding is an algorithmic framework that was first introduced in the seminal work by Raghavan and Thompson [27]. The idea is simple yet powerful: Define a boolean integer programming formulation of optimization problem at hand. Then, solve the linear programming relaxation of that integer programming formulation. Finally, use the values x_i^* of the optimal fractional solution in your randomized rounding procedure; for each i, independently set $x_i = 1$ with probability x_i^* and $x_i = 0$ with probability $1 - x_i^*$. The obtained integral solution x_i satisfies or almost satisfies each constraint with high probability by the Chernoff bound. Hence, by the union bound, all constraints are almost satisfied with high probability as well.

This framework and its variants proved to be useful for many optimization problems arising in Computer Science, Operations Research and Combinatorial Optimization. However, its performance degrades when the mathematical programming formulation involves many (exponential number of) constraints or when the constraints have few non-zero entries, and hence the probability that the integral solution x_i does not satisfy a constraint can be too large for the union bound to give meaningful results. Therefore, over the years researchers developed methods that round the fractional solution in a *dependent* fashion such that the final integer solution is guaranteed to be feasible on the set of constraints that are too sensitive for the Chernoff bound. One class of such methods developed recently is the randomized rounding procedure designed by Chekuri, Vondrák and Zenklusen [10, 11] for linear optimization problems where the "hard" constraints are a part of matroid base and matroid intersection polytopes. In this work, we generalize the rounding method of Chekuri, Vondrák and Zenklusen [10, 11] to handle not only linear constraints but also polynomial constraints. We will show in Section 3 various specific problems that can be captured by our framework.

*The conference version of this paper appeared at SODA 2012.

†University of Warwick, sviri@dcs.warwick.ac.uk, Work supported by EPSRC grant EP/J021814/1, EP/D063191/1, FP7 Marie Curie Career Integration Grant and Royal Society Wolfson Research Merit Award.

One of the ways to define a large number of constraints that are still algorithmically manageable is to use the language of Matroid Theory. A matroid $\mathcal{M}$ is an ordered pair $(V, \mathcal{I})$, where V is the (finite) ground set of $\mathcal{M}$, and $\mathcal{I}$ is the set of independent sets of $\mathcal{M}$. We recall that the defining properties for the set $\mathcal{I}$ of independent sets of a matroid are as follows: (i) $\varnothing \in \mathcal{I}$; (ii) $X \subset Y \in \mathcal{I} \Rightarrow X \in \mathcal{I}$; (iii) $X, Y \in \mathcal{I}$, $|X| > |Y| \Rightarrow \exists\, e \in X \setminus Y$ such that $Y \cup \{e\} \in \mathcal{I}$. See books of Oxley [26] and Schrijver [29] for further information on matroids and associated algorithmics.

For a given matroid $\mathcal{M}$, the associated *matroid constraint* is: $S \in \mathcal{I}(\mathcal{M})$. In our usage, we deal with two matroids $\mathcal{M}_i = (V, \mathcal{I}_i)$, $i = 1, 2$, on the common ground set V. We assume that each matroid is given by an *independence oracle*, answering whether or not $S \in \mathcal{I}_i$. For any set S let $rank(S)$ be the rank of S, that is the size of a largest independent subset of S. The set function $rank(S)$ is called the rank function.

For a given matroid $\mathcal{M}$, the associated *polymatroid* $\mathcal{P}(\mathcal{M})$ is a polytope defined by the constraints

$$\sum_{i \in S} x_i \leq rank(S), \qquad \forall S \subseteq V,$$
$$x_i \geq 0, \qquad \forall i \in V.$$

The base polymatroid has one additional constraint $\sum_{i \in V} x_i = rank(V)$ and will be denoted $\mathcal{B}(\mathcal{M})$.

The matroid constraint naturally models various combinatorial constraints, such as cardinality, acyclicity, degree etc., in one unified framework. Let $f(x)$, $g_j(x)$ for $j = 1, \ldots, k$ and $g'_j(x)$ for $j = 1, \ldots, k'$ be arbitrary real-valued functions defined for $x \in [0, 1]^n$ (in this paper we are interested in degree q multivariate polynomials with coefficients in the interval $[0, 1]$). In this work we study two general optimization problems

$$\begin{aligned} &\min f(x), && && (1)\\ &g_j(x) \geq C_j, && j = 1, \ldots, k, && (2)\\ &g'_j(x) \leq C'_j, && j = 1, \ldots, k', && (3)\\ &x \in \mathcal{B}(\mathcal{M}), && && (4)\\ &x_i \in \{0, 1\}, && \forall i \in V. && (5) \end{aligned}$$

$$\begin{aligned} &\max f(x), && && (6)\\ &g_j(x) \geq C_j, && j = 1, \ldots, k, && (7)\\ &g'_j(x) \leq C'_j, && j = 1, \ldots, k', && (8)\\ &x \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2), && && (9)\\ &x_i \in \{0, 1\}, && \forall i \in V. && (10) \end{aligned}$$

Both problems model a variety of computationally difficult optimization problems arising in various applications such as scheduling, network design, facility location etc. (see Section 3).

In many applications, the optimization problems (1)–(5) and (6)–(10) become more tractable if one replaces the integrality constraints (5) and (10) with the constraint $x_i \in [0, 1]$ for all $i \in V$. More precisely, assume that there exists a polynomial (or super-polynomial in some applications) algorithm that finds a relaxed fractional solution of the optimization problems (1)–(5) and (6)–(10), possibly with relaxed right hand side for constraints (2), (3), (7), (8). We apply new dependent randomized rounding procedures to the fractional relaxed solution of (1)–(5) or the fractional relaxed solution of (6)–(10). The formal description of this procedures can be found in Section 2. We show that for a wide class of problems the integral solution obtained by these methods is concentrated around the value of the original fractional solution that implies bounds on the performance guarantee of the final integral solution for a wide variety of applications (see Theorems 3 and 4 and Section 3). The key difference between our work and the work of Chekuri, Vondrák and Zenklusen

[10, 11] is that their concentration bounds hold only for linear and submodular functions which limits the potential applications of the method.

In this paper we consider the case when the functions $f(x)$, $g_j(x)$ for $j = 1, \ldots, k$ and $g'_j(x)$ for $j = 1, \ldots, k'$ are degree q polynomials with coefficients at most 1, i.e. of the form

$$\sum_{\substack{d_1, d_2, \ldots, d_n \in \mathcal{Z}^+ \\ \text{s.t. } \sum_i d_i \leq q}} c_{d_1, \ldots, d_n} x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n}$$

where $0 \leq c_{d_1, \ldots, d_n} \leq 1$. For a given fractional solution x_i^* for $i = 1, \ldots, n$, let $r = \sum_{i \in V} x_i^*$. The value r is called the fractional rank of the solution x^*. A small technical observation is that we can assume that r is lower bounded by the rank R of some optimal solution since we can guess R (n options) and add the constraint $\sum_{i=1}^n x_i \geq R$ to our mathematical programming formulation. We will use the notation $\chi(S)$ to denote the characteristic vector of the set S. Our first main result implies a randomized rounding procedure for a fractional solution for the optimization problems (1)–(5).

Theorem 1. *For any matroid $\mathcal{M}$ and corresponding base polymatroid $\mathcal{B}(\mathcal{M})$, there exists a polynomial time rounding algorithm that given a vector $x^* \in \mathcal{B}(\mathcal{M})$, outputs a random set J, $\chi(J) \in \mathcal{B}(\mathcal{M})$ with*

$$\mathbf{E}[\chi(J)] = x^*, \tag{11}$$

such that for every polynomial f of degree q with coefficients in the range $[0, 1]$ the following inequality holds:

$$\mathbf{Pr}[|f(\chi(J)) - f(x^*)| \geq \lambda] \leq (q+1) e^{-\frac{C\lambda^2}{Dqf(x^*) + Dq\lambda}}, \tag{12}$$

where C is an absolute constant (not depending on any other parameters) and $D = \max(\lceil r \rceil, 2)^{q-1}$.

Our second main result implies a randomized rounding procedure for a fractional solution for the optimization problems (6)–(10).

Theorem 2. *For any matroids $\mathcal{M}_1$, $\mathcal{M}_2$ and corresponding polymatroids $\mathcal{P}(\mathcal{M}_1)$, $\mathcal{P}(\mathcal{M}_2)$, there exists a polynomial time rounding algorithm that given a vector $x^* \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$ and integer parameter $p \geq 2$ outputs a random set $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ with*

$$\mathbf{E}[\chi(J)] = \tilde{x}, \tag{13}$$

such that for every polynomial f of degree q with coefficients in the range $[0, 1]$ the following inequality holds:

$$\mathbf{Pr}[|f(\chi(J)) - f(\tilde{x})| \geq \lambda] \leq (q+1) e^{-\frac{C\lambda^2}{Dpqf(\tilde{x}) + Dpq\lambda}}, \tag{14}$$

where $\tilde{x} = (1 - 1/p)x^*$, $D = \max(\lceil r \rceil, 2)^{q-1}$, and C is an absolute constant (not depending on any other parameters).

There are a few key differences between our bounds and the bounds from [10, 11]. The bounds in [10, 11] are "dimension-free" for linear functions $g(x)$ with nonnegative coefficients, i.e. they do not depend on $|V|$ or r. While in the case of linear functions $g(x)$ our bounds are similar to the bounds in [10, 11] we cannot expect this property to hold for polynomials of higher degree as demonstrated in the example below. Nevertheless, we will see that our bound has many interesting applications.

The following simple example shows that the dependence on r^{q-1} is necessary. Let $x_1^* = 1/2$ and $x_i^* = 1$ for $i = 2, \ldots, n$. Consider a degree-q polynomial

$$g(x) = \sum_{S \subseteq [n] \setminus \{1\}, |S| = q-1} x_1 \prod_{i \in S} x_i.$$

Obviously, $g(x^*) = \binom{n-1}{q-1}/2$. On the other side any rounding that preserves marginal expectations of variables has $g(\chi(J)) = 0$ or $g(\chi(J)) = \binom{n-1}{q-1}$, each with probability 1/2. Therefore D must be at least $\Omega(\binom{n-1}{q-1}) = \Omega(r^{q-1})$ (hiding functions of q) to guarantee that for $\lambda < \binom{n-1}{q-1}/2$ the probability bound is at least 1.

The key complication in extending concentration results from linear functions as in [10, 11] to polynomials is that the influence of each variable, which we approximate with a derivative, is no longer a constant and worse yet is no longer independent of the other variables. These influences matter because they affect the step-sizes of the martingale used in the analysis. Each derivative is a lower-degree polynomial, which we can show to be concentrated by induction. The algorithm of [10, 11] does not appear to extend to polynomials (at least not with bounds independent of the dimension n) because of the difficulty of handling the dependence between *which variables* are available for rounding at each step of the algorithm and *the derivatives* with respect to those variables. For example we cannot afford to take a union bound over the n variables. We work around this difficulty by using a new and cleaner algorithm: run a certain Markov chain over independent sets for a fixed period of time. This cleaner algorithm can be analyzed using techniques such as the self-bounding property of polynomials which allow to apply the method of bounded variances to the martingale corresponding to our Markov chain.

Due to dependence on r^{q-1} in the exponent of the probability estimates in Theorems 1 and 2 our concentration inequalities provide meaningful bounds only if $\lambda^2 = \Omega(f(x^*)r^{q-1})$ and $\lambda = \Omega(r^{q-1})$. Since in most applications we are interested $\lambda = \varepsilon f(x^*)$ for some small constant $\varepsilon > 0$ we must have $f(x^*) > Cr^{q-1}$ for some large constant C to obtain nontrivial bounds. While this condition is restrictive in some settings it still covers many interesting optimization problems. For example, all applications considered by Arora, Frieze and Kaplan [2] are covered by our framework, they consider the assignment problem (which is a special case of the matroid intersection problem) with polynomial objectives satisfying the condition that the optimal value of objective function is $\Omega(n^q)$. We will follow [2] and call such polynomial objectives *dense*. In particular, our results imply the following theorem.

Theorem 3. *There exists a quasi-polynomial time randomized approximation scheme for the matroid intersection problem with the objective function (that could be maximized or minimized) that is a degree q polynomial and such that the value of the optimal solution is $\Omega(r^q)$ where r is the matroid rank of the optimal solution.*

An analogous result can be stated for the matroid base polytope. For many specific applications such as maximum acyclic subgraph problem or minimum linear arrangement problem the algorithms could be made polynomial using the techniques from [2].

If in addition the polynomials involved in the objective function are convex then we don't need to go through the machinery from [2] to obtain a feasible solution of the continuous relaxation. We just use the polynomial time algorithms for convex programming.

Theorem 4. *There exists a polynomial time randomized approximation scheme for the maximum matroid intersection problem with a concave polynomial objective function and minimum matroid base problem with a convex polynomial objective such that the value of the optimal solutions is $\Omega(r^{q-1}/\varepsilon)$ where r is the matroid rank of the optimal solution and $\varepsilon > 0$ is a precision parameter.*

Notice that here we have a density condition more general than the one in [2]. Also we can add an arbitrary number of convex dense constraints which is very useful since many applications come with multiple objective functions. For example for convex quadratic objective functions we just need the optimal solution to have value $\Omega(r/\varepsilon)$.

Nonlinear matroid intersection problems were studied before in the series of papers [5, 20, 6, 7]. In their setting the objective function is a composition of a complicated function of d variables with d linear functions of n variables defined on the ground set. If d is a constant then polynomial time algorithms can be derived in many settings. For larger values of d there are polynomial time approximation algorithms with performance guarantees depending on d and properties of the function f. While this set of problems is related to our general problems (1)–(5) and (6)–(10) and some applications are shared, these two settings are quite different. Algorithmically we are relying on the randomized rounding of the fractional solutions while [5, 20, 6, 7] use more combinatorial methods.

