Approximation algorithms via randomized rounding: a survey

Aravind Srinivasan*

Abstract

Approximation algorithms provide a natural way to approach computationally hard problems. There are currently many known paradigms in this area, including greedy algorithms, primal-dual methods, methods based on mathematical programming (linear and semidefinite programming in particular), local improvement, and "low distortion" embeddings of general metric spaces into special families of metric spaces. Randomization is a useful ingredient in many of these approaches, and particularly so in the form of *randomized rounding* of a suitable relaxation of a given problem. We survey this technique here, with a focus on correlation inequalities and their applications.

1 Introduction

It is well-known that several basic problems of discrete optimization are computationally intractable (NP-hard or worse). However, such problems do need to be solved, and a very useful practical approach is to design a heuristic tailor-made for a particular application. Another approach is to develop and implement approximation algorithms for the given problem, which come with *proven* guarantees. Study of the approximability of various classes of hard (combinatorial) optimization problems has greatly bloomed in the last two decades. In this survey, we study one important tool in this area, that of *randomized rounding*. We have not attempted to be encyclopaedic here, and sometimes do not present the best-known results; our goal is to present some of the underlying principles of randomized rounding without necessarily going into full detail.

For our purposes here, we shall consider an algorithm to be efficient if it runs in polynomial time (time that is bounded by a fixed polynomial of the length of the input). We will not explicitly present the running times of our algorithms; it will mostly be clear from the context that they are polynomial-time. However, we remark that most of our algorithms are indeed reasonably efficient; for instance, most of them involve network flows and other special cases of linear programming, for which efficient algorithms and codes are available. Our

*Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974-0636, USA. This article was written when the author was with the School of Computing, National University of Singapore, Singapore 119260. Research supported in part by National University of Singapore Academic Research Fund Grant RP970607.

focus throughout will be on the *quality of approximation* guaranteed by our approximation algorithms; we now recall the notion of an approximation algorithm.

Given an optimization problem $\mathcal{P}$ and an instance I of $\mathcal{P}$, let $OPT(I)$ denote the optimal objective-function value for I; each feasible solution for I will have a non-negative objective-function value, for all problems $\mathcal{P}$ studied here. If $\mathcal{P}$ is a *maximization* problem, an approximation algorithm $\mathcal{A}$ for $\mathcal{P}$ is an efficient algorithm that, for some $\lambda \geq 1$, produces a feasible solution of value at least $OPT(I)/\lambda$ for all instances I. $\mathcal{A}$ is called a λ-*approximation algorithm* for $\mathcal{P}$; λ is the *approximation guarantee* or *approximation bound* of $\mathcal{A}$. To maintain the convention that $\lambda \geq 1$ for minimization problems also, we define an algorithm $\mathcal{A}$ to be a λ-approximation algorithm for a *minimization* problem $\mathcal{P}$, if $\mathcal{A}$ produces a feasible solution of value at most $OPT(I) \cdot \lambda$ for all instances I of $\mathcal{P}$. For all problems considered, the goal will be to to develop polynomial-time algorithms with improved (smaller) approximation guarantees.

Thus, the two key phrases in approximation algorithms are *efficiency* and *proven approximation guarantees*. As mentioned in the abstract, various paradigms have been developed in designing approximation algorithms; furthermore, beautiful connections between error-correcting codes, interactive proof systems, complexity classes and approximability have been established, showing that some basic problems (such as finding a maximum independent set in a given graph) are hard to approximate within any "reasonable" factor. From the viewpoint of computational complexity, an interesting observation that has resulted from the study of approximation algorithms is that though NP-complete problems are "equivalent" in the sense of *exact* solvability, their ("natural") optimization versions turn out to lie in a wide spectrum in terms of approximability. The reader is referred to Hochbaum [32] for a comprehensive collection of articles discussing positive and negative results on approximation algorithms; the chapter in it by Motwani, Naor & Raghavan discusses randomized approximation algorithms [55]. Also, the survey by Shmoys [65] is a good source for work on approximation algorithms via linear programming.

This survey will focus on one useful approach in designing approximation algorithms: *randomized rounding*. Recall the classical notion of a *relaxation* of an optimization problem $\mathcal{P}$: given an instance I of $\mathcal{P}$, we *enlarge* the set of feasible solutions for I in such a way that the objective function can be efficiently optimized over the enlarged set. Let x^* denote an (efficiently computed) optimal solution over the enlarged set. Randomized rounding refers to the use of randomization to map x^* back to a solution that is indeed feasible for I. Let y^* denote the optimal objective function value on the enlarged set; i.e., the objective function value at x^*. It is easily seen that $y^* \geq OPT(I)$ (resp., $y^* \leq OPT(I)$) for maximization (resp., minimization) problems $\mathcal{P}$. Thus, suppose $\mathcal{P}$ is a maximization problem, say, and that we can analyze our randomized rounding process to show that it results in a feasible solution with objective function value at least y^*/λ (in expectation or with high probability); thus, since $y^* \geq OPT(I)$, the result is a λ-approximation (in expectation or with high probability). The situation is similar for minimization problems. Why the term "rounding"? One reason is that the relaxation often involves relaxing integrality constraints on variables (such as "$x_i \in \{0, 1\}$") to their real analogs ("$x_i \in [0, 1]$"). Thus, potentially non-integral values will

have to be *rounded* to appropriate integers, by the randomized algorithm.

Let us dive in by presenting two elegant examples of randomized rounding. Though these do not lead to the currently best-known results, they demonstrate the power and elegance of the method. In the sequel, $\mathbf{E}[X]$ will denote the expected value of a random variable X, and $\Pr[A]$ will denote the probability of event A. To appreciate our two examples, let us recall: (i) that if X is a random variable taking values in $\{0,1\}$, then $\mathbf{E}[X] = \Pr[X = 1]$, (ii) that the *uniform distribution* on a finite set S places equal probability $(1/|S|)$ on each element of S; the uniform distribution on a finite real interval $[a,b)$ has density function $1/(b-a)$, and (iii) *linearity of expectation*: for any finite number of arbitrary random variables $X_1, X_2, \ldots$, $\mathbf{E}[\sum_i X_i] = \sum_i \mathbf{E}[X_i]$.

(a) $s-t$ cuts in graphs. This result is from Teo [73]. For some classical integer programming problems, it is known that they are well-characterized by a natural linear programming formulation, i.e., that the linear and integral optima are the same. One famous example is the following *min-cut* problem. Given an undirected graph $G = (V, E)$ with a cost $c_{i,j}$ on each edge $\{i,j\}$ and two different distinguished vertices s and t, the problem is to remove a minimum-cost set of edges from G such that s gets disconnected from t. In other words, we want a minimum-cost "cut" (partition of V into two sets) separating s and t.