Another related line of research is the area of probability theory studying concentration of measure in general and concentration inequalities in particular. There are many beautiful and useful results in this area, see surveys and books [14, 12, 8, 22]. The most relevant series of concentration inequalities are due to Boucheron etal. [9], Kim and Vu [16], Vu [37, 38] and Schudy and Sviridenko [30, 31]. The Kim-Vu [16] concentration inequality and its generalizations and improvements [37, 38, 30, 31] show an upper bound on the probability that a polynomial of independent random variables deviates significantly from its mean value. The main difference between our inequality and the ones in [16, 37, 38, 30, 31] is that our polynomial depends on random variables that are generated by some sophisticated iterative randomized rounding procedure, actually we expect all our random variables to be dependent from each other while the polynomial in [16, 37, 38, 30, 31] consists of independent random variables.

2 Rounding algorithms

2.1 Preliminaries

We now present a new matroid intersection rounding algorithm inspired by the algorithm due to Chekuri, Vondrák and Zenklusen [11]. Every fractional solution $x^* \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$ can be represented as a convex combination of n integral solutions $x^{(i)} \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$, i.e. $x^* = \sum_{i=1}^{n} \lambda_i x^{(i)}$ (see Theorem 41.13 in [29]) by a polynomial time algorithm (actually n can be replaced by the number of non-zero components in vector x^*). We assume without loss of generality that $r = \|x^*\|_1$ is an integer greater than 1 since we can always increase r by adding dummy elements, i.e. the elements independent with all other elements in matroid, and this will not change the inequalities 12 and 14. Moreover, by adding dummy elements to the matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ and truncating them at r i.e. adding the cardinality constraint with upper bound r, we can assume that each $x^{(i)}$ is a characteristic vector of a base in matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, and thus $\|x^{(i)}\|_1 = r$.

The algorithm of Chekuri, Vondrák and Zenklusen [11] finds a convex decomposition of a given vector x^*: $x^* = \sum_{i=1}^{n} \lambda_i x^{(i)}$. Then it merges vectors $x^{(i)}$ using swap rounding (see Lemma 20). The merge is performed in many rounds. The details of the algorithm are quite complex. We simplify this approach: we show how to randomly merge an arbitrary feasible solution $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ with a fractional feasible solution x^*. Then, we argue that by performing this merge many times we obtain a set J satisfying properties of Theorem 2.

Lemma 5. *For every common independent set $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$, vector $x^* \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$, parameter $p \in \{2, \ldots, r\}$, and $\tilde{x} = (1 - 1/p)x^*$ there exists a distribution of sets $\pi(J) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$, such that*

- *for every $u \in J$, $\mathbf{Pr}\left[u \notin \pi(J)\right] = (1 - \tilde{x}_u)/r$;*
- *for every $u \notin J$, $\mathbf{Pr}\left[u \in \pi(J)\right] = \tilde{x}_u/r$;*
- *$|J \triangle \pi(J)| \leq 2p$.*

Moreover, the distribution over sets $\pi(J)$ can be computed in polynomial-time (i.e. there is a polynomial number of sets with non-zero probabilities and a polynomial time algorithm to compute such non-zero probabilities).

The first two point in Lemma 5 imply $\mathbf{E}\left[\chi(\pi(J))\right] = \frac{1}{r}\tilde{x} + (1 - \frac{1}{r})\chi(J)$. Intuitively, this lemma shows that given a common independent set J and a vector x^* in the intersection of two polymatroids, we can "reshuffle" our common independent set J and obtain a new common independent set such that the number of changed elements is at most $2p$ and we have precise bounds on probabilities of adding and removing elements to J. The parameter p represents the tradeoff between the size of the "shuffle" and precision, i.e. closeness of $\tilde{x}$ to x^*.

We prove this lemma (relying on the swap procedure of Chekuri, Vondrák and Zenklusen [11]) in Section A. Our lemma could be seen as a reformulation of the Chekuri, Vondrák and Zenklusen [11]) result in the language convenient for our purposes.

2.2 Algorithm

We first give an algorithm for rounding a vector in the matroid intersection polytope (Theorem 2). We describe discrete and continuous time stochastic processes that change a random set $J(t) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ over time $t \in T = [t_0, 1]$. The reason to present both continuous and discrete processes together is that readers from different communities could find one or another more intuitive. At time t_0, we let $J(t_0) = J_0$ where J_0 is a random set satisfying $\mathbf{E}\left[\chi(J_0)\right] = \tilde{x}$ and $J_0 \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$. We obtain J_0 by first finding a convex decomposition $x^* = \sum \lambda_i \chi(I^{(i)})$, $I^{(i)} \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$; picking a random $I^{(i)}$ with probability λ_i; removing every element from $I^{(i)}$ with probability $1/p$; and letting J_0 to be the resulting set. (In fact, we could set $J(t_0)$ in almost arbitrary way e.g., $J(t_0) = \varnothing$). Then, at some random moments $t \in T$, we add and remove some elements from $J(t)$. In the end, the algorithm outputs the set $J(1)$. We then show how to simulate the process in polynomial-time.

We let $t_0 = 1/(4qr)$. We pick a sufficiently large integer number N and let $\Delta t = (1 - t_0)/N$; $T_N = \{t_0 + k\Delta t : 0 \leq k \leq N\}$. Note, that N may be super exponential. In fact, later we let $N \to \infty$ and $\Delta t \to 0$. Throughout the paper we write $o(1)$, $o(\Delta t)$, $O(\Delta t)$ etc. assuming that $\Delta t \to 0$ and all other parameters are fixed. If $\|X_N - Y_N\|_\infty \leq a_N = o(1)$ (i.e., $\lim_{N \to \infty} a_N = 0$), where X_N, Y_N are random variables, and a_N is a sequence of numbers, we write $X = Y + o(1)$.

We describe the stochastic process in two different ways: first, as a limit of discrete time Markov stochastic processes, each of which is generated by an algorithm, and then as a continuous time Markov process.

Discrete Time Stochastic Process. Fix a sufficiently large N. Set $J^N(t_0) = J_0$. At every moment $t \in T_N$, with probability $r\Delta t/t$, the algorithm replaces the set $J^N(t)$ with the set $J^N(t+\Delta t) = \pi(J^N(t))$, where $\pi(J^N(t))$ is the set from Lemma 5; and with probability $(1-r\Delta t/t)$, it keeps the set $J^N(t)$ unchanged i.e., sets $J^N(t+\Delta t) = J^N(t)$. Here, we assume that N is sufficiently large and, hence, $r\Delta t/t \in [0,1]$. In other words, the algorithm generates a Markov process with transition probability (for $J' \neq J$),

$$\mathbf{Pr}\left[J^N(t+\Delta t) = J' \;\mid\; J^N(t) = J\right] = \Lambda(t)\, p(J,J')\, \Delta t, \tag{15}$$

where $\Lambda(t) = r/t$ and $p(J,J') = \mathbf{Pr}\left[\pi(J) = J'\right]$ is the probability of picking the set J' defined in the proof of Lemma 5. Note, that since the set T contains N elements, the running time of the algorithm described above is not polynomial or even exponential. We give an efficient algorithm later.

We always assume that N is sufficiently large. As $N \to \infty$, the processes $J^N(t)$ tend to a continuous time Markov process[1] $J(t)$. We formally define the continuous time Markov process $J(t)$ below.

Continuous Time Stochastic Process. The stochastic process $J(t)$ is a non-homogeneous continuous time Markov process with finitely many states defined on the time interval $T = [t_0, 1]$. The states of the process are sets $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$. We denote the state at time $t \in T$ by $J(t)$. At time t_0, $J(t_0) = J_0$. The generator of the process is $\Lambda(t)p(J,J')$, where as before $\Lambda(t) = r/t$ and $p(J,J')$ is the probability of picking the set J' in Lemma 5. In other words, the process is defined using equation (for $J' \neq J$),

$$\mathbf{Pr}\big[J(t+\Delta t) = J' \;\mid\; J(t) = J\big] = \Lambda(t)\, p(J,J')\, \Delta t + o(\Delta t), \tag{15$'$}$$

when $\Delta t \to 0$. The desired set J returned by the algorithm is the set $J(1)$.

Connection with the Poisson Process. The stochastic process $J(t)$ can be described in a natural way using a non-homogeneous Poisson process[2] $P(t)$ with rate $\Lambda(t)$ defined on $[t_0, 1]$. At moment t_0, $P(t_0) = 0$. Whenever the Poisson process $P(t)$ jumps, the process $J(t)$ goes from one state J to another state $\pi(J)$. Note that the transitions $J \mapsto \pi(J)$ do not depend on t. Thus,

$$J(t) = \pi^{P(t)}(J_0).$$

The same is true for processes $J^N(t)$:

$$J^N(t) = \pi^{B^N(t)}(J_0),$$

[1] The limit is taken in the Skorohod topology. However, we only need that $J^N(1) \to J(1)$ in distribution.

[2] We remind the reader the definition of the Poisson process with rate $\Lambda(t)$. The Poisson process is a process with independent increments. For every $t_1 \leq t_2$, the random variable $P(t_2) - P(t_1)$ is distributed as a Poisson random variable with parameter $\int_{t_1}^{t_2} \Lambda(t)dt$. Each realization of the process is a monotone right-continuous (càdlàg) step function taking values in $\mathbb{N}$. We say that the process jumps at a point t, if $P(t) = \lim_{t'\uparrow t} P(t') + 1$ i.e., the trajectory $P(t)$ has a jump discontinuity at t. Note, that for every t, $(P(t) - \lim_{t'\uparrow t} P(t')) \in \{0,1\}$. The probability that the process jumps in the interval $[t, t+\Delta t]$ is $\Lambda(t)\Delta t + o(\Delta t)$.

here B^N is the number of jumps of the process $J^N(t)$ in the interval $[t_0, t]$. As $N \to \infty$, $B^N(t) \to P(t)$ (in distribution), hence, for every t, the limiting distribution of $J^N(t)$ equals the distribution of $J(t)$.

Algorithm. To prove Theorem 2 we need to show that, first, the process can be simulated in polynomial-time and, second, that the set $J(1)$ satisfies the properties (13) and (14).

The algorithm for computing $J(1)$ first computes the number of jumps of the Poisson process, $P(1)$, and then applies $P(1)$ times, the algorithm from Lemma 5.

Algorithm 1 (expected polynomial-time rounding algorithm)

1. Compute $P = P(1)$.
2. Let $J = J_0$.
3. Repeat P times:
   - $J = \pi(J)$, where $\pi(J)$ is the distribution returned by the algorithm from Lemma 5.
4. Output J.

The number of jumps in the interval $[t_0, 1]$, i.e., $P(1)$, is distributed according to the Poisson distribution with the parameter

$$\tilde{\Lambda} = \int_{t_0}^{1} \Lambda(t)dt = \int_{t_0}^{1} \frac{rdt}{t} = r\log(1/t_0).$$

That is, for every $k \in \mathbb{N}$, $\mathbf{Pr}\left[P(1) = k\right] = e^{-\tilde{\Lambda}}\tilde{\Lambda}^k/(k!)$. Since $\mathbf{E}\left[P(1)\right] = \tilde{\Lambda} = r\log(1/t_0)$, the algorithm running time is polynomial in expectation.

To make the algorithm truly polynomial-time, we need to slightly modify it: either by allowing an exponentially small probability of a failure or by replacing P with a random variable P' which is distributed as P conditioned on $\{P \le 2e\tilde{\Lambda}\}$. Namely, instead of P, we define $\tilde{P}$ as

$$\mathbf{Pr}\left[\tilde{P} = k\right] = \Pr(P(1) = k \;\mid\; P(1) \le 2e\tilde{\Lambda}),$$

and let $J = \pi^{\tilde{P}}(J_0)$. Thus, the number of loop iterations is always bounded by $2e\tilde{\Lambda}$. Observe, that

$$\mathbf{Pr}\left[P(1) \ge 2e\tilde{\Lambda}\right] = \sum_{k \ge 2e\tilde{\Lambda}} \frac{e^{-\tilde{\Lambda}}\tilde{\Lambda}^k}{k!} \le \sum_{k \ge 2e\tilde{\Lambda}} e^{-\tilde{\Lambda}} \left(\frac{e}{k}\right)^k \tilde{\Lambda}^k \le 2^{-\tilde{\Lambda}} = 2^{-r\log(1/t_0)}.$$

Hence, if we condition on $P(1) \le 2e\tilde{\Lambda}$, the probability of every event changes by a factor at most $(1 - 2^{-r})$, particularly inequality (14) still holds possibly with a slightly different C. We also need to verify that $\mathbf{E}[\chi(J)] = \tilde{x}$. Observe, that $\mathbf{E}\left[\pi^k(J_0)\right] = \tilde{x}$ for every fixed natural k. This is shown by induction (using Lemma 5):

$$\begin{aligned}\mathbf{Pr}\left[u \in \pi^{k+1}(J_0)\right] &= \mathbf{Pr}\left[u \in \pi^k(J_0)\right] \cdot \big(1 - \frac{1-\tilde{x}_u}{r}\big) + \mathbf{Pr}\left[u \notin \pi^k(J_0)\right] \cdot \frac{\tilde{x}_u}{r} \\ &= \tilde{x}_u \cdot \big(1 - \frac{1-\tilde{x}_u}{r}\big) + (1-\tilde{x}_u) \cdot \frac{\tilde{x}_u}{r} \\ &= \tilde{x}_u.\end{aligned}$$

Then,

$$\mathbf{E}[\chi(J)] = \sum_{k=0}^{\infty} \mathbf{E}\left[\pi^k(J_0)\right] \mathbf{Pr}\left[\tilde{P} = k\right] = \sum_{k=0}^{\infty} \tilde{x}\, \mathbf{Pr}\left[\tilde{P} = k\right] = \tilde{x}.$$

Lemma 6. *The running time of our randomized rounding algorithm is $O(r \log r)$ times the running time of the algorithm from Lemma 5 generating the probability distribution over independent sets.*

Analysis. By Lemma 22, for every $t \in T$, $J(t) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$. Hence, $J(1) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$. Define

$$\begin{aligned} x^N(t) &= t\,\chi(J^N(t)) + (1-t)\,\tilde{x}; \\ x(t) &= t\,\chi(J(t)) + (1-t)\,\tilde{x}; \end{aligned}$$

here, as in Lemma 22, $\tilde{x} = (1 - 1/p)x^*$. We prove that $x^N(t)$ is almost a martingale: for every $t \in T_N$,

$$\mathbf{E}\left[x_u^N(t + \Delta t) \mid J^N(t)\right] = x_u^N(t) + O(\Delta t^2),$$

and $x(t)$ is a martingale.

Lemma 7. *At every step $t \in T_N$,*

- *for every $u \in J^N(t)$, $\mathbf{Pr}\left[u \notin J^N(t+\Delta t) \mid J^N(t)\right] = (1 - \tilde{x}_u)\Delta t / t$;*
- *for every $u \notin J^N(t)$, $\mathbf{Pr}\left[u \in J^N(t+\Delta t) \mid J^N(t)\right] = \tilde{x}_u \Delta t / t$.*
- *for every $u \in V$, $\mathbf{E}\left[x_u^N(t+\Delta t) \mid J^N(t)\right] = x_u^N(t) + O((\Delta t)^2)$ (uniformly as $\Delta t \to 0$).*
- *for every $u \in V$, $t_2 > t_1$, $\mathbf{E}[x_u(t_2) \mid J(t_1)] = x_u(t_1)$.*

Proof. The probability, that at step t an element $u \in J^N(t)$ is removed from $J^N(t)$ equals the probability that the algorithm chooses to change $J^N(t)$, which is $r\Delta t/t$, times the probability that $u \notin \pi(J^N(t))$, which is $(1 - \tilde{x}_u)/r$ by Lemma 5. Thus, for $u \in J^N(t)$,

$$\mathbf{Pr}\left[u \notin J^N(t+\Delta t) \mid J^N(t)\right] = \frac{(1-\tilde{x}_u)\Delta t}{t}.$$

Similarly, for every $u \notin J^N(t)$ we have

$$\mathbf{Pr}\left[u \in J^N(t) \mid J^N(t)\right] = \frac{r\Delta t}{t} \cdot \frac{\tilde{x}_u}{r} = \frac{\tilde{x}_u \Delta t}{t}.$$

Hence, if $u \in J^N(t)$, then

$$\begin{aligned}
\mathbf{E}\left[x_u^N(t+\Delta t) - x_u^N(t) \ \mid\ J^N(t)\right] &= \mathbf{E}\left[(t+\Delta t)\chi_u(J^N(t+\Delta t)) - t\chi_u(J^N(t)) \ \mid\ J^N(t)\right] - \Delta t\, \tilde{x}_u \\
&= \mathbf{Pr}\left[u \in J^N(t+\Delta t) \ \mid\ J^N(t)\right] \times \Delta t \\
&\qquad - \mathbf{Pr}\left[u \notin J^N(t+\Delta t) \ \mid\ J^N(t)\right] \times t - \tilde{x}_u\, \Delta t \\
&= (1 - O(\Delta t)) \times \Delta t - \frac{(1-\tilde{x}_u)\Delta t}{t} \times t - \tilde{x}_u\, \Delta t \\
&= O((\Delta t)^2).
\end{aligned}$$

If $u \notin J^N(t)$, then

$$\mathbf{E}\left[x_u^N(t+\Delta t) - x_u^N(t) \ \mid\ J^N(t)\right] = \frac{\tilde{x}_u \Delta t}{t} \times (t + \Delta t) - \tilde{x}_u \Delta t = O((\Delta t)^2).$$

The last item follows from item 3: subdivide the interval $[t_1, t_2]$ into $N' \to \infty$ subintervals. The expected change of $x_u(t)$ from one endpoint of each subinterval to another is at most $O((\Delta t)^2) = O((1/N')^2)$. The number of intervals is N'. Thus, $\mathbf{E}\left[x_u(t_2) - x_u(t_1) \ \mid\ J(t_1)\right] = \lim_{N' \to \infty} O((1/N')^2)\, N' = 0$. $\square$

Note, that $\mathbf{E}\left[x(t_0)\right] = t_0 \mathbf{E}\left[\chi(J(t_0))\right] + (1 - t_0)\tilde{x} = \tilde{x}$. Thus, by Lemma 7, because $x(t)$ is a martingale,

$$\mathbf{E}\left[\chi(J(1))\right] = \mathbf{E}\left[x(1)\right] = \mathbf{E}\left[x(t_0)\right] = \tilde{x}.$$

We now estimate the value of $f(\chi(J(1))) = f(x(1))$. To do so, we analyze the behavior of the process $f(x^N(t))$ as $N \to \infty$. Observe, that $\|x^N(t_0) - \tilde{x}\|_1 = \|\chi(J_0) - \tilde{x}\|_1 t_0 < 2rt_0$ (always), and by Lemma 12 (which we prove later),

$$|f_0 - f(\tilde{x})| \equiv |f(x^N(t_0)) - f(\tilde{x})| \leq 8t_0 r\, r^{q-1} = 8t_0 r^q = 4r^{q-1} \leq 4D. \tag{16}$$

Here, we denote $f_0 = f(t_0\chi(J_0) + (1 - t_0)\tilde{x})$. Recall that $D = \max(\lceil r \rceil, 2)^{q-1}$. That is, $f(x^N(t_0))$ is always very close to $f(\tilde{x})$. Thus, we want to show that $f(x^N(t))$ does not change much over time. We fix N and express each $\Delta f(t) \equiv f(x^N(t + \Delta t)) - f(x^N(t))$ as the sum of the linear term, which we denote by $\Delta Y(t)$, and the non-linear term, which we denote by $\Delta Z(t)$:

$$\begin{aligned}
\Delta f(t) &\equiv f(x^N(t+\Delta t)) - f(x^N(t)) \qquad (17)\\
&= \underbrace{\Big(\sum_{u \in V} \frac{\partial f(x^N(t))}{\partial x_u}(x_u^N(t+\Delta t) - x_u^N(t))\Big)}_{\Delta Y(t)} \\
&\quad + \underbrace{\Big(f(x^N(t+\Delta t)) - f(x^N(t)) - \sum_{u \in V} \frac{\partial f(x^N(t))}{\partial x_u}(x^N(t+\Delta t) - x^N(t))\Big)}_{\Delta Z(t)}.
\end{aligned}$$

For $t \in T_N$, denote $Y^N(t) = \sum_{t' \in T_N : t' < t} \Delta Y(t')$; $Z^N(t) = \sum_{t' \in T_N : t' < t} \Delta Z(t')$. Then,

$$f(x^N(t)) = f_0 + Y^N(t) + Z^N(t). \tag{18}$$

Recall that $p \geq 2$ is an integer tradeoff parameter from Theorem 2. We first prove a concentration inequality for $Z^N(t)$. Namely, we prove the following lemma.

Lemma 8. *The following inequality holds, for every $\lambda \geq 128p$, $D \geq r^{q-1}$ and $r \geq 2$,*

$$\mathbf{Pr}\left[\max_{t\in T_N} |Z^N(t)| \geq \lambda D\right] \leq e^{-\frac{\lambda}{171p}} + o(1),$$

as $N \to \infty$.

We present the proof in Section 2.4. To simplify the proof we make no attempt to optimize constants. Then, in Section 2.5, we prove a bound on $Y^N(t)$.

Lemma 9. *The following inequality holds, for every $\lambda \geq 128p$, $D \geq r^{q-1}$, and $r \geq 2$*

$$\mathbf{Pr}\left[\max_{t\in T_N} |Y^N(t)| \geq \lambda D \;\mid\; f_0\right] \leq 2qe^{-\frac{\lambda^2}{30pq(D^{-1}f_0+3\lambda)}} + (q-1)e^{-\frac{\lambda}{171p}} + o(1),$$

as $N \to \infty$. Here $f_0 \equiv f(x(t_0))$.

As a corollary of Lemma 8, Lemma 9 and equation (18) we get the following claim.

Claim 10. *For every $\lambda \geq 128p$, $D \geq r^{q-1}$ and $r \geq 2$*

$$\mathbf{Pr}\left[\max_{t\in T_N} |f(x^N(t)) - f_0| \geq 2\lambda D \;\mid\; f_0\right] \leq 2qe^{-\frac{\lambda^2}{30pq(D^{-1}f_0+3\lambda)}} + qe^{-\frac{\lambda}{171p}} + o(1),$$

as $N \to \infty$. Here $f_0 \equiv f(x(t_0))$.

Note, that in the proof of Lemma 9 we assume by induction that Claim 10 holds for all $q' < q$ (the base is $q' = 0$). For the sake of analysis, we also assume that if $f(x^N(t')) - f_0 \geq 2\lambda D$ for some t', then we stop the process i.e., for every $t \geq t'$, we let $x^N(t) = x^N(t')$. This modification of the process does not change the probability $\mathbf{Pr}\left[\max_{t\in T_N} |f(x^N(t)) - f_0| \geq 2\lambda D \;\mid\; f_0\right]$, since if $f(x^N(t')) - f_0 \geq 2\lambda D$, then always $\max_{t\in T_N} |f(x^N(t)) - f_0| \geq 2\lambda D$. Thus, we may assume that if the algorithm changes $x^N(t)$, then $f(x^N(t)) - f_0 < 2\lambda D$.

We now show that Claim 10 implies Theorem 2. Let $\lambda = \lambda'/(4D)$. First, assume that $\lambda' \geq 512pD$ (then the condition $\lambda > 128p$ of Claim 10 is satisfied). By (16), $|f_0 - f(\tilde{x})| \leq 4D < \lambda'/2$ and, hence, $f_0 < f(\tilde{x}) + \lambda'/2$. Write,

$$\begin{aligned}
\mathbf{Pr}\left[|f(x^N(1)) - f(\tilde{x})| \geq \lambda' \;\mid\; f_0\right] &\leq \mathbf{Pr}\left[|f(x^N(1)) - f_0| \geq \lambda'/2 \;\mid\; f_0\right] \\
&= \mathbf{Pr}\left[|f(x^N(1)) - f_0| \geq 2\lambda D \;\mid\; f_0\right] \\
&\leq 2qe^{-\frac{\lambda'^2/(16D^2)}{30pq(D^{-1}f_0+3\lambda'/(4D))}} + qe^{-\frac{\lambda'/(4D)}{171p}} + o(1) \\
&\leq 2qe^{-\frac{\lambda'^2/(16D^2)}{30pq(D^{-1}(f(\tilde{x})+\lambda'/2)+3\lambda'/(4D))}} + qe^{-\frac{\lambda'/(4D)}{171p}} + o(1) \\
&\leq 2qe^{-\frac{C_1\lambda'^2}{Dpq(f(\tilde{x})+\lambda')}} + qe^{-\frac{C_2\lambda'}{Dp}} + o(1).
\end{aligned}$$

for some sufficiently small absolute constants C_1, C_2. If $\lambda' \leq 512pD$, then the right hand side is greater than 1 (for $C_1 < 1/16$; since $C_1\lambda'^2/(Dpq(f(\tilde{x}) + \lambda')) \leq C_1\lambda'/(Dp) \leq 1/2$ and $e^{1/2} < 2$), and the inequality obviously holds. Since this inequality holds for every f_0, we have

$$\mathbf{Pr}\left[|f(x^N(1)) - f(\tilde{x})| \geq \lambda'\right] \leq 2qe^{-\frac{C_1\lambda'^2}{Dpq(f(\tilde{x})+\lambda')}} + qe^{-\frac{C_2\lambda'}{Dp}} + o(1).$$

In the limit as $N \to \infty$, we get

$$\mathbf{Pr}\left[|f(x(1)) - f(\tilde{x})| \geq \lambda'\right] \leq 2qe^{-\frac{C_1\lambda'^2}{Dpq(f(\tilde{x})+\lambda')}} + qe^{-\frac{C_2\lambda'}{Dp}}.$$

We now slightly simplify the right hand side. The second term can be upper bounded by $qe^{-\frac{C_2\lambda'^2}{Dpq(f(\tilde{x})+\lambda')}}$. Hence,

$$\mathbf{Pr}\left[|f(x(1)) - f(\tilde{x})| \geq \lambda'\right] \leq 3qe^{-\frac{C_3\lambda'^2}{Dpq(f(\tilde{x})+\lambda')}},$$

where $C_3 = \min(C_1, C_2)$. Observe, that for every ε,

$$\min(3qe^{-\varepsilon}, 1) \leq \sqrt{3qe^{-\varepsilon}} \leq (q+1)e^{-\varepsilon/2}.$$

Therefore,

$$\mathbf{Pr}\left[|f(x(1)) - f(\tilde{x})| \geq \lambda'\right] \leq (q+1)e^{-\frac{C_3\lambda'^2}{2Dpq(f(\tilde{x})+\lambda')}}.$$

This concludes the proof of Theorem 2.