Let us formulate the problem in a natural way as an integer linear program (ILP). Given a cut that separates s and t, let $x_i = 0$ for all vertices i that are reachable from s, and $x_i = 1$ for all other vertices. (Thus, $x_s = 0$ and $x_t = 1$.) Let $z_{i,j}$ be the indicator variable for edge $\{i,j\}$ crossing the cut; so $z_{i,j} = |x_i - x_j|$. Thus, we have an integer program

$$\text{minimize} \sum_{\{i,j\} \in E} c_{i,j} z_{i,j} \text{ subject to}$$

$$\begin{aligned}
&\forall \{i,j\} \in E,\ z_{i,j} \geq (x_i - x_j) \ \text{ and } z_{i,j} \geq (x_j - x_i); \\
&x_s = 0 \ \text{ and } x_t = 1; && (1) \\
&\forall i\ \forall j, \quad x_i, z_{i,j} \in \{0,1\}. && (2)
\end{aligned}$$

It is easy to check that in any *optimal* solution to this integer program, we will have $z_{i,j} = |x_i - x_j|$; hence, this is indeed a valid formulation. Now, suppose we relax (2) to $\forall i \forall j,\ x_i, z_{i,j} \in [0,1]$. Thus, we get a linear program (LP) and as seen above, its optimal value y^* is a lower bound on the optimal objective function value OPT of the above integer program.

Now, it is well-known that in fact we have $y^* = OPT$; we now give a quick proof via randomized rounding. Let

$$\{x_i^*, z_{i,j}^* \in [0,1] :\ i,j \in V, \{i,j\} \in E\}$$

denote an optimal solution to the LP relaxation; once again, $z_{i,j}^* = |x_i^* - x_j^*|$ holds for all $\{i,j\} \in E$. Pick a $u \in [0,1)$ using the uniform distribution; for each $i \in V$, define $x_i := 0$ if $x_i^* \leq u$, and $x_i := 1$ otherwise. This is our "randomized rounding" process here. Note that constraint (1) is indeed satisfied by this method. What is the quality (cost) of the cut

produced? Fix any $\{i,j\} \in E$. This edge will cross the cut iff $u \in [\min\{x_i^*, x_j^*\}, \max\{x_i^*, x_j^*\})$; this happens with probability $|x_j^* - x_i^*|$, i.e., $z_{i,j}^*$. Thus, $\mathbf{E}[z_{i,j}] = z_{i,j}^*$ and hence by linearity of expectation,

$$\mathbf{E}[\sum_{\{i,j\}\in E} c_{i,j} z_{i,j}] = \sum_{\{i,j\}\in E} c_{i,j}\mathbf{E}[z_{i,j}] = \sum_{\{i,j\}\in E} c_{i,j} z_{i,j}^* = y^*. \tag{3}$$

The random variable $\sum_{\{i,j\}\in E} c_{i,j} z_{i,j}$ is piecewise continuous as a function of u (with only finitely many break-points). So, by (3), there must be some value of u that leads to an integral solution of cost no more than y^*! Thus $y^* = OPT$. See Teo & Sethuraman [74] for applications of more sophisticated versions of this idea to the stable matching problem.

(b) Maximum satisfiability (MAX-SAT). This is a natural optimization version of the satisfiability problem. Given a Boolean formula F in conjunctive normal form and a non-negative weight w_i associated with each clause C_i, the objective is to find a Boolean assignment to the variables that maximizes the total weight of the satisfied clauses. This problem is clearly NP-hard. An approximation algorithm that always produces a solution of weight at least 3/4th of the optimal weight OPT, had been proposed by Yannakakis [77]; we now describe a simpler algorithm of Goemans & Williamson that matches this [27].

The idea is to consider two different randomized schemes for constructing the Boolean assignment, and observe that they have *complementary* strengths in terms of their approximation guarantees. The first works well when each clause has "several" literals, while the second will be good if each clause has "few" literals. Thus, we could run both and take the better solution; in particular, we could choose one of the two schemes uniformly at random, and the resulting solution's objective function value will be the arithmetic mean of the respective solutions of the two schemes. Let us now present these simple schemes and analyze them.

The first scheme is to set each variable uniformly at random to True or False, independent of the other variables. Let $|C_i|$ denote the *length* of (i.e., number of literals in) clause C_i. It is easy to check that

$$p_{i,1} = \Pr[C_i \text{ satisfied}] = 1 - 2^{-|C_i|}; \tag{4}$$

hence, this scheme works well if all clauses are "long".

However, it is intuitively clear that such an approach may not work well for all CNF formulae: those with most clauses being short, in particular. Our second scheme is to start with a (fairly obvious) integer programming formulation. For each clause C_i, let $P(i)$ denote the set of unnegated variables appearing in it, and $N(i)$ be the set of negated variables in it. For each variable j, let $x_j = 1$ if this variable is set to True, and $x_j = 0$ if the variable is set to False. Letting $z_i \in \{0,1\}$ be the indicator for clause C_i getting satisfied, we have the constraint

$$\forall i,\ z_i \leq (\sum_{j\in P(i)} x_j) + (\sum_{j\in N(i)} (1 - x_j)). \tag{5}$$

Subject to these constraints, the objective is to maximize $\sum_i w_i z_i$. It is easy to check that this is a correct formulation for the problem on hand.

As above, suppose we take the LP relaxation obtained by relaxing each x_j and z_i to be a *real* in $[0,1]$, subject to (5). Then, y^* is an upper bound on OPT. Let $\{x_j^*, z_i^*\}$ be the values of the variables in an optimal solution to the LP. The key is to interpret each x_j^* as a *probability* [58]. Thus, our randomized rounding process will be, independently for each j, to set $x_j := 1$ (i.e., make variable j True) with probability x_j^* and $x_j := 0$ (make variable j False) with probability $1 - x_j^*$. One intuitive justification for this is that if x_j^* were "high", i.e., close to 1, it may be taken as an indication by the LP that it is better to set variable j to True; similarly for the case where x_j is close to 0. This is our second rounding scheme; since we are using information provided by the LP optimum implicitly, the hope is that we may be using the given formula's structure better in comparison with our first scheme.