2.3 Self-Bounding Properties of Polynomials

We prove several easy bounds on degree-q polynomials. These properties of polynomials will be used later to apply the method of bounded variances and corresponding concentration inequalities (McDiarmid [22]).

Lemma 11. *For every $q \in \mathbb{N}$ and every degree-q polynomial $f(x)$ with non-negative coefficients that are at most 1, $x \in [0,1]^n$, $\|x\|_1 \leq r$, and $r \geq 2$ the following inequalities hold*

1. $f(x) \leq \frac{r^{q+1}-1}{r-1} \leq 2r^q$;
2. $\frac{\partial f(x)}{\partial x_i} \leq 2\frac{(r^q-1)}{r-1} \leq 4r^{q-1}$;
3. $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \leq 8r^{q-2}$ *for* $i \neq j$.
4. $\frac{\partial^2 f(x)}{\partial x_i^2} \leq 16r^{q-2}$.

Proof. We first verify that for $q = 0$ all inequalities above hold. If $q = 0$, then $f(x)$ is a constant, $f(x) \leq 1$. Thus, (1) $f(x) \leq 1$; (2) $\frac{\partial f(x)}{\partial x_i} = 0$; (3) $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = 0$; (4) $\frac{\partial^2 f(x)}{\partial x_i^2} = 0$. We now consider the case $q > 0$.

1. If f contains only monomials of degree q, then

$$f(x) \leq \Big(\sum_{k=1}^{n} x_k\Big)^q = \|x\|_1^q \leq r^q.$$

Now, for arbitrary f, we have

$$f(x) \leq \sum_{k=0}^{q} r^k = \frac{r^{q+1}-1}{r-1} \leq 2r^q.$$

2. Write, $f(x) = x_i g(x) + h(x)$, where h does not depend on x_i. Then, assuming that the inequality holds for $q' < q$, we get

$$\frac{\partial f(x)}{\partial x_i} = x_i \frac{\partial g(x)}{\partial x_i} + g(x) \leq \frac{\partial g(x)}{\partial x_i} + 2r^{q-1} \leq 2\frac{(r^{q-1}-1)}{r-1} + 2r^{q-1} = 2\frac{(r^q-1)}{r-1} \leq 4r^{q-1}.$$

3. Write $f(x) = x_i x_j g(x) + h(x)$, where h does not have monomials multiple $x_i x_j$. Assuming that we proved the inequality for all $q' < q$, we get

$$\begin{aligned} \frac{\partial^2 f(x)}{\partial x_i \partial x_j} &= g(x) + x_i \frac{\partial g(x)}{\partial x_i} + x_j \frac{\partial g(x)}{\partial x_j} + x_i x_j \frac{\partial^2 g(x)}{\partial x_i \partial x_j} \\ &\leq 2r^{q-2} + 4r^{q-3} + 4r^{q-3} + \frac{\partial^2 g(x)}{\partial x_i \partial x_j} \leq 2r^{q-2} + 2r^{q-2} + 2r^{q-2} + 2r^{q-2} \leq 8r^{q-2}. \end{aligned}$$

4. Write $f(x) = x_i^2 g(x) + h(x)$, where h does not have monomials multiple x_i^2. Assuming that we proved the inequality for all $q' < q$, we get

$$\begin{aligned} \frac{\partial^2 f(x)}{\partial x_i^2} &= 2g(x) + 4x_i \frac{\partial g(x)}{\partial x_i} + x_i^2 \frac{\partial^2 g(x)}{\partial x_i^2} \\ &\leq 4r^{q-2} + 16r^{q-3} + 16r^{q-4} \leq 4r^{q-2} + 8r^{q-2} + 4r^{q-2} \leq 16r^{q-2}. \end{aligned}$$

□

Lemma 12. *For any degree-q polynomial $f(x)$ with non-negative coefficients that are at most 1,*

$$\left| f(y) - f(x) - \sum_{i=1}^{n} (y_i - x_i) \frac{\partial f(x)}{\partial x_i} \right| \leq 8\|x-y\|_1^2 r^{q-2},$$

and

$$\left| f(y) - f(x) \right| \leq 4\|x-y\|_1 r^{q-1}$$

for all $x, y \in [0,1]^n$ such that $\|x\|_1 \leq r$, $\|y\|_1 \leq r$ and $r \geq 2$.

Proof. Let $y = x + \delta$ for some $\delta \in [-1,1]^n$. Then using the Taylor expansion of $g(t) = f(x + t \cdot \delta)$ for $t \in [0,1]$ with the Lagrange form for the remainder term we obtain $g(1) = g(0) + g'(0) + g''(\xi)/2$ for some $\xi \in [0,1]$. Therefore,

$$\begin{aligned} \left| f(y) - f(x) - \sum_{i=1}^{n} (y_i - x_i) \frac{\partial f(x)}{\partial x_i} \right| &= \frac{1}{2} \left| \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \delta_j \frac{\partial^2 f(x + \xi \cdot \delta)}{\partial x_i \partial x_j} \right| \\ &\leq \frac{1}{2} \Big(\sum_{i=1}^{n} \sum_{j=1}^{n} |\delta_i \delta_j| \Big) \times \max_{\substack{z = x + \xi \cdot \delta \\ i,j}} \left| \frac{\partial^2 f(z)}{\partial x_i \partial x_j} \right| \\ &= \frac{1}{2} \Big(\sum_{i=1}^{n} |\delta_i| \Big)^2 \times \max_{\substack{z = x + \xi \cdot \delta \\ i,j}} \left| \frac{\partial^2 f(z)}{\partial x_i \partial x_j} \right| \\ &\leq \frac{1}{2} \Big(\sum_{i=1}^{n} |\delta_i| \Big)^2 \times \max_{\substack{z \in [0,1]^n; \|z\|_1 \leq r \\ i,j}} \left| \frac{\partial^2 f(z)}{\partial x_i \partial x_j} \right| \\ &\leq \frac{1}{2} \|x-y\|_1^2 \times 16\, r^{q-2} \leq 8\|x-y\|_1^2\, r^{q-2}. \end{aligned}$$

We have used that $\|z\|_1 = \|(1-\xi)\cdot x + \xi \cdot y\|_1 \le r$, since the norm is a convex function. We also used Lemma 11 (items 3 and 4).

Similarly, $g(1) = g(0) + g'(\xi)$ for some $\xi \in [0,1]$. Thus,

$$|f(y) - f(x)| \le \Big(\sum_{i=1}^{n} |\delta_i|\Big) \times \max_{\substack{z\in[0,1]^n;\|z\|_1\le r\\ i,j}} \left|\frac{\partial f(z)}{\partial x_i}\right| \le 4\|x-y\|_1 r^{q-1}.$$

□

Lemma 13. *For any degree-q polynomial $f(x)$ with non-negative coefficients and vector x,*

$$\sum_u \frac{\partial f(x)}{\partial x_u} x_u \le q f(x).$$

Proof. It is sufficient to prove the inequality for every term $g(x) = a\prod_u (x_u)^{d_u}$ of polynomial $f(x)$ where $\sum_u d_u \le q$,

$$\sum_u x_u \frac{\partial g(x)}{\partial x_u} = \sum_u d_u g(x) \le q g(x).$$

□

2.4 Proof of Lemma 8

We use the following concentration inequality. This inequality is a reformulation of Theorem 3.15 (page 224), McDiarmid [22]. For details see Section B.

Theorem 14. *Let*

$$x(t) = \sum_{t'\in T: t'<t} \Delta x(t)$$

(where $t \in T$) be a stochastic process adapted to the filtration $\mathcal{F}(t)$ ($t \in T$) (i.e., loosely speaking, $x(t)$ depends only on $\mathcal{F}(t)$). Suppose that $\mathbf{E}[\Delta x(t) \mid \mathcal{F}(t)] \le \Delta\mu(t)$ and $\Delta x(t) \le b$ a.s. for a nonrandom sequence $\Delta\mu(t) \ge 0$ and a (nonrandom) constant $b \ge 0$. Then for any $\lambda \ge 0$ and $v \ge 0$,

$$\mathbf{Pr}\left[\max_{t\in T}(x(t)) - \sum_{t\in T}\Delta\mu(t) \ge \lambda \text{ and } \mathcal{V} \le v\right] \le e^{-\frac{\lambda^2}{2v(1+(b\lambda/(3v)))}},$$

where the random variable $\mathcal{V}$ (the predictable quadratic variation of $x(t)$) is the sum of conditional variances

$$\mathcal{V} = \sum_{t\in T} \mathbf{Var}[\Delta x(t) \mid \mathcal{F}(t)] = \sum_{t\in T} \mathbf{E}\left[(\Delta x(t) - \mathbf{E}[\Delta x(t) \mid \mathcal{F}(t)])^2 \mid \mathcal{F}(t)\right].$$

Lemma 8. *The following inequality holds, for every $\lambda \ge 128p$, $D \ge r^{q-1}$ and $r \ge 2$,*

$$\mathbf{Pr}\left[\max_{t\in T_N} |Z^N(t)| \ge \lambda D\right] \le e^{-\frac{\lambda}{171p}} + o(1),$$

as $N \to \infty$.

Proof. Recall, that we defined $\Delta Z(t)$ in equation (17) and $Z^N(t)$ in equation (18). By Lemma 12,

$$|\Delta Z(t)| \le 8r^{q-2}\,\|x^N(t+\Delta t) - x^N(t)\|_1^2 \le \frac{8D}{r}\|x^N(t+\Delta t) - x^N(t)\|_1^2.$$

Denote $\Delta H(t) = \|x^N(t+\Delta t) - x^N(t)\|_1$, $H^N(t) = \sum_{t'\in T_N : t'<t} \Delta H(t)$. At every step the algorithm changes at most $2p$ elements of $J^N(t)$ i.e., $|J^N(t+\Delta t) \triangle J^N(t)| \le 2p$, thus

$$\begin{aligned}\Delta H(t) &= \|(t+\Delta t)\chi(J^N(t+\Delta t)) - t\chi(J^N(t)) - \Delta t\tilde{x}\|_1 \\ &= t\,\|\chi(J^N(t+\Delta t)) - \chi(J^N(t))\|_1 + O(\Delta t) \\ &\le 2pt + O(\Delta t).\end{aligned}$$

and

$$|Z^N(t)| \le \frac{8D}{r}\sum_{t'<t}(\Delta H(t))^2 \le \frac{8D}{r}\max_{t'<t}\Delta H(t')\sum_{t''<t}\Delta H(t'') \le \frac{16Dpt}{r}\,H^N(t) + O(\Delta t). \qquad (19)$$

Then,

$$\begin{aligned}\mathbf{E}\left[\Delta H(t) \mid J^N(t)\right] &\le \mathbf{E}\big[\|(t+\Delta t)\chi(J^N(t+\Delta t)) - t\chi(J^N(t))\|_1 \mid J^N(t)\big] + \Delta t\|\tilde{x}\|_1 \\ &\le \mathbf{E}\big[\|\chi(J^N(t+\Delta t)) - \chi(J^N(t))\|_1 t + \|\chi(J^N(t+\Delta t))\|_1\Delta t \mid J^N(t)\big] + \Delta t\|\tilde{x}\|_1 \\ &= \sum_{u\in J^N(t)} \mathbf{Pr}\left[u \notin J^N(t+\Delta t) \mid J^N(t)\right] t + \\ &\quad + \sum_{u\notin J^N(t)} \mathbf{Pr}\left[u \in J^N(t+\Delta t) \mid J^N(t)\right] t \\ &\quad + \mathbf{E}\left[|J^N(t+\Delta t)| \mid J^N(t)\right]\Delta t + \|\tilde{x}\|_1\Delta t.\end{aligned}$$

Note, that $|J^N(t+\Delta t)| \le r$ (because $J^N(t+\Delta t)$ is an independent set, and r is the rank of $\mathcal{M}_1$ and $\mathcal{M}_2$), and $|\tilde{x}| < |x^*| = r$. Also, by Lemma 7,

$$\begin{aligned}&\sum_{u\in J^N(t)} \mathbf{Pr}\left[u \notin J^N(t+\Delta t) \mid J^N(t)\right] t + \sum_{u\notin J^N(t)} \mathbf{Pr}\left[u \in J^N(t+\Delta t) \mid J^N(t)\right] t \\ &\qquad = \left(\sum_{u\in J^N(t)} \frac{1-\tilde{x}_u}{t}\,t\Delta t\right) + \left(\sum_{u\notin J^N(t)} \frac{\tilde{x}_u}{t}\,t\Delta t\right) \\ &\qquad \le |J^N(t)|\Delta t + \|\tilde{x}\|_1\Delta t \le 2r\Delta t.\end{aligned}$$

Denote $\Delta\mu(t) = 4r\Delta t$. Then, $\mathbf{E}\left[\Delta H(t) \mid J^N(t)\right] \le \Delta\mu(t)$ and $\sum_{t\in T_N}\Delta\mu(t) \le 4r$. Now, we bound the predictable quadratic variation of $H^N(t)$,

$$\begin{aligned}\mathcal{V} &\equiv \sum_{t\in T_N}\mathbf{Var}\left[\Delta H(t) \mid J^N(t)\right] \le \sum_{t\in T_N}\mathbf{E}\left[(\Delta H(t))^2 \mid J^N(t)\right] \\ &\le (2p+o(1))\sum_{t\in T_N}\mathbf{E}\left[\Delta H(t) \mid J^N(t)\right] \le 8pr + o(1).\end{aligned}$$

Applying the concentration inequality from Theorem 14 (with $v = 8pr + o(1) > \mathcal{V}$ and $b = 2p + o(1)$), we get

$$\mathbf{Pr}\left[\max_{t \in T_N}(H(t)) - 4r \geq \lambda'\right] \leq e^{-\frac{(\lambda')^2}{16pr + 4p\lambda'/3}} + o(1),$$

if $\lambda' \geq 4r$, then

$$\mathbf{Pr}\left[\max_{t \in T_N}(H^N(t)) \geq 2\lambda'\right] \leq \mathbf{Pr}\left[\max_{t \in T_N}(H^N(t)) - 4r \geq \lambda'\right] \leq e^{-\frac{(\lambda')^2}{4p\lambda' + 4p\lambda'/3}} + o(1) = e^{-\frac{\lambda'}{16/3\,p}} + o(1).$$

Therefore, for every positive $\lambda' \geq 4r$ (using (19)),

$$\mathbf{Pr}\left[\max_{t \in T_N}|Z^N(t)| \geq \frac{32\lambda' Dp}{r}\right] = \mathbf{Pr}\left[\max_{t \in T_N}|Z^N(t)| \geq \frac{16Dp}{r} \times 2\lambda'\right] \leq e^{-\frac{\lambda'}{16/3\,p}} + o(1);$$

and for $\lambda = 32p\lambda'/r$ (i.e., every $\lambda \geq 128p$),

$$\mathbf{Pr}\left[\max_{t \in T_N}|Z^N(t)| \geq \lambda D\right] < e^{-\frac{3\lambda r}{512p^2}} + o(1) < e^{-\frac{\lambda}{171p}} + o(1).$$

$\square$

2.5 Proof of Lemma 9

Lemma 9. *The following inequality holds, for every $\lambda \geq 128p$, $D \geq r^{q-1}$, and $r \geq 2$*

$$\mathbf{Pr}\left[\max_{t \in T_N}|Y^N(t)| \geq \lambda D \;\mid\; f_0\right] \leq 2qe^{-\frac{\lambda^2}{30pq(D^{-1}f_0 + 3\lambda)}} + (q-1)e^{-\frac{\lambda}{171p}} + o(1),$$

as $N \to \infty$. Here $f_0 \equiv f(x(t_0))$.