Let us lower-bound the probability of clause C_i getting satisfied. We can assume without loss of generality that all variables appear unnegated in C_i. Thus, by (5), we will have

$$z_i^* = \min\{ \sum_{j \in P(i)} x_j^*, 1\}.$$

Given this, it is not hard to check that $\Pr[C_i \text{ satisfied}] = 1 - \prod_{j \in P(i)}(1 - x_j^*)$ is minimized when each x_j^* equals $z_i^*/|C_i|$. Thus,

$$p_{i,2} = \Pr[C_i \text{ satisfied}] \geq 1 - (1 - z_i^*/|C_i|)^{|C_i|}. \tag{6}$$

For a fixed value of z_i^*, the term $1-(1-z_i^*/|C_i|)^{|C_i|}$ decreases monotonically as $|C_i|$ increases. This is the sense in which our two schemes are complementary.

So, as mentioned above, suppose we choose one of the two schemes uniformly at random, in order to balance their strengths. Then,

$$\begin{aligned} \Pr[C_i \text{ satisfied}] &= (p_{i,1} + p_{i,2})/2 \\ &\geq 1 - (2^{-|C_i|} + (1 - z_i^*/|C_i|)^{|C_i|})/2 \\ &\geq (3/4)z_i^*, \end{aligned}$$

via elementary calculus and the fact that $z_i^* \in [0,1]$. (For any fixed positive integer ℓ, $f(\ell, x) = 1 - (2^{-\ell} + (1 - x/\ell)^\ell)/2 - 3x/4$ has a non-positive derivative for $x \in [0,1]$. Thus, it suffices to show that $f(\ell, 1) \geq 0$ for all positive integers ℓ. We have $f(1,1) = f(2,1) = 0$. For $\ell \geq 3$, $2^{-\ell} \leq 1/8$ and $(1 - 1/\ell)^\ell \leq 1/e$; so $f(\ell, 1) \geq 0$ for $\ell \geq 3$.) So by linearity of expectation, the expected weight of the final Boolean assignment is at least $\sum_i (3/4) w_i z_i^* = (3/4)y^* \geq (3/4)OPT$. Thus, at least in expectation, the assignment produced has a good approximation ratio; this randomized algorithm can be "derandomized" (turned into an efficient deterministic algorithm) as shown in [27].

Can this analysis be improved? No; we cannot guarantee any bound larger than $(3/4)y^*$. To see this, consider, e.g., the situation where we have two variables x_1 and x_2, all 4 clauses that can be formed using both variables, and with all the w_i being 1. It is easy to see that $y^* = 4$ and $OPT = 3$ here. Thus, there are indeed situations where $OPT = (3/4)y^*$. This also shows a simple approach to decide if a certain relaxation-based method is best possible

using that particular relaxation: to show that there are situations where the gap between the relaxed optimum and the optimum is (essentially) as high as the approximation bound guaranteed for that method. The worst-case ratio between the fractional and integral optima is called the *integrality gap*: clearly, a straightforward LP-based method cannot guarantee an approximation ratio better than the integrality gap.

In the $s-t$ cut example, we saw an instance of *dependent randomized rounding*: no two x_i are independent. In contrast, the underlying random choices were made *independently* in the MAX-SAT example; this is the basic approach we will focus on for most of this survey. The general situation here is that for some "large" N, we must choose, for $i = 1, 2, \ldots, N$, x_i from a finite set S_i (in order to optimize some function of the x_i, subject to some constraints on the choices). The randomized rounding approach will be to choose each x_i *independently* from S_i, according to some distribution D_i. This approach can be broadly classified as follows. First, as in our first scheme in the MAX-SAT example, we could take the arguably simplest approach–in this context–of letting each D_i be the uniform distribution on S_i; we call this *uniform randomized rounding*. This approach, while most natural for some problems, is in general inferior to choosing all the D_i through linear programming (as in the second scheme above in the MAX-SAT example), which we term *LP-based randomized rounding*.

The rest of this survey is organized as follows. We begin with some preliminaries in Section 2. In Section 3, we study approximation algorithms for packet-routing and job-shop scheduling that use uniform randomized rounding. Sections 4, 5 and 6 then consider LP-based randomized rounding. In Section 4, we show how a useful correlation inequality, the FKG inequality, can be applied to derive improved approximation algorithms for a class of routing, packing and covering problems. Section 5 considers a family of situations where correlations complementary to those handled by the FKG inequality arise. In Section 6, we study applications of a recent extension of the Lovász Local Lemma. Section 7 sketches how many such rounding problems are instances of questions in *discrepancy theory*, and briefly describes some seminal work of Beck & Fiala, Beck, Spencer and Banaszczyk in this area. A brief sketch of some other relaxation methods and open questions are presented in Section 8, which concludes.

Due to the immense current interest in communication networks and routing algorithms, many optimization problems we consider will be based on routing and scheduling. Two other basic themes of much of this survey are: (a) the use of various correlation inequalities in analyzing randomized rounding processes, and (b) their applications for ILPs with *column-sparse* coefficient matrices in particular. (Roughly speaking, a matrix A is column-sparse if we can bound the maximum number of nonzero entries in, or the L_1 norm of, any column of A by an appropriate parameter.)

2 Preliminaries

Let "r.v." denote the phrase "random variable"; we will be concerned only with discrete-valued r.v.s here. Given a non-negative integer k, $[k]$ will denote the set $\{1, 2, \ldots, k\}$; logarithms are to the base two unless specified otherwise. Let Z_+ denote the set of non-negative

integers. Given an event $\mathcal{E}$, its *indicator random variable* $\chi(\mathcal{E})$ is defined to be 1 if $\mathcal{E}$ holds, and 0 otherwise. Note that $\mathbf{E}[\chi(\mathcal{E})] = \Pr[\mathcal{E}]$.

Informally, a randomized algorithm is an algorithm equipped with a random number generator that can output any desired number of unbiased (fair) and, more importantly, *independent* random bits. The algorithm requests any number of random bits needed for its computation, from the source; we may charge, e.g., unit time to get one random bit. As seen in the introduction, it is also convenient to be able to draw an r.v. that is uniformly distributed over some finite real interval $[a, b)$ or some finite set; we shall assume that our random source has such features also, when necessary. The "random bits" model can typically approximate these requirements with sufficient accuracy in the context of efficient randomized algorithms.

There are several advantages offered by randomness to computation. See the books of Alon, Spencer & Erdős [4] and Motwani & Raghavan [56] for various aspects and applications of randomized computation.

2.1 Large deviation bounds

We now present a basic class of results which are often needed in randomized computation. These are (upper) bounds on the probability of certain types of r.v.s deviating significantly from their mean, and hence are known as *large deviation* or *tail probability* bounds. We shall present just a few of these results here that are relevant to randomized algorithms.

For the rest of this subsection, we let μ denote $\mathbf{E}[X]$ for a random variable X. One of the most basic tail inequalities is Markov's inequality, whose proof is left as an exercise.

Lemma 2.1 (Markov's inequality) *If an r.v. X takes on only* non-negative *values, then for any $a > 0$, $\Pr[X \geq a] \leq \mu/a$.*

Markov's inequality is best-possible if no further information is given about the r.v. X. However, it is often too weak, e.g., when we wish to upper bound the *lower tail* probability, or probability of staying much *below* the mean, of an r.v. X, even if we know good upper and lower bounds on the extent of its support. More information about the r.v. X of interest can lead to substantially stronger bounds, as we shall soon see. Nevertheless, Markov's inequality still underlies the approach behind many of these bounds. A well-known such strengthening of Markov's inequality is *Chebyshev's inequality*, which can be applied if we can upper bound the *variance* of X, $\mathbf{E}[(X - \mu)^2]$, well. Recall that the positive square root of the variance of X is known as the *standard deviation* of X.

Lemma 2.2 (Chebyshev's inequality) *For any r.v. X and any $a > 0$, $\Pr[|X - \mu| \geq a] \leq \sigma^2/a^2$, where σ denotes the standard deviation of X.*

If X is a sum of independent r.v.s, each of which lies in $[0, 1]$, then it is not hard to verify that $\mathbf{E}[(X - \mu)^2] \leq \mathbf{E}[X]$. Thus we get

Corollary 2.1 *Let X be a sum of independent r.v.s, each of which lies in $[0, 1]$. Then, for any $a > 0$, $\Pr[|X - \mu| \geq a] \leq \mu/a^2$.*

Hence, r.v.s X with a small standard deviation have good tail behavior. But this can be strengthened much more, for a class of r.v.s X that occur often in randomized computation: sums of *independent* r.v.s X_i, where each $X_i \in [0,1]$. The idea here is to observe that

$$\Pr[X \geq a] = \Pr[e^{tX} \geq e^{ta}] \leq \mathbf{E}[e^{tX}]/e^{ta}$$

for any $t > 0$, where the last step follows from Markov's inequality. We may then (approximately) minimize the last ratio over $t > 0$, to get strong bounds. Note also that this approach holds for any r.v. X, not just for sums of independent random variables. Similarly, for $a < \mu$, we can upper bound $\Pr[X \leq a]$ by $\min_{t>0} \mathbf{E}[e^{-tX}]/e^{-ta}$.

Let $X_1, X_2, \ldots X_n$ be *independent* r.v.s that take on values in $[0,1]$, with $\mathbf{E}[X_i] = p_i$, $1 \leq i \leq n$. Let $X \doteq \sum_{i=1}^{n} X_i$, and $\mu \doteq \mathbf{E}[X] = \sum_{i=1}^{n} p_i$. We want good upper bounds on $\Pr[X \geq \mu(1+\delta)]$, for $\delta > 0$; we recall some such bounds now, as presented in [61]. Chernoff [18] showed that for identically distributed $\{0,1\}$ r.v.s $X_1, X_2, \ldots, X_n$ and for $a > \mu$,

$$\min_t \frac{\mathbf{E}[e^{tX}]}{e^{at}} \leq \Lambda(n, \mu, a) = (\frac{\mu}{a})^a (\frac{n-\mu}{n-a})^{n-a}.$$

Hoeffding [33] extended this by showing that $\Lambda(n, \mu, a)$ is an upper bound for the above minimum even if the X_i's are not identically distributed and range between 0 and 1. Replacing a with $\mu(1+\delta)$ in the Hoeffding estimate $\Lambda(\cdot,\cdot,\cdot)$ gives, for $\delta \geq 0$,

$$\Pr[X \geq \mu(1+\delta)] \leq F(n, \mu, \delta) \doteq \frac{(1 + \frac{\mu\delta}{(n-\mu(1+\delta))})^{n-\mu(1+\delta)}}{(1+\delta)^{\mu(1+\delta)}}. \tag{7}$$

For $\delta \in [0,1]$, Hoeffding's approach also gives

$$\Pr[X \leq \mu(1-\delta)] = \Pr[n - X \geq n - \mu(1-\delta)] \leq F(n, \mu, -\delta) \doteq \frac{(1 - \frac{\mu\delta}{(n-\mu(1-\delta))})^{n-\mu(1-\delta)}}{(1-\delta)^{\mu(1-\delta)}}. \tag{8}$$

The following related results are useful (see, e.g., [33, 5, 4, 56]).

$$\text{If } \delta \geq 0,\ \Pr[X \geq \mu(1+\delta)] \leq \frac{\prod_{i\in[n]} \mathbf{E}[(1+\delta)^{X_i}]}{(1+\delta)^{\mu(1+\delta)}} \leq G(\mu, \delta) \doteq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu; \tag{9}$$

$$\text{If } \delta \in [0,1],\ \Pr[X \leq \mu(1-\delta)] \leq \frac{\prod_{i\in[n]} \mathbf{E}[(1-\delta)^{X_i}]}{(1-\delta)^{\mu(1-\delta)}} \leq H(\mu, \delta) \doteq e^{-\mu\delta^2/2}. \tag{10}$$

It can be checked that for $\delta \leq 1$, $G(\mu, \delta) \leq e^{-\delta^2\mu/3}$; for $\delta > 1$, $G(\mu, \delta) \leq e^{-(1+\delta)\ln(1+\delta)\mu/4}$.

Remark 2.1 It is useful to be conversant with these bounds; it is especially convenient to remember that: $G(\mu, \delta)$ (i) decays exponentially in $\mu\delta^2$ for "small" δ ($\delta \leq 1$), and (ii) decays exponentially in $\mu(1+\delta)\ln(1+\delta)$ for larger δ ($\delta > 1$). Also, some of the constants such as 3 and 4 in the exponents above, can be improved slightly.