Proof. By Lemma 7, $\mathbf{E}\left[x_u^N(t + \Delta t) - x_u^N(t) \;\mid\; J^N(t)\right] = O((\Delta t)^2)$ (for all $u \in V$), thus

$$\mathbf{E}\left[\Delta Y(t) \;\mid\; J^N(t)\right] = \mathbf{E}\left[\sum_{u \in V} \frac{\partial f(x^N(t))}{\partial x_u}(x_u^N(t + \Delta t) - x_u^N(t)) \;\mid\; J^N(t)\right] = O((\Delta t)^2).$$

and $\mathbf{E}\left[Y^N(t)\right] = o(1)$. Note, that $\|x^N(t+\Delta t) - x_u^N(t)\|_1 \leq 2pt + O(\Delta t)$ and $\frac{\partial f(x^N(t))}{\partial x_u} \leq 4r^{q-1} \leq 4D$ (by Lemma 11). Hence, $|\Delta Y(t)| \leq 8pD + o(1)$.

We now estimate the sum of conditional variances of the process $Y^N(t)$.

$$\mathcal{V} = \sum_{t \in T_N} \mathbf{Var}\left[\Delta Y(t) \;\mid\; J^N(t)\right] \leq \sum_{t \in T_N}\Big(\mathbf{E}\left[(\Delta Y(t))^2 \;\mid\; J^N(t)\right]\Big).$$

Then,

$$\mathbf{E}\left[(\Delta Y(t))^2 \;\mid\; J^N(t)\right] = \mathbf{E}\Big[\Big(\sum_{u \in J^N(t)} \frac{\partial f(x(t))}{\partial x_u}\, t \times (1 - \chi_u(J(t + \Delta t))) + \sum_{u \notin J^N(t)} \frac{\partial f(x(t))}{\partial x_u}\, t \times \chi_u(J(t + \Delta t)) + O(\Delta t)\Big)^2 \;\mid\; J^N(t)\Big].$$

The process $J^N(t)$ makes at most one jump in the interval $[t, t+\Delta t]$, so $|J^N(t) \triangle J^N(t+\Delta t)| \le 2p$, and the number of non-zero summands in the expression above is at most $2p$. Thus (by Cauchy–Schwarz),

$$\mathbf{E}\left[(\Delta Y(t))^2 \;\mid\; J^N(t)\right] \le 2p\mathbf{E}\Big[\sum_{u\in J^N(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\, t \times (1-\chi_u(J(t+\Delta t)))\Big)^2 + \sum_{u\notin V(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\, t \times \chi_u(J(t+\Delta t))\Big)^2 \;\mid\; J^N(t)\Big] + o(\Delta t).$$

Using Lemma 7, we find

$$\begin{aligned}\mathbf{E}\left[(\Delta Y(t))^2 \;\mid\; J^N(t)\right] &\le 2p\Bigg(\sum_{u\in J^N(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\, t\Big)^2 \times \frac{(1-\tilde{x}_u)\Delta t}{t} \\ &\quad + \sum_{u\notin J^N(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\, t\Big)^2 \times \frac{\tilde{x}_u \Delta t}{t}\Bigg) + o(\Delta t) \\ &\le 2p\Big(\sum_{u\in J^N(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\Big)^2 t\Delta t + \sum_{u\notin J^N(t)} \Big(\frac{\partial f(x(t))}{\partial x_u}\Big)^2 \tilde{x}_u t\Delta t\Big) + o(\Delta t).\end{aligned}$$

Now, we bound $\big(\frac{\partial f(x(t))}{\partial x_u}\big) \le 4r^{q-1} \le 4D$ (by Lemma 11) and in the first sum, we replace t with $x_u(t) \ge t\chi_u(J^N(t)) = t$ (for $u \in J^N(t)$) (note: all derivatives of f are nonnegative since f has nonnegative coefficients),

$$\mathbf{E}\left[(\Delta Y(t))^2 \;\mid\; J^N(t)\right] \le 8pD \sum_{u\in V} \frac{\partial f(x(t))}{\partial x_u}\, x_u(t)\Delta t + 8tpD \sum_{u\in V} \frac{\partial f(x(t))}{\partial x_u}\, \tilde{x}_u \Delta t + o(\Delta t).$$

We bound the first term using self bounding properties of polynomials (Lemma 13):

$$8pD \sum_{u\in V} \frac{\partial f(x(t))}{\partial x_u}\, x_u(t)\Delta t \le 8pDqf(x_u(t))\Delta t \le 8pDq(f_0 + 2\lambda D)\Delta t.$$

Here we used the fact that we stop the process if $f(x_u(t)) \ge f(x(t_0)) + 2\lambda D$. We bound the second term by the inductive hypothesis (see Claim 10) applied to the polynomial $g(x) = \frac{1}{q}\sum_{u\in V} \frac{\partial f(x(t))}{\partial x_u}\, \tilde{x}_u$ (whose degree is $(q-1)$ and whose coefficients are in the range $[0,1]$). Let $\mathcal{E}$ be the event

$$\mathcal{E} = \left\{\max_{t\in T_N} \Big(\frac{1}{q}\sum_{u\in V} \frac{\partial f(x(t))}{\partial x_u}\, \tilde{x}_u\Big) \le \frac{1}{q}\sum_{u\in V} \frac{\partial f(x(t_0))}{\partial x_u}\, \tilde{x}_u + 2\lambda D\right\},$$

then,

$$\mathbf{Pr}\left[\mathcal{E} \;\mid\; g_0\right] \ge 1 - (q-1)\left(2e^{-\frac{\lambda^2}{30p(q-1)(D^{-1}g_0+3\lambda)}} + e^{-\frac{\lambda}{171p}}\right),$$

where $g_0 = g(x(t_0))$. We estimate g_0. Using that $x_u(t_0) = (1-t_0)\tilde{x}_u + t_0\chi_u(J(t_0)) \ge (1-t_0)\tilde{x}_u$ and $t_0 = 1/(4qr)$, we get $\tilde{x}_u \le 4qr/(4qr-1)\; x_u(t_0) \le q/(q-1)\; x_u(t_0)$. Then,

$$g_0 = \frac{1}{q}\sum_{u\in V} \frac{\partial f(x(t_0))}{\partial x_u}\, \tilde{x}_u \le \frac{q}{q-1}\left(\frac{1}{q}\sum_{u\in V} \frac{\partial f(x(t_0))}{\partial x_u}\, x_u(t_0)\right) \le \frac{qf(t_0)}{q-1} = \frac{qf_0}{q-1}.$$

The last inequality follows from Lemma 13. Thus,

$$\mathbf{Pr}\left[\mathcal{E} \;\mid\; f_0\right] \geq 1-(q-1)\left(2e^{-\frac{\lambda^2}{30pq(D^{-1}f_0+3\lambda)}}+e^{-\frac{\lambda}{171p}}\right).$$

Assume, that $\mathcal{E}$ holds, then

$$8tpD\Delta t\Big(\sum_{u\in V}\frac{\partial f(x(t))}{\partial x_u}\,\tilde{x}_u\Big) \leq 8tpD\Delta t\Big(\sum_{u\in V}\frac{\partial f(x(t_0))}{\partial x_u}\,\tilde{x}_u+2\lambda qD\Big).$$

Since $\tilde{x}_u \leq 4qr/(4qr-1)\,x_u(t_0) \leq 8/7\,x_u(t_0)$,

$$\begin{aligned}
8tpD\Delta t\Big(\sum_{u\in V}\frac{\partial f(x(t))}{\partial x_u}\,\tilde{x}_u\Big) &\leq 8tpD\Delta t\Big(\frac{8}{7}\sum_{u\in V}\frac{\partial f(x(t_0))}{\partial x_u}\,x_u(t_0)+2\lambda qD\Big)\\
&\leq 10tpD\Delta t\big(qf_0+2\lambda qD\big).
\end{aligned}$$

In the last inequality, we again used the self bounding properties of polynomials (Lemma 13).

We thus get that the following upper bound on the sum of conditional variances (assuming $\mathcal{E}$ holds),

$$\mathcal{V} \leq 10tpDq(f_0+2\lambda D)\int_{t_0}^{1}(1+t)dt+o(1) < 15pqD(f_0+2\lambda D).$$

Applying the concentration inequality from Theorem 14 (with $v = 15pqD(f_0+2\lambda D)$, $b = 15pD+o(1)$, $\Delta\mu(t) = O((\Delta t)^2)$), we get (note $Y(t_0)=0$)

$$\begin{aligned}
\mathbf{Pr}\left[\max_{t\in T}|Y^N(t)| \geq \lambda D \text{ and } \mathcal{V}\leq v \;\mid\; f_0\right] &\leq 2e^{-\frac{\lambda^2D^2}{30pqD(f_0+2\lambda D)+16pD^2\lambda/3}}+o(1)\\
&= 2e^{-\frac{\lambda^2}{30pq(D^{-1}f_0+2\lambda)+16p\lambda/3}}+o(1)\\
&\leq 2e^{-\frac{\lambda^2}{30pq(D^{-1}f_0+3\lambda)}}+o(1).
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathbf{Pr}\left[\max_{t\in T}|Y^N(t)| \geq \lambda D \;\mid\; f_0\right] &\leq \mathbf{Pr}\left[\max_{t\in T}|Y^N(t)| \geq \lambda D \text{ and } \mathcal{V}\leq v \;\mid\; f_0\right]+\mathbf{Pr}\left[\mathcal{V}\geq v \;\mid\; f_0\right]\\
&\leq 2qe^{-\frac{\lambda^2}{30pq(D^{-1}f_0+3\lambda)}}+(q-1)e^{-\frac{\lambda}{171p}}+o(1).
\end{aligned}$$

□

2.6 Base Polytope Rounding

We now discuss Theorem 1. We use the same algorithm as before. We only need to make a minor change. We replace Lemma 5 with the following lemma.

Lemma 15 (Analog of Lemma 5)**.** *For every set $J \in \mathcal{B}(\mathcal{M})$, vector $x^* \in \mathcal{B}(\mathcal{M})$ there exists a probabilistic distribution of sets $\pi(J)$, $\chi(\pi(J)) \in \mathcal{B}(\mathcal{M})$, such that*

- *for every* $u \in J$, $\mathbf{Pr}\left[u \notin \pi(J)\right] = (1 - x_u)/r$*;*
- *for every* $u \notin J$, $\mathbf{Pr}\left[u \in \pi(J)\right] = x_u/r$*;*
- $|J \triangle \pi(J)| \leq 2$.

Moreover, the distribution over sets $\pi(J)$ *can be computed in polynomial-time.*

In the algorithm, we use π as the transition function for the set J. The lemma follows from the standard basis exchange property (compare with Lemma 22): for every two bases $I, J \in \mathcal{B}(\mathcal{M})$, the exists a matching $\{(a_i, b_i)\}$ between $J \setminus I$ and $I \setminus J$ such that for every i,

$$\chi((J \setminus a_i) \cup \{b_i\}) \in \mathcal{B}(\mathcal{M}).$$

3 Applications

For every potential application that can be formulated as the mathematical programming problems (1)–(5) or (6)–(10), we need to show two things. First, we must show that we can find a good fractional solution within a reasonable time. A very powerful tool to solve this problem is convex and linear programming, i.e. we must show how to reduce one non-linear (and possibly non-convex) constraint to a set of more tractable convex constraints and guarantee decent approximation of the original set of constraints. Second, we need to demonstrate that the concentration bounds from Theorems 1 and 2 imply meaningful bounds in terms of the parameters of that specific application.