We next present a useful result from [61], which offers a new look at the Chernoff-Hoeffding (CH) bounds. For real x and any positive integer r, let $\binom{x}{r} \doteq \frac{x(x-1)\cdots(x-r+1)}{r!}$ as usual, with $\binom{x}{0} \doteq 1$. Define, for $z = (z_1, z_2, \ldots, z_n) \in \Re^n$, a family of symmetric polynomials $\psi_j(z), j = 0, 1, \ldots, n$, where $\psi_0(z) \equiv 1$, and for $1 \leq j \leq n$,

$$\psi_j(z) \doteq \sum_{1 \leq i_1 < i_2 \cdots < i_j \leq n} z_{i_1} z_{i_2} \cdots z_{i_j}.$$

A small extension of a result of [61] is:

Theorem 2.1 ([61]) *Given r.v.s $X_1, \ldots, X_n \in [0,1]$, let $X = \sum_{i=1}^n X_i$ and $\mu = E[X]$. (a) For any $\delta > 0$, any nonempty event Z and any $k \leq \mu(1+\delta)$, $\Pr[X \geq \mu(1+\delta)|Z] \leq \mathbf{E}[Y_k|Z]$, where $Y_k = \psi_k(X_1, \ldots, X_n)/\binom{\mu(1+\delta)}{k}$. (b) If the X_is are independent and $k = \lceil \mu\delta \rceil$, then $\Pr[X \geq \mu(1+\delta)] \leq \mathbf{E}[Y_k] \leq G(\mu, \delta)$.*

Proof: Suppose $b_1, b_2, \ldots b_n \in [0,1]$ satisfy $\sum_{i=1}^n b_i \geq a$. It is not hard to show that for any non-negative integer $k \leq a$, $\psi_k(b_1, b_2, \ldots, b_n) \geq \binom{a}{k}$. (This is immediate if each b_i lies in $\{0,1\}$, and takes a little more work if $b_i \in [0,1]$.) Now this holds even conditional on any positive probability event Z. Hence,

$$\Pr[X \geq \mu(1+\delta)|Z] \leq \Pr[Y_k \geq 1|Z] \leq \mathbf{E}[Y_k|Z],$$

where the second inequality follows from Markov's inequality. See [61] for a proof of (b). ■

3 Uniform randomized rounding

We start our discussion in Section 3.1 with a basic random process that is related to many situations in uniform randomized rounding. More sophisticated versions of this process are then used in Section 3.2 to present approximation algorithms for job-shop scheduling, and in Section 3.4 for a version of packet routing (which is an important special case of job-shop scheduling).

3.1 Simple balls-and-bins processes

Consider the situation where each of n given balls is thrown uniformly at random, *independently*, into one of n bins. (We can view this as throwing darts at random.) Let X denote the maximum number of balls in any bin. It is often useful to know facts about (or bounds on) various statistics of X. For instance, it is known that $\mathbf{E}[X] = (1 + o(1)) \ln n / \ln \ln n$, where the $o(1)$ term goes to 0 as $n \to \infty$. Here, we content ourselves with showing that X is not much above this expectation, with high probability.

Let $Y_{i,j}$ be the indicator random variable for ball i being thrown into bin j. Then, the random variable X_j that denotes the number of balls in bin j, equals $\sum_i Y_{i,j}$. (We have

$X = \max_j X_j$.) For any given i, j, $\mathbf{E}[Y_{i,j}] = \Pr[Y_{i,j} = 1] = 1/n$. Thus, by linearity of expectation, $\mathbf{E}[X_j] = n \cdot (1/n) = 1$. Now since $X_j = \sum_i Y_{i,j}$ is a sum of bounded and *independent* r.v.s, we have by the CH bounds that for any $\delta > 1$,

$$\Pr[X_j \geq (1+\delta)] \leq e^{-(1+\delta)\ln(1+\delta)/4},$$

as seen before. The reader can verify that for any given constant $c_1 > 0$, there is some constant $c_2 > 0$ such that if we take $\delta \geq c_2 \log n / \log\log n$, then $e^{-(1+\delta)\ln(1+\delta)/4} \leq n^{-c_1}$. Thus, for a suitable $\delta = \Theta(\log n / \log\log n)$, we can ensure that

$$\Pr[X_j \geq (1+\delta)] \leq n^{-c_1}, \tag{11}$$

for each j and for any desired fixed $c_1 > 0$.

But recall that we really want to bound $\Pr[X \geq (1+\delta)]$. To do so, let us introduce yet another basic but very useful inequality. Suppose we are given some events $E_1, E_2, \ldots, E_m$, and wish to upper-bound $\Pr[\bigvee_i E_i]$. (We will later consider some such situations where it will suffice to show that $\Pr[\bigvee_i E_i] \neq 1$.) A simple approach for this is to use the *union bound* or *Boole's inequality*:

$$\Pr[\bigvee_{i \in [m]} E_i] \leq \sum_{i \in [m]} \Pr[E_i], \tag{12}$$

with equality holding if and only if the events E_i are pairwise mutually exclusive. The union bound is often used when correlations are hard to analyze. Of course, this is often not tight (and useless if $\sum_i \Pr[E_i] \geq 1$), and we shall see some tighter inequalities later. But even the union bound is of reasonable use sometimes, as we now illustrate.

Returning to our balls-and-bins, let $c_1 > 1$ in (11); say $c_1 = 2$. Now,

$$\begin{aligned}
\Pr[X \geq (1+\delta)] &= \Pr[\bigvee_{j \in [n]} (X_j \geq (1+\delta))] \\
&\leq \sum_{j \in [n]} \Pr[X_j \geq (1+\delta)] \text{ (union bound)} \\
&\leq n \cdot n^{-c_1} \text{ (by (11))} \\
&= n^{1-c_1},
\end{aligned}$$

which is quite small for $c_1 > 1$.

The reader is referred to [41, 30, 19] for other such useful results about this and related processes.

3.2 Approximation algorithms for job-shop scheduling

We now show a more involved application of the above approach to job-shop scheduling, which is a classical NP-hard minimization problem [44]. In it, we have n jobs and m machines. A job consists of a sequence of operations, each of which is to be processed on a specific machine for a specified integral amount of time; a job can have more than one operation on

a given machine. The operations of a job must be processed in the given sequence, and a machine can process at most one operation at any given time. The problem is to schedule the jobs so that the *makespan*, the time when all jobs have been completed, is minimized. An important special case is *preemptive* scheduling, wherein machines can suspend work on operations, switch to other operations, and later resume the suspended operations; this is often reasonable, for instance, in scheduling jobs in operating systems. Note that in the preemptive setting, all operation lengths may be taken to be one. If pre-emption is not allowed, we have the hard non-preemptive case, which we study here.

More formally, a job-shop scheduling instance consists of jobs $J_1, J_2, \ldots, J_n$, machines $M_1, M_2, \ldots, M_m$, and for each job J_j, a sequence of ℓ_j *operations* $(M_{j,1}, t_{j,1})$, $(M_{j,2}, t_{j,2}), \ldots,$ $(M_{j,\ell_j}, t_{j,\ell_j})$. Each operation is a (machine, processing time) pair: each $M_{j,k}$ represents some machine M_i, and the pair $(M_{j,i}, t_{j,i})$ signifies that the corresponding operation of job J_j must be processed on machine $M_{j,i}$ for an *uninterrupted* integral amount of time $t_{j,i}$. A machine can process at most one operation at any time, and the operations of each job must be processed in the given order.