3.1 Quadratic Minimum Spanning Tree

Our first application is a generalization of the classical Minimum Spanning Tree Problem. In the classical problem we are given an undirected weighted graph $G = (V, E)$ the goal is to find a spanning tree T of minimum weight. In the Quadratic Minimum Spanning Tree Problem we are given nonnegative weights $c_e \geq 0$ for each edge and weights $w_{e,e'}$ for each pair of edges. We assume without loss of generality that $w_{e,e} = 0$. The goal is to find a spanning tree T in graph G that minimizes the objective function $\sum_{e \in T} c_e + \sum_{e,e' \in T} w_{e,e'}$. This problem has applications in transportation, telecommunication, irrigation and energy distribution and received some attention in Operations Research literature [3, 15, 25, 39]. In many applications one needs to balance between various objectives which are usually modeled as constraints in the mathematical programming formulation such spanning tree problems also received a lot of attention in OR literature [17, 23, 35]. We consider the Multi-Objective Quadratic Spanning tree problem.

Let $c_{ej} \in [0, 1]$ and $w_{e,e',j} \in [0, 1]$ be the coefficients in the j-th objective for $j = 0, \ldots, k$. Let $\mathcal{M}$ be the graphic matroid of graph G, i.e. forests in graph G are the independent sets in $\mathcal{M}$ and spanning trees in G are bases in $\mathcal{M}$. Consider the following continuous mathematical programming relaxation of the problem

$$\begin{aligned}
\min \sum_{e \in E} c_{e0} x_e^2 + \sum_{e,e' \in E} w_{e,e',0} x_e x_{e'}, & \\
\sum_{e \in E} c_{ej} x_e^2 + \sum_{e,e' \in E} w_{e,e',j} x_e x_{e'} \leq C_j, & \qquad j = 1, \ldots, k, \\
x \in \mathcal{B}(\mathcal{M}), & \\
0 \leq x_e \leq 1, & \qquad \forall e \in E.
\end{aligned}$$

Let A^j be the $|E| \times |E|$ matrix with diagonal elements $A^j(e,e) = c_{ej}$ and off-diagonal elements $A^j_{e,e'} = w_{e,e',j}/2$. Then we can re-write the above mathematical programming problem as

$$\min \sum_{e \in E} x^\intercal A^0 x, \tag{20}$$

$$x^\intercal A^j x \leq C_j, \qquad j = 1, \ldots, k, \tag{21}$$

$$x \in \mathcal{B}(\mathcal{M}), \tag{22}$$

$$0 \leq x_e \leq 1, \qquad \forall e \in E. \tag{23}$$

The objective function and constraints of this mathematical programming problem are convex if all matrices A^j are positive semidefinite. One possible way to define positive semidefinite matrices is to associate a set of labels S_{ej} with each edge $e \in E$ and define $w_{e,e',j} = |S_{ej} \cap S_{e'j}|/\Lambda$ where $\Lambda = \max_e |S_e|$. This is a measure of how different the label sets for edges e and e' are. Then the matrix A^j is positive semidefinite if $c^j_e \geq |S_{ej}|/\Lambda$. In this case we can solve our continuous mathematical programming relaxation with arbitrary precision by known methods in convex programming. Note that if all sets have the same cardinality, then for any integral solution x the value of $x^\intercal A^j x$ lies in the interval $[\Theta(n), \Theta(n^2)]$ (depending on the structure of label sets).

Let x^* be an optimal fractional solution of the relaxation (20)–(23). Applying the randomized rounding from the proof of Theorem 1 we obtain an integral solution $\tilde{x}$ such that the probability that we have an error larger than εC_j for objective function $g_j(x^*) = x^{*\intercal} A^j x^*$ is $e^{-\varepsilon^2 C_j/\Theta(n)}$ since $\sum_{e \in E} x^*_e = r = n - 1$. Therefore, if the right hand side $C_j \gg n$ then with high probability our integral solution violates the right hand side by a factor of at most $1 + \varepsilon$. Analogously, the error term for the objective function $g_0(x)$ is negligible with high probability if $\varepsilon g_0(x^*) \gg n$.

3.2 Unrelated Parallel Machine Scheduling

We consider the problem of scheduling unrelated parallel machines with multiple objectives. An instance of the problem consists of a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n jobs and a set $M = \{M_1 \ldots, M_m\}$ of m machines. The job J_j has a processing time p_{ij} if it is assigned to be processed on machine M_i. We assume that $p_{ij} \in [0,1]$ or $p_{ij} = +\infty$ (which means that this job cannot be processed on machine M_i). There are costs $c_{ij} \in [0,1]$ associated with processing job J_j on machine M_i. Each job must be processed without interruption for the respective amount of time on one of the m machines. Every machine can process at most one job at a time. The goal is to assign each job to a machine and find an ordering of the jobs assigned to each machine to optimize suitable objectives.

The unrelated parallel machine scheduling is one of the classical scheduling models arising in various applications with various and sometimes multiple objective functions (see for example [21, 4, 36]). Here we just choose two objectives to illustrate our method. The first objective is the total squared load of each machine. Formally, let L_i be the total sum of processing times of jobs assigned to the machine M_i. Then we would like to minimize $\sum_{i=1}^m L_i^2$ (see [4, 36] for additional motivation, references and algorithms on unrelated parallel machines with this objective). Intuitively, this objective tries to balance the load on different machines.

On the other side, our jobs can belong to different customers, i.e. $\mathcal{J} = \cup_{s=1}^q r_s$ and we would like to be fair to all customers and balance the cost of processing jobs for all the customers. Note that this objective is different from bounding the total cost of processing jobs as in [32]. If C_s is the total cost of jobs in group R_s then our second objective function is to minimize $\sum_{s=1}^q C_s^2$. Actually, this objective function is substantially more difficult than the sum of squared machine loads since each term C_s includes variables from different jobs and machines. This complication basically breaks the type of analysis from [36, 10] that was based on negative correlation of variables with the same machine (or job) indices.

We formulate the problem as follows

$$\min g_1(x) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} p_{ij} x_{ij} \right)^2, \tag{24}$$

$$g_2(x) = \sum_{s=1}^{q} \left(\sum_{j \in R_s} \sum_{i=1}^{m} c_{ij} x_{ij} \right)^2 \leq L, \tag{25}$$

$$\sum_{M_i \in M} x_{ij} = 1, \quad J_j \in \mathcal{J}, \tag{26}$$

$$x_{ij} \in \{0, 1\}, \quad J_j \in \mathcal{J}, M_i \in \mathcal{M}. \tag{27}$$

The constraints (26) are matroid base constraints in the partition matroid on the ground set consisting of pairs (M_i, J_j) for $M_i \in M$ and $J_j \in \mathcal{J}$. Moreover, both the objective function (24) and the constraint (25) are convex. Therefore, we can solve the continuous relaxation of this problem in polynomial time with arbitrary precision. Let x^* be an optimal solution of such relaxation.

Theorem 1 implies that the error in the objective function (24) and the constraint (25) is at most $\varepsilon g_\tau(x^*)$ for $\tau = 1, 2$ with high probability if $g_\tau(x^*) \gg n$. This is a reasonable assumption since the value of the quadratic polynomial in (24) could be as high as n^2 and the value of the quadratic polynomial in (25) could be as high as n^2 depending on the instance.

Obviously we can easily extend our result for any constant (or relatively slowly growing) number of objective functions. We could have many different partitions of jobs into customer groups such as zipcodes, income levels, political preferences etc., for each such partition we could define the fairness objective function like (25). We can obviously add any polynomial number of linear constraints, like cardinality constraints for each machine or upper bounds for machine loads. The total weighted completion time objective can be also added to our framework by using the method developed by Skutella [34] and Sethuraman and Squillante [28] to approximate such an objective with a convex function.

3.3 Scheduling with Time Windows

Scheduling and Vehicle Routing with Time Windows is a large area of Operations Research. It studies the problems when each job (client) comes with a set of time windows where it can be executed (served); see surveys [24, 33] for further applications and references. We consider one specific problem in this area.

An instance of the problem consists of a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n jobs and a set $M = \{M_1 \ldots, M_m\}$ of m machines. Each job has a unit processing time. For each job J_j, machine M_i and time t we define $c_{ijt} = 1$ if this job can be processed on M_i at time t and $c_{ijt} = 0$, otherwise. We define a matroid $\mathcal{M}$ with elements $\mathcal{J}$. Each set of jobs that can be simultaneously be assigned to a feasible machine-time slot is independent. To check that $\mathcal{M}$ is indeed a matroid we just need to notice that this is exactly the definition of a *transversal matroid* [29]. For each pair (i, t), let A_{it} be the set of jobs that can be processed on machine M_i at time t. Then any independent set in $\mathcal{M}$

is exactly a partial transversal of the set system $\{A_{it}\}$. In addition, jobs are partitioned into sets $X_1, \ldots, X_q$ where all jobs from the same set are incompatible and cannot be scheduled together. This constraint can be modeled as a standard partition matroid constraint.

The usual objective function is linear, we would like to maximize the total weight of processed jobs. In addition, we associate a cost $c_j \in [0,1]$ with processing of every job and ask to balance the cost distribution between various groups of jobs as in (25), i.e. $\mathcal{J} = \cup_{s=1}^{q} r_s$ and we consider the constraint

$$\sum_{s=1}^{q} \left(\sum_{j \in R_s \cap S} c_j \right)^2 \leq L \tag{28}$$

where S is the set of chosen jobs. One can easily define a continuous convex relaxation as in the previous section. The bounds of the Theorem 2 imply that one can find an integral solution of value close to the value of the fractional optimal solution with high probability. The fairness constraint (28) is violated by the amount εL with probability $e^{-\varepsilon^2 L/\Theta(r)}$ which is negligible if $L \gg r = rank(\mathcal{M})$.

3.4 Constraint Satisfaction

In this section we will show implications of our concentration theorems for constraints arising from the classical constraint satisfaction problems. For illustration purposes consider the constraint corresponding to the maximum cut problem in undirected graphs with edge weights $w_{ij} \in [0, 1/2]$, i.e.

$$g(x) = \sum_{(i,j) \in E} w_{ij}(x_i + x_j - 2x_i x_j) \geq W. \tag{29}$$

That is we consider a problem of general type (1)–(5) or (6)–(10) when some of the constraints have above form. Consider the linearization of the constraint (29)

$$\sum_{(i,j) \in E} w_{ij} z_{ij} \geq W, \tag{30}$$

$$z_{ij} \leq x_i + x_j, \tag{31}$$

$$z_{ij} \leq 2 - x_i - x_j \tag{32}$$

$$0 \leq z_{ij} \leq 1 \qquad (i,j) \in E. \tag{33}$$

We solve the continuous relaxation of the problem (1)–(5) or (6)–(10) with linearized constraints. Let x^* be the optimal fractional solution and

$$L(x^*) = \sum_{(i,j) \in E} w_{ij} \max\{x_i^* + x_j^*, 2 - x_i^* - x_j^*\}.$$

It is known [1] that $g(x) \geq L(x)/2$ for any $x \in [0,1]^n$. Therefore, $g(x^*) \geq W/2$. Since the polynomial $g(x)$ has positive and negative coefficients we apply our concentration theorems for the linear function $\sum_{(i,j) \in E} w_{ij}(x_i + x_j)$ and polynomial $\sum_{(i,j) \in E} 2w_{ij} x_i x_j$ separately. Note, that $\sum_{(i,j) \in E} w_{ij}(x_i^* + x_j^*) = \Theta(r)$ and polynomial $\sum_{(i,j) \in E} 2w_{ij} x_i^* x_j^* = \Theta(r^2)$. We obtain that for a rounded solution $g(\tilde{x}) \geq (1 - \varepsilon)W/2$ with high probability $1 - e^{-\varepsilon^2 W^2/\Theta(r^3)}$ if $W \gg r^{3/2}$. Note that this is not very restrictive since a random cut contains half of the graph edges.

More generally, instead of maximum cut constraint (29) we could have any maximum constraint satisfaction type of problem (e.g. Max k-SAT or Not-All-Equal-k-SAT) as long as we have two conditions. First, there is a reasonable (preferably constant) gap between original (like (29)) and linearized (like (30–33)) constraints. Second, the right hand side W is much larger than square root of the denominator of the fraction in Theorems 1 or 2.

3.5 Dense Polynomials

Consider the following optimization problem

$$\max f(x), \tag{34}$$
$$g_j(x) \geq C_j, \qquad j = 1, \ldots, k, \tag{35}$$
$$x \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2), \tag{36}$$
$$x_i \in \{0, 1\}, \qquad \forall i \in V, \tag{37}$$

where $f(x)$ and $g_j(x)$ are degree-q polynomials for all $j = 1, \ldots, k$ with absolute value of coefficients upper bounded by one. Let $\tilde{r} = \min\{rank(\mathcal{M}_1), rank(\mathcal{M}_2)\}$. Arora, Frieze and Kaplan [2] proved (implicitly) the following theorem.

Theorem 16 ([2]). *Given the optimization problem (34)–(37), it is possible to define $O(n^{O(q^4 k \log n/\varepsilon^2)})$ linear programs such that for at least one linear program every feasible solution of (34)–(37) is a feasible solution of that linear program. Moreover, for every optimal fractional solution x^* of that linear program and every optimal integral solution $\bar{x}$ of (34)–(37) we have $f(x^*) = f(\bar{x}) \pm \varepsilon \tilde{r}^q$ and $g_j(x^*) = g_j(\bar{x}) \pm \varepsilon \tilde{r}^q$ for $j = 1, \ldots, k$.*

Theorem 16 implies that in time $O(n^{O(q^4 k \log n/\varepsilon^2)})$ one can find a reasonable approximate solution x^* to the continuous relaxation of the problem (34–37). Combining Theorem 2 and our rounding algorithm applied to the fractional solution x^*, we obtain an integral solution $\tilde{x}$ such that with high probability $\max_j\{|f(\tilde{x}) - f(x^*)|, |g_j(\tilde{x}) - g_j(x^*)|\} = O\left(\sqrt{r^{q-1} r^q}\right) = O\left(r^{q-1/2}\right)$ where $r = \sum_{i \in V} x_i^* \leq \tilde{r}$, i.e. for large enough $\tilde{r}$ the error term due to the randomized rounding procedure is negligible in comparison with εr^q (the error term due to Theorem 16).