Even some very restricted special cases of job-shop scheduling are NP-hard. Furthermore, while the theory of NP-completeness shows all NP-complete problems to be "equivalent" in a sense, NP-optimization problems display a wide range of difficulty in terms of exact solvability in practice; some of the hardest such problems come from job-shop scheduling. Define a job-shop instance to be *acyclic* if no job has two or more operations that need to run on any given machine. A single instance of acyclic job-shop scheduling with 10 jobs, 10 machines and 100 operations resisted attempts at exact solution for 22 years until its resolution [17]. See also Applegate & Cook [6]. We will show here that good *approximation* algorithms do exist for job-shop scheduling.

There are two natural lower bounds on the makespan of any job-shop instance: $P_{\max}$, the maximum total processing time needed for any job, and $\Pi_{\max}$, the maximum total amount of time for which any machine has to process operations. For the NP-hard special case of acyclic job-shop scheduling wherein all operations have unit length, the amazing result that a schedule of makespan $O(P_{\max} + \Pi_{\max})$ always exists, was shown in [46]. (We shall sketch the approach of [46] in Section 3.4.) Such a schedule can also be computed in polynomial time [47]. Can such good bounds hold if we drop any one of the two assumptions of acyclicity and unit operation lengths? See [23] for some recent advances on questions such as this.

Returning to general job-shop scheduling, let $\mu = \max_j \ \ell_j$ denote the maximum number of operations per job, and let $p_{\max}$ be the maximum processing time of any operation. By invoking ideas from [46, 62, 63] and by introducing some new techniques, good approximation algorithms were developed in [66]; their deterministic approximation bounds were slightly improved in [61]. (Recall that an approximation algorithm is an efficient algorithm that always produces a feasible solution that is to within a certain guaranteed factor of optimal.)

Theorem 3.1 ([66, 61]) *There is a deterministic polynomial-time algorithm that delivers a schedule of makespan* $O((P_{\max} + \Pi_{\max}) \cdot \frac{\log(m\mu)}{\log\log(m\mu)} \cdot \log(\min\{m\mu, p_{\max}\}))$ *for general job-shop scheduling.*

This has been improved further in [29]; we just discuss some of the main ideas behind Theorem 3.1 now. A key ingredient here is a "random delays" technique due to [46], and its intuition is as follows. To have a tractable analysis, suppose we imagine for now that each job J_j is run continuously after an initial wait of d_j steps: we wish to pick a suitable delay d_j from $S_j = \{0, 1, \ldots, B-1\}$ for an appropriate B, in such a way that the resulting "contentions" on machines is kept hopefully low. It seems a reasonable idea to conduct uniform randomized rounding, i.e., to pick each d_j uniformly and independently at random from $\{0, 1, \ldots, B-1\}$. In a manner similar to our dart-throwing analysis, we will then argue that, with high probability, not too many jobs contend for any given machine at the same time. We then resolve contentions by "expanding" the above "schedule"; the "low contention" property is invoked to argue that a small amount of such expansion suffices.

Let us proceed more formally now. A *delayed schedule* of a job-shop instance is constructed as follows. Each job J_j is assigned a delay d_j in $\{0, 1, \ldots, \Pi_{\max} - 1\}$. In the resulting "schedule", the operations of J_j are scheduled consecutively, starting at time d_j. Recall that all operation lengths are integral. For a delayed schedule $\mathcal{S}$, the *contention* $C(M_i, t)$ is defined to be the number of operations scheduled on machine M_i in the time interval $[t, t+1)$. A *randomly delayed schedule* is a delayed schedule wherein the delays are chosen independently and uniformly at random from $\{0, 1, \ldots, \Pi_{\max} - 1\}$.

Remark 3.1 Recall that $p_{\max}$ denotes the maximum processing time of any operation. It is shown in [66] that, in deterministic polynomial time, we can reduce the general shop-scheduling problem to the case where (i) $p_{\max} \leq n\mu$, and where (ii) $n \leq \text{poly}(m, \mu)$, while incurring an *additive* $O(P_{\max} + \Pi_{\max})$ term in the makespan of the schedule produced. By conducting these reductions, we assume from now on that $p_{\max}$ and n are bounded by $\text{poly}(m, \mu)$.

Lemma 3.1 *There is a sufficiently large constant $c_0 > 0$ such that the following holds. Let $\alpha = c_0 \log(m\mu)/\log\log(m\mu)$. Then, with high probability, a randomly delayed schedule satisfies:*

$$\forall i \in [m]\ \forall t \in \{0, 1, \ldots, P_{\max} + \Pi_{\max} - 1\},\ C(M_i, t) \leq \alpha.$$

Proof: Fix any positive integer k, and any M_i. For any set $U = \{u_1, u_2, \ldots, u_k\}$ of k units of processing that need to be done on M_i, let Collide(U) be the event that all these k units get scheduled at the same unit of time on M_i. (Our dart-throwing analysis using the CH bounds can also be used here, but we use a different, more direct, approach from [29] for variety.) Conditional on u_1 getting scheduled on M_i at any given time T, $\Pr[\text{Collide}(U)] \leq (1/\Pi_{\max})^{k-1}$; thus, unconditionally, $\Pr[\text{Collide}(U)] \leq (1/\Pi_{\max})^{k-1}$. As there are at most $\binom{\Pi_{\max}}{k}$ ways of choosing U, we apply the union bound over all of these to have

$$\Pr[\exists t :\ C(M_i, t) \geq k] \leq \binom{\Pi_{\max}}{k} (1/\Pi_{\max})^{k-1} \leq (\Pi_{\max})^k/(k!(\Pi_{\max})^{k-1}) = \Pi_{\max}/k!.$$

Now applying a union bound over all m possible values for i, we get

$$\Pr[\exists t\ \exists i:\ C(M_i,t) \geq k] \leq m\Pi_{\max}/k!.$$

But $\Pi_{\max} \leq n\mu p_{\max}$, which by our assumption of Remark 3.1 is $\mathrm{poly}(m,\mu)$. So the lemma holds if c_0 is sufficiently large. ∎

Thus, with high probability, the contention on all machines at all times is at most $\alpha = \mathrm{O}(\log(m\mu)/\log\log(m\mu))$. Now, if all operations were of the same (say unit) length, this can be converted into a valid schedule by stretching each time unit by a factor of α. The work of [66] shows how, even if the operations are of diverse lengths, this conversion can be accomplished by a suitable expansion of this schedule by a factor of $\mathrm{O}(\alpha \log(\min\{m\mu, p_{\max}\}))$; this leads to Theorem 3.1. To summarize, the natural random delays idea, an instance of uniform randomized rounding, thus leads to a simple and good approximation algorithm for a hard optimization problem.