References

[1] A. Ageev and M. Sviridenko, Pipage rounding: a new method of constructing algorithms with proven performance guarantee. Journal of Combinatorial Optimimization 8 (2004), no. 3, 307–328.

[2] S. Arora, A. Frieze and H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arrangement problems, Mathematical Programming, 92(1), 2002, 1–36.

[3] A. Assad and W. Xu, The quadratic minimum spanning tree problem, Naval Research Logistics v39. (1992), 399-417.

[4] Y. Azar and A. Epstein, Convex programming for scheduling unrelated parallel machines, In Proceedings of STOC 2005, pp. 331–337.

[5] Y. Berstein, J. Lee, S. Onn and R. Weismantel, Parametric nonlinear discrete optimization over well-described sets and matroid intersections, Math. Program. 124(1-2) (2010), pp. 233–253.

[6] Y. Berstein, J. Lee, H. Maruri-Aguilar, S. Onn, E. Riccomagno, R. Weismantel and H. Wynn, Nonlinear Matroid Optimization and Experimental Design, SIAM Journal on Discrete Mathematics 22(3) (2008), pp. 901–919.

[7] Y. Berstein and S. Onn, Nonlinear bipartite matching, Discrete Optimization 5(1) (2008), pp. 53–65.

[8] S.Boucheron, G. Lugosi and P.Massart, CONCENTRATION INEQUALITIES. Oxford University Press. March 2013.

[9] S. Boucheron, O. Bousquet, G. Lugosi and P. Massart, Moment inequalities for functions of independent random variables. The Annals of Probability, 33(2) (2005), 514-560.

[10] C. Chekuri, J. Vondrák, R. Zenklusen, Dependent randomized rounding via exchange properties of combinatorial structures, FOCS 2010, pp.575–584.

[11] C. Chekuri, J. Vondrák, R. Zenklusen, Multi-budgeted matchings and matroid intersection via dependent rounding, SODA 2011, pp. 1080–1097.

[12] F. Chung and L. Lu, Concentration inequalities and martingale inequalities: a survey, Internet Math. 3 (2006), no. 1, 79-127.

[13] R. Cordone and G. Passeri, Heuristic and exact approaches to the Quadratic Minimum Spanning Tree Problem, CTW 2008: 52-55.

[14] D. Dubhashi and A. Panconesi, Concentration of measure for the analysis of randomized algorithms, Cambridge University Press, Cambridge, 2009.

[15] J. Gao and Lu, M., Fuzzy quadratic minimum spanning tree problem, Applied Mathematics and Computation v.164 (2005), pp. 773–788.

[16] J. Kim and V. Vu, Concentration of multivariate polynomials and its applications, Combinatorica 20 (2000), no. 3, 417-434.

[17] R. Kumar, P. Singh and P. Chakrabarti, Improved quality of solutions for multiobjective spanning tree problem using distributed evolutionary algorithm, High Performance Computing - HiPC 2004 Lecture Notes in Computer Science, 2005, Volume 3296/2005, 97-112.

[18] J. Lee, V. Mirrokni, V. Nagarajan, M. Sviridenko. Maximizing non-monotone submodular functions under matroid and knapsack constraints, SIAM Journal on Discrete Mathematics 23(4): 2053–2078 (2010).

[19] J. Lee, M. Sviridenko, J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties, Mathematics of Operations Research, V. 35 (2010), pp. 795–806.

[20] J. Lee, S. Onn and R. Weismantel, Approximate nonlinear optimization over weighted independence systems, SIAM Journal on Discrete Mathematics 23(4) (2009), pp. 1667–1681.

[21] J. Leung, L. Kelly and J. Anderson. Handbook of Scheduling: Algorithms, Models, and Performance Analysis. CRC Press, Inc. Boca Raton, FL, USA, 2004.

[22] C. McDiarmid, Concentration, In Probabilistic Methods for Algorithmic Discrete Mathematics, M. Habib, C. McDiarmid, J. Ramirez-Alfonsin and B. Reed editors, pp. 195–248, Springer, 1998.

[23] F. Neumann and I. Wegener, Minimum spanning trees made easier via multi-objective optimization, In Proceedings of the 2005 conference on Genetic and Evolutionary Computation, pp. 763–769.

[24] K. Neumann, C. Schwindt and J. Zimmermann, Project scheduling with time windows and scarce resources. Temporal and resource-constrained project scheduling with regular and non-regular objective functions. Second edition. Springer-Verlag, Berlin, 2003.

[25] T. Oncan and A. Punnen, The quadratic minimum spanning tree problem: A lower bounding procedure and an efficient search algorithm, Computers and Operations Research v.37, (2010), 1762–1773.

[26] J. Oxley, Matroid theory, Oxford University Press, New York, 1992,

[27] R. Prabhakar Raghavan and C. Thompson, Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica 7(4): 365–374 (1987).

[28] J. Sethuraman and M. S. Squillante, Optimal scheduling of multiclass parallel machines, In Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (1999), 963–964.

[29] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.

[30] W. Schudy and M. Sviridenko, Concentration and moment inequalities for polynomials of independent random variables, arXiv:1104.4997, extended abstract appeared in proceedings of SODA2012.

[31] W. Schudy and M. Sviridenko, Bernstein-like concentration and moment inequalities for polynomials of independent random variables: multilinear case, arXiv:1109.5193.

[32] D. Shmoys and E. Tardos, An approximation algorithm for the generalized assignment problem. Mathematical Programming 62 (1993), no. 3, Ser. A, 461–474.

[33] M. Solomon and J. Desrosiers, Time window constrained routing and scheduling problems, Transportation Science V. 22, No. 1, 1988, pp. 1–13.

[34] M. Skutella, Convex quadratic and semidefinite programming relaxations in scheduling, Journal of ACM 48 (2001), 206–242, FOCS 1998, pp. 472–481.

[35] F. Sourd and O. Spanjaard, A multi-objective branch-and-bound framework. Application to the bi-objective spanning tree problem, INFORMS Journal of Computing 20 (2008), 472–484.

[36] V. Anil Kumar, M. Marathe, S. Parthasarathy and A. Srinivasan, A unified approach to scheduling on unrelated parallel machines, Journal of the ACM V.56 (2009)

[37] V. Vu, Concentration of non-Lipschitz functions and applications, Probabilistic methods in combinatorial optimization. Random Structures Algorithms 20 (2002), no. 3, 262–316.

[38] V. Vu, On the concentration of multivariate polynomials with small expectation, Random Structures Algorithms 16 (2000), no. 4, 344-363.

[39] G. Zhou and M. Gen, An effective genetic algorithm approach to the quadratic minimum spanning tree problem, Computers and Operations Research, v.25 (1998), 229–237.

A Fractional Swap Rounding

We need several definitions and a decomposition lemma to describe fractional swap rounding.

Definition 17. For a common independent set $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ and an arbitrary set of elements $I \subseteq V$, the directed bipartite exchange graph $D_{\mathcal{M}_1}(J, I)$ is a bipartite graph with arcs (j, i) for any pair of elements $i \in I \setminus J$, $j \in J$, such that $J \cup \{i\} \setminus \{j\} \in \mathcal{I}(\mathcal{M}_1)$. The directed bipartite exchange graph $D_{\mathcal{M}_2}(J, I)$ has an arc (i, j) for any pair of elements $i \in I \setminus J$, $j \in J$ such that $J \cup \{i\} \setminus \{j\} \in \mathcal{I}(\mathcal{M}_2)$. The directed graph $D_{\mathcal{M}_1,\mathcal{M}_2}(J, I)$ is the union of two digraphs $D_{\mathcal{M}_1}(J, I)$ and $D_{\mathcal{M}_2}(J, I)$.

Definition 18. A directed path or cycle P in $D_{\mathcal{M}_1,\mathcal{M}_2}(J, I)$ is called irreducible if $P \cap D_{\mathcal{M}_1}(J, I)$ is a unique perfect matching on the vertex set of P in $D_{\mathcal{M}_1}(J, I)$ and $P \cap D_{\mathcal{M}_2}(J, I)$ is a unique perfect matching on the vertex set of P in $D_{\mathcal{M}_2}(J, I)$. Moreover, if P is a path, then both its endpoints must be in J.

The next two lemmas explain why it is important to consider irreducible paths. These lemmas were shown by Chekuri, Vondrák and Zenklusen [11] based on the framework developed in [19].

Lemma 19 ([11])**.** *For any irreducible path in $D_{\mathcal{M}_1,\mathcal{M}_2}(J, I)$ we have $J \triangle V(P) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$.*

Lemma 20 ([11])**.** *We are given $I, J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ with $|I| = |J|$. For any integer $p \geq 2$, we can find in polynomial time a collection of irreducible paths and cycles $\{P_1, \dots, P_m\}$ of length at most $2p - 1$ and $m \leq p|I \triangle J|$ in $D_{\mathcal{M}_1,\mathcal{M}_2}(I, J)$ with coefficients $\gamma_i \geq 0$, $\sum_{i=1}^m \gamma_i = 1$ such that for some $\gamma > 0$,*

$$\sum_{i=1}^{m} \gamma_i \chi(P_i) = \gamma\Big((1 - \frac{1}{p})\chi(I \setminus J) + \chi(J \setminus I)\Big).$$

Chekuri, Vondrák and Zenklusen used these lemmas in their algorithm to merge common independent sets from the convex representation of x^* into one solution. The merge is performed in many phases each consisting of many applications of the lemma.

We prove several easy corollaries that allows us to merge a common independent set $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ with the fractional solution x^*. Then, our algorithm performs a certain (random) number of merges and outputs the result. This significantly simplifies the algorithm and allows us analyze it for the case of polynomials.

Corollary 21. *We are given $I, J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ and $|I| = |J|$. For any integer $p \geq 2$, we can find in polynomial time a collection of set pairs $\{(A_i, B_i)\}$ with coefficients $\gamma'_i \geq 0$, $\sum_i \gamma'_i = 1$ such that for some $\gamma' > 0$,* *1.* $A_i \subseteq J$, $B_i \subseteq I$, $|A_i|, |B_i| \le p$, $A_i \neq \varnothing$ *for all* i*;*

2. $(J \setminus A_i) \cup B_i \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$*;*

3. $B_i \cap (J \setminus A_i) = \varnothing$*;*

4. $\sum_i \gamma'_i \chi(A_i) = \gamma' \chi(J)$ *and* $\sum_i \gamma'_i \chi(B_i) = \gamma'(1 - \frac{1}{p})\chi(I)$*.*

Proof. Find a collection of paths $\{P_1, \ldots, P_m\}$ and weights γ_i as in Lemma 20. Then, enumerate all elements in the set $I \cap J$ and let u_i be the i-th element for $i = 1, \ldots, |I \cap J|$. Let $\Gamma = 1 + \gamma|I \cap J|$ and $\gamma' = \gamma/\Gamma$. We define

- $A_i = J \cap V(P_i)$, $B_i = I \cap V(P_i)$, $\gamma'_i = \gamma_i/\Gamma$ for $i = 1, \ldots, m$;
- $A_i = \{u_{i-m}\}$, $B_i = \{u_{i-m}\}$, $\gamma'_i = \gamma(1 - 1/p)/\Gamma$ for $i = m+1, \ldots, m + |I \cap J|$;
- $A_i = \{u_{i-(m+|I\cap J|)}\}$, $B_i = \varnothing$, $\gamma'_i = \gamma/(p\Gamma)$, for $i = m + |I \cap J| + 1, \ldots, m + 2|I \cap J|$.

We now verify that all sets A_i, B_i satisfy the required conditions. The property 1 follows from the fact that $|P_i| \le 2p$ and P_i alternates between sets $I \setminus J$ and $J \setminus I$. Therefore, $|A_i| \le p$ and $|B_i| \le p$ for $i \le m$; and $|A_i|, |B_i| \le 1 < p$ for $i > m$. Also by construction, $A_i \subset J$ and $B_i \subset I$.

To show Property 2, we use Lemma 19. We get for $i \le m$,

$$(J \setminus A_i) \cup B_i = J \triangle V(P_i) \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2).$$

For $i > m$, $(J \setminus A_i) \cup B_i \subset J$, hence $(J \setminus A_i) \cup B_i \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$.

Property 3 holds for $i \le m$, because $B_i \subseteq I \setminus J$ (see Definition 17) for $i \le m$. Property 3 holds for $i > m$, because $B_i \subseteq A_i$ for $i \le m$.

We now show Property 4,

$$\begin{aligned}\sum_i \gamma'_i \chi(A_i) &= \frac{\gamma}{\Gamma}\chi(J \setminus I) + \frac{\gamma(1-1/p)}{\Gamma}\chi(I \cap J) + \frac{\gamma/p}{\Gamma}\chi(I \cap J) \\ &= \frac{\gamma}{\Gamma}\chi(J) = \gamma'\chi(J).\end{aligned}$$

Similarly,

$$\begin{aligned}\sum_i \gamma'_i \chi(B_i) &= \frac{\gamma}{\Gamma}(1 - 1/p)\chi(I \setminus J) + \frac{\gamma(1-1/p)}{\Gamma}\chi(I \cap J) \\ &= (1 - 1/p)\frac{\gamma}{\Gamma}\chi(I) = (1 - 1/p)\gamma'\chi(I).\end{aligned}$$

Finally,

$$\sum_i \gamma'_i = \frac{1}{\Gamma}\Big(\sum_{i=1}^{m} \gamma_i + |I \cap J|\,\gamma\big(1 - \frac{1}{p}\big) + |I \cap J|\frac{\gamma}{p}\Big) = \frac{1 + \gamma|I \cap J|}{1 + \gamma|I \cap J|} = 1.$$

□

Lemma 22. *For every common independent set* $J \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$*, vector* $x^* \in \mathcal{P}(\mathcal{M}_1) \cap \mathcal{P}(\mathcal{M}_2)$ *and parameter* $p \in \{2, \ldots, r\}$*, there exists a probabilistic distribution of sets* (A, B)*,* $A \subset J$ *and* $B \subset V$*, a positive* $\alpha \ge 1/r$*, and vector* $\tilde{x} = (1 - 1/p)x^*$ *such that*

- $\pi(J) \equiv (J \setminus A) \cup B \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$;
- *for every* $u \notin J$, $\mathbf{Pr}[u \in \pi(J)] = \alpha \tilde{x}_u$;
- *for every* $u \in J$, $\mathbf{Pr}[u \notin \pi(J)] = \alpha(1 - \tilde{x}_u)$;
- $|A|, |B| \le p$; $A \neq \varnothing$.

Moreover, the distribution over pairs (A, B) *can be computed in polynomial-time.*

Proof. Represent x^* as a convex combination [29] of common independent sets $I_k \in \mathcal{I}(\mathcal{M}_1) \cap \mathcal{I}(\mathcal{M}_2)$ with coefficient λ_k:

$$x^* = \sum_k \lambda_k \chi(I_k).$$

Then, for every I_k, using Corollary 21, find a collection of pairs (A_{ki}, B_{ki}) such that

$$\sum_i \gamma_{ki} \chi(A_{ki}) = \gamma_k \chi(J) \text{ and } \sum_i \gamma_{ki} \chi(B_{ki}) = \gamma_k (1 - \frac{1}{p}) \chi(I_k).$$

Let

$$\Gamma = \sum_k \frac{\lambda_k}{\gamma_k}.$$

Pick a random $\hat{k}$ with probability $\lambda_{\hat{k}}/(\gamma_{\hat{k}}\Gamma)$, and then a random $\hat{i}$ with probability $\gamma_{\hat{k}\hat{i}}$. Thus, we pick a pair (k, i) with probability $\lambda_k \gamma_{ki}/(\gamma_k \Gamma)$. Output $(A, B) = (A_{\hat{k}\hat{i}}, B_{\hat{k}\hat{i}})$, $\pi(J) = (J \setminus A) \cup B$.

We now compute $\mathbf{Pr}[u \in \pi(J)]$. If $u \notin J$, then

$$\begin{aligned}
\mathbf{Pr}[u \in \pi(J)] &= \mathbf{Pr}[u \in B] = \mathbf{E}\left[\chi_u(B_{\hat{k}\hat{i}})\right] \\
&= \sum_k \frac{\lambda_k}{\gamma_k \Gamma} \Big(\sum_i \gamma_{ki} \chi_u(B_{ki}) \Big) \\
&= \sum_k \frac{\lambda_k}{\gamma_k \Gamma} \times \gamma_k (1 - \frac{1}{p}) \chi_u(I_k) \\
&= \frac{1}{\Gamma}(1 - \frac{1}{p}) \sum_k \lambda_k \chi_u(I_k) = \frac{1}{\Gamma}(1 - \frac{1}{p}) x_u^*.
\end{aligned}$$

If $u \in J$, then, similarly,

$$\begin{aligned}
\mathbf{Pr}[u \notin \pi(J)] &= \mathbf{Pr}[u \in A \setminus B] = \mathbf{E}\left[\chi_u(A_{\hat{k}\hat{i}} \setminus B_{\hat{k}\hat{i}})\right] \\
&= \sum_k \frac{\lambda_k}{\gamma_k \Gamma} \Big(\sum_i \gamma_{ki} \chi_u(A_{ki} \setminus B_{ki}) \Big).
\end{aligned}$$

If $u \in B_{ki}$, then $u \in A_{ki}$, because $B_{ki} \cap (J \setminus A_{ki}) = \varnothing$ (see Corollary 21, item 3) and $u \in J$. Hence, $\chi_u(A_{ki} \setminus B_{ki}) = \chi_u(A_{ki}) - \chi_u(B_{ki})$ and

$$\begin{aligned}
\mathbf{Pr}[u \notin \pi(J)] &= \sum_k \frac{\lambda_k}{\gamma_k \Gamma} \Big(\sum_i \gamma_{ki} (\chi_u(A_{ki}) - \chi_u(B_{ki})) \Big) \\
&= \sum_k \frac{\lambda_k}{\gamma_k \Gamma} \times \gamma_k \Big(\chi_u(J) - (1 - \frac{1}{p}) \chi_u(I_k) \Big) \\
&= \frac{1}{\Gamma} \sum_k \lambda_k \Big(\chi_u(J) - (1 - \frac{1}{p}) \chi_u(I_k) \Big) = \frac{1}{\Gamma}(1 - (1 - \frac{1}{p}) x_u^*).
\end{aligned}$$

Now, we estimate $\alpha = 1/\Gamma$. It is easy to see that for every $u \in J$,

$$\mathbf{Pr}\left[u \in A\right] = \sum_k \frac{\lambda_k}{\gamma_k \Gamma}\Big(\sum_i \gamma_{ki}\chi_u(A_{ki})\Big) = \frac{\chi_u(J)}{\Gamma} = \frac{1}{\Gamma}.$$

Since, $A \neq \varnothing$, we derive $\mathbf{E}|A| \geq 1$. On the other hand, $\mathbf{E}|A| = \alpha r$. Therefore, $\alpha \equiv 1/\Gamma \geq 1/r$. □

Lemma 5. *For every common independent set $J \in \mathcal{I}(\mathcal{M}_1)\cap\mathcal{I}(\mathcal{M}_2)$, vector $x^* \in \mathcal{P}(\mathcal{M}_1)\cap\mathcal{P}(\mathcal{M}_2)$, parameter $p \in \{2,\dots,r\}$, and $\tilde{x} = (1-1/p)x^*$ there exists a probabilistic distribution of sets $\pi(J) \in \mathcal{I}(\mathcal{M}_1)\cap\mathcal{I}(\mathcal{M}_2)$, such that*

- *for every $u \in J$, $\mathbf{Pr}\left[u \notin \pi(J)\right] = (1-\tilde{x}_u)/r$;*
- *for every $u \notin J$, $\mathbf{Pr}\left[u \in \pi(J)\right] = \tilde{x}_u/r$;*
- *$|J \triangle \pi(J)| \leq 2p$.*

Moreover, the distribution over sets $\pi(J)$ can be computed in polynomial-time (i.e. there is a polynomial number of sets with non-zero probabilities and a polynomial time algorithm to compute such non-zero probabilities).

Proof. The algorithm generates sets A, B as in Lemma 22. Then, it outputs $\pi(J) = (J \setminus A)\cup B$ with probability $(\alpha r)^{-1}$; and $\pi(J) = J$ with probability $1-(\alpha r)^{-1}$. □

B On Theorem 14

In this section, we explain why Theorem 14 easily follows from Theorem 3.15 (page 224), McDiarmid [22].

Theorem 23 (see Theorem 3.15 in [22])**.** *Let $x(t)$ (where $t \in T$) be a martingale w.r.t. the filtration $\mathcal{F}(t)$. Suppose that $\Delta x(t) = x(t+\Delta t) - x(t) \leq b$ a.s. for a (nonrandom) constant b. Then for any $\lambda \geq 0$ and $v \geq 0$,*

$$\mathbf{Pr}\big[x(1) - \mathbf{E}\left[x(1)\right] \geq \lambda \text{ and } \mathcal{V} \leq v\big] \leq e^{-\frac{\lambda^2}{2v(1+(b\lambda/(3v)))}},$$

where the random variable $\mathcal{V}$ (the predictable quadratic variation of $x(t)$) is the sum of conditional variances

$$\mathcal{V} = \sum_{t\in T}\mathbf{Var}\left[\Delta x(t) \mid \mathcal{F}(t)\right] = \sum_{t\in T}\mathbf{E}\left[(\Delta x(t) - \mathbf{E}\left[\Delta x(t) \mid \mathcal{F}(t)\right])^2 \mid \mathcal{F}(t)\right].$$

We derive an easy corollary (Theorem 14) which is convenient for our purposes.

Proof of Theorem 14. As in the rest of the paper, we assume that $T = \{t_0 + k\Delta t : 0 \leq k \leq N\}$ for some N and $\Delta t = (1-t_0)/N$. Let $\mu(t) = \sum_{t'\in T:t'<t}\Delta\mu(t')$ and $y(t) = x(t) - \mu(t)$. Observe, that

$$\begin{aligned}\mathbf{Pr}\left[\max_{t\in T}(x(t)) - \sum_{t\in T}\Delta\mu(t) \geq \lambda \text{ and } \mathcal{V} \leq v\right] &\leq \mathbf{Pr}\left[\max_{t\in T}(x(t) - \sum_{t'\in T:t'<t}\Delta\mu(t')) \geq \lambda \text{ and } \mathcal{V} \leq v\right]\\ &= \mathbf{Pr}\left[\max_{t\in T} y(t) \geq \lambda \text{ and } \mathcal{V} \leq v\right].\end{aligned}$$

Since $\mathbf{E}\left[\Delta x(t) \;\mid\; \mathcal{F}(t)\right] \leq \Delta\mu(t)$, $y(t)$ is a supermartingale. Note that

$$\mathcal{V}_y = \sum_{t \in T} \mathbf{Var}\left[\Delta y(t) \;\mid\; \mathcal{F}(t)\right] = \sum_{t \in T} \mathbf{Var}\left[\Delta x(t) \;\mid\; \mathcal{F}(t)\right] = \mathcal{V},$$

because $\Delta\mu(t)$ is a nonrandom sequence. Define

$$y'(t) = \begin{cases} y(t), & \text{if } \max\{y(t') : t' \in T; t' \leq t\} < \lambda; \\ \lambda, & \text{otherwise.} \end{cases}$$

It is easy to see that $y'(t+\Delta t) - y'(t) \leq y(t+\Delta t) - y(t)$, thus $y'(t)$ is also a supermartingale, and

$$y'(t+\Delta t) - y'(t) \leq y(t+\Delta t) - y(t) \leq x(t+\Delta t) - x(t) \leq b.$$

For $\Delta y'(t) = y(t'+\Delta t) - y'(t)$,

$$\mathcal{V}_{y'} = \sum_{t \in T} \mathbf{Var}\left[\Delta y'(t) \;\mid\; \mathcal{F}(t)\right] \leq \sum_{t \in T} \mathbf{Var}\left[\Delta y(t) \;\mid\; \mathcal{F}(t)\right] = \mathcal{V}.$$

We have

$$\begin{aligned} \mathbf{Pr}\left[\max_{t \in T} y(t) \geq \lambda \text{ and } \mathcal{V} \leq v\right] &= \mathbf{Pr}\left[y'(1) = \lambda \text{ and } \mathcal{V} \leq v\right] \\ &\leq \mathbf{Pr}\left[y'(1) = \lambda \text{ and } \mathcal{V}_{y'} \leq v\right]. \end{aligned}$$

It is now sufficient to show that

$$\mathbf{Pr}\left[y'(1) = \lambda \text{ and } \mathcal{V}_{y'} \leq v\right] \leq e^{-\frac{\lambda^2}{2v(1+(b\lambda/(3v)))}}.$$

We would like to apply Theorem 23. The process $y'(t)$ satisfies all the conditions of Theorem 23 except $y'(t)$ is not a martingale, but a supermartingale. However, we can slightly increase the values of $\Delta y'(t)$, so that $y''(t)$ is a martingale. To do so, we pick a threshold $\xi(t) \leq 0$ (which is a $\mathcal{F}(t)$ measurable random variable) so that $\mathbf{E}\left[\Delta y''(t) \;\mid\; \mathcal{F}(t)\right] = 0$, where

$$\Delta y''(t) = \max(y'(t), \xi(t)).$$

Such $\xi(t)$ exists since $\mathbf{E}\left[\Delta y'(t) \;\mid\; \mathcal{F}(t)\right] \leq 0$, but $\mathbf{E}\left[\max(\Delta y'(t), 0) \;\mid\; \mathcal{F}(t)\right] \geq 0$. Now, $y''(t) = \sum_{t' \in T : t' < t} \Delta y''(t)$ is a martingale. We have $\Delta y''(t) \leq \max(\Delta y'(t), 0) \leq b$, and

$$\mathcal{V}_{y''} = \sum_{t \in T} \mathbf{Var}\left[\Delta y''(t) \;\mid\; \mathcal{F}(t)\right] \leq \sum_{t \in T} \mathbf{Var}\left[\Delta y'(t) \;\mid\; \mathcal{F}(t)\right] \leq \mathcal{V}_{y'}.$$

So, by Theorem 23,

$$\mathbf{Pr}\left[y'(1) = \lambda \text{ and } \mathcal{V}_{y'} \leq v\right] \leq \mathbf{Pr}\left[y''(1) = \lambda \text{ and } \mathcal{V}_{y''} \leq v\right] e^{-\frac{\lambda^2}{2v(1+(b\lambda/(3v)))}}.$$

$\square$