3.3 Improved dart-throwing

In our balls-and-bins example, suppose we make the following small modification, which should intuitively only reduce the maximum number of balls in any bin. Suppose there are n balls and n bins, and let d be some positive integer. When each ball arrives, it picks d bins at random, and chooses to go to the one among these d that is least loaded at that point, breaking ties arbitrarily. (The case $d = 1$ corresponds to ordinary dart-throwing; so we just consider the case $d \geq 2$ now.) A remarkable fact shown in [7] is that the expected value of the maximum number of balls in any bin here, is just $\mathrm{O}((\ln\ln n)/\ln d) + \Theta(1)$. Note the significant improvement over ordinary dart-throwing, even for the case of $d = 2$. Such a result may naturally be expected to be algorithmically significant: applications to dynamic load-balancing and hashing are shown in [7]. Also see [37, 52] for related resource-allocation and hashing processes.

In light of this, a natural question may be whether there is a variant of "random initial delays" that leads to an improvement in the approximation bound for job-shop scheduling. However, by a random construction, it has been shown in [60] that there exist job-shop instances for which *whatever* initial delays we assign to each job, the contention on machines will be essentially that of Lemma 3.1, for many time instances. Thus, more sophisticated randomization may be necessary (if possible at all) to significantly improve the approximation bounds of [66, 29] for job-shop scheduling.

3.4 The Lovász Local Lemma and packet-routing

One of the most basic problems that arises in the design and analysis of randomized algorithms and in probabilistic combinatorics, is to get a good estimate (upper bound, lower bound, or both) of $\Pr[\bigvee_{i\in[m]} E_i]$, for some given events E_i. Equivalently, a good lower bound, upper bound or both, is required for $\Pr[\bigwedge_{i\in[m]} \overline{E_i}]$. As seen above, one approach would be

to use the union bound (12) which is unfortunately quite weak in general. Another obvious approach, which works when the events E_i are independent, is to use

$$\Pr[\bigwedge_{i\in[m]} \overline{E_i}] = \prod_{i\in[m]} \Pr[\overline{E_i}],$$

the "independence sieve".

A common situation where we have to look for such bounds is when the events E_i are "bad" events, *all* of which we wish to avoid simultaneously. The minimal requirement for this is that $\Pr[\bigwedge_{i\in[m]} \overline{E_i}] > 0$; however, even to prove this, the independence sieve will often not apply and the counting sieve could be quite weak. However, the independence sieve does suggest something: can we say something interesting if each E_i is independent of *most other* E_j? Let e denote the base of natural logarithms as usual. The powerful Lovász Local Lemma (LLL) due to Erdős and Lovász ([20]) is often very useful in such cases; see Chapter 5 of [4] for many such applications. The LLL (symmetric case) shows that all of a set of events E_i can be avoided under some conditions:

Lemma 3.2 ([20]) *Let $E_1, E_2, \ldots, E_m$ be any events with $\Pr[E_i] \leq p\ \forall i$. If each E_i is mutually independent of all but at most d of the other events E_j and if $ep(d+1) \leq 1$, then*

$$\Pr[\bigwedge_{i=1}^{m} \overline{E_i}] \geq (1-ep)^m > 0.$$

Why is the LLL considered such a powerful tool? For the union bound to be useful, we see from (12) that $\Pr[E_i]$, averaged over all i, must be less than $1/m$, which can be quite small if m is "large". In contrast, the LLL shows that even $\max_i \Pr[E_i]$ can be as high as $(e(d+1))^{-1}$, which is substantially bigger than $1/m$ if $d \ll m$ (say, if $d = \mathrm{O}(\mathrm{polylog}(m))$). Thus, under the "locality" assumption that $d \ll m$, we can get away with $\max_i \Pr[E_i]$ being much bigger than $1/m$. There is also a generalized ("asymmetric") form of the LLL. See [4] for this and related points.

We now present a portion of a surprising result on routing due to Leighton, Maggs & Rao, which makes involved use of the LLL in conjunction with uniform randomized rounding [46]. Suppose we have a routing problem on a graph, where the paths also have been specified for each of the packets. Concretely, we are given an undirected graph G, a set of pairs of vertices (s_i, t_i), and a path P_i in G that connects s_i with t_i, for each i. Initially, we have one packet residing at each s_i; this packet has to be routed to t_i using the path P_i. We assume that a packet takes unit time to traverse each edge, and the main constraint as usual is that an edge can carry at most one packet per (synchronous) time step. Subject to these restrictions, we want to minimize the maximum time taken by any packet to reach its destination. (If we view each edge as a machine and each packet as a job, this is just a special case of the job-shop scheduling problem considered in Section 3.2. We may further assume that the paths P_i are all *edge-simple*, i.e., do not repeat any edge; thus, we have an *acyclic* job-shop problem, in the notation of Section 3.2. Our objective here also is to keep the makespan small.) Since we are also given the paths P_i, the only question is the queuing discipline we

need to provide at the vertices (what a node must do if several packets that are currently resident at it, want to traverse the same edge on their next hop).

One motivation for assuming such a model (wherein the paths P_i are prespecified) is that many routing strategies can be divided into two phases: (a) choosing an intermediate destination for each packet (e.g., the paradigm of choosing the intermediate destination randomly and independently for each packet [75, 76]) and taking each packet on some canonical path to the intermediate destination, and (b) routing each packet on a canonical path from the intermediate destination to the final destination t_i. Such a strategy can thus use two invocations of the above "prespecified paths" model. Section 7 contains a brief description of recent work of [72] concerning situations where the paths P_i are not prespecified.

Let us study the objective function, *i.e.*, the maximum time taken by any packet to reach its destination. Two relevant parameters are the *congestion* c, the maximum number of paths P_i that use any given edge in G, and the *dilation* d, the length of a longest path among the P_i. It is immediate that each of c and d is a lower bound on the objective function. (The congestion and dilation are the respective analogs of $\Pi_{\max}$ and $P_{\max}$ here.) In terms of upper bounds, the simple greedy algorithm that never lets an edge go idle if it can carry a packet at a given time step, terminates within cd steps; this is because any packet can be delayed by at most $c-1$ other packets at any edge in the network.

Can we do better than cd? Recall our assumption that the paths P_i are all *edge-simple*. Under this assumption, the work of [46] shows the amazing result that there exists a schedule of length $O(c+d)$ with *bounded* queue-sizes at each edge, independent of the topology of the network or of the paths, and the total number of packets! The proof makes heavy use of the LLL, and we just show a portion of this beautiful result here.

Henceforth, we assume without loss of generality that $c = d$, to simplify notational cluttering such as $O(c+d)$. (However, we also use both c and d in places where such a distinction would make the discussion clear.) As in Section 3.2, imagine giving each packet a random initial delay, an integer chosen uniformly at random and independently from $\{1, 2, \ldots, ac\}$ for a suitable absolute constant $a > 1$. We think of each packet waiting out its initial delay period, and then traversing its path P_i without interruption to reach its delay. Of course this "schedule" is likely to be infeasible, since it may result in an edge having to carry more than one packet at a time step. Nevertheless, the LLL can be used to show that some such "schedules" exist with certain very useful properties, as we shall see now.

The above (infeasible) schedule clearly has a length of at most $ac+d$. Let us partition this period into *frames*, contiguous time intervals starting at time 1, with each frame having a length (number of time steps) of $b \ln c$ each, for a suitably large absolute constant b. Our basic idea is as follows. We shall prove, using the LLL, that every edge has a congestion of at most $b \ln c$ in each such frame, with positive probability. Suppose indeed that this is true; fix such a choice for the initial delays. Then, we would have essentially broken up the problem into at most $(ac+d)/(b \ln c)$ subproblems, one corresponding to each frame, wherein the congestion and the dilation are at most $b \ln c$ in each subproblem. Furthermore, we can solve each subproblem recursively and independently, and "paste together" the resulting schedules in an obvious way. Finally, the facts that:

- the congestion and dilation go from d to $O(\ln d)$, and
- a problem with constant congestion and dilation can be scheduled in constant time (*e.g.*, by the above-seen greedy protocol),

will guarantee a schedule of length $(c+d)\cdot 2^{O(\log^*(c+d))}$ for us. (Let $\log^{(k)}$ denote the logarithm iterated k times, i.e., $\log^{(1)} x = \log x$, $\log^{(2)} x = \log\log x$, etc. Recall that for $x > 0$, $\log^* x$ is the very slowly growing function that equals the smallest k for which $\log^{(k)} x$ is non-positive. Note that $\log^* x \leq 6$ for all practical purposes.)

We now use the LLL to prove that every edge has a congestion of at most $b \ln c$ in each frame, with positive probability. For any given edge f, let E_f denote the event that there is some frame in which it has a congestion of more than $b \ln c$. We want to show that $\Pr[\bigwedge_f \overline{E_f}] > 0$. For any given edge f and any given frame F, let $E'(f, F)$ denote the event that the congestion of f in F is more than $b \ln c$. It is easy to see that the expected congestion on any given edge at any given time instant, is at most $c/ac = 1/a$, in our randomized schedule. Thus, the expectation of the congestion $C(f, F)$ of f in F is at most $b(\ln c)/a$. Now, crucially, using our assumption that the paths P_i are *edge-simple*, it can be deduced that $C(f, F)$ is a sum of *independent* indicator random variables. Thus, by the Chernoff-Hoeffding bounds, we see that $\Pr[E'(f, F)] \leq c^{-b\ln(a/e)}$. Hence,

$$\Pr[E_f] \leq \sum_F \Pr[E'(f, F)] \leq O(c + d)c^{-b\ln(a/e)},$$

i.e., a term that can be made the reciprocal of an arbitrarily large polynomial in c, by just increasing the constants a and b appropriately. To apply the LLL, we also need to upper-bound the "dependency" among the E_f, which is easy: E_f "depends" only on those E_g where the edges f and g have a common packet that traverses both of them. By our definitions of c and d, we see that each E_f "depends" on at most $cd = c^2$ other events E_g; combined with our above upper-bound on each $\Pr[E_f]$, the LLL completes the result for us (by choosing a and b to be suitably large positive constants). Sophisticated use of similar ideas leads to the main result of [46]–the existence of a schedule of length $O(c+d)$ with constant-sized queues.

An interesting point is that the LLL usually only guarantees an extremely small (though positive) probability for the event being shown to exist. In the notation of Lemma 3.2, $p \gg 1/m$ in many applications, and hence the probability upper bound of $(1 - ep)^m$ is tiny, though positive. (For instance, in the applications of the LLL in the above routing result, p could be $(c + d)^{-\Theta(1)}$ while m could be $\Omega(N)$, where N is the total number of packets; note that c and d could be very small compared to N.) In fact, it can be proven in many applications of the LLL, that the actual probability of the desired structure occurring is very small. Thus, the LLL is often viewed as proving the existence of a *needle in a haystack*; this is in contrast to the much weaker union bound, which, when applicable, usually also gives a good probability of existence for the desired structure (thus leading to an efficient randomized or even deterministic algorithm to produce the structure). Breakthrough work of Beck [11] showed polynomial-time algorithms to produce several structures guaranteed to exist by the LLL; these results have been generalized by Alon [3] and Molloy & Reed [54].

While these techniques have been used to constructivize many known applications of the LLL, some applications still remain elusive (non-constructive).

For the above specific problem of routing using prespecified paths, polynomial-time algorithms have been presented in [47]. However, routing problems often need to be solved in a *distributed* fashion where each packet knows its source and destination and where there is no global controller who can guide the routing; significant progress toward such a distributed algorithmic analog of the above result of [46], has been made in [59]. Another interesting question is to remove the assumption on the paths P_i being edge-simple. While the edge-simplicity assumption seems natural for routing, it is not necessarily so if this routing problem is interpreted in the manner seen above as a job-shop scheduling problem.

4 Analyses based on the FKG inequality

We now move to approximation algorithms via LP-based randomized rounding. Section 4.1 presents some some classical work on LP-based randomized rounding for lattice approximation and packing problems [14, 58, 57]. Section 4.2 and its Theorem 4.3 in particular, then show a somewhat general setting under which certain "bad" events can all be simultaneously avoided; the well-known FKG inequality is one basic motivation behind Theorem 4.3. Theorem 4.3 is one of the main tools that we apply in Sections 4.3 and 4.4. In Section 4.3, we sketch improved approximation algorithms for certain families of NP-hard ILPs (packing and covering integer programs) due to [69]; these improve on the packing results shown in Section 4.1. Section 4.4 briefly discusses the work of [71, 42, 9] on disjoint paths and related packing and low-congestion routing problems.

4.1 Lattice approximation and packing integer programs

For a matrix A and vector v, let $(Av)_i$ denote the ith entry (row) of the vector Av. Given an $m \times n$ matrix $A \in [0,1]^{m \times n}$ and a vector $p \in [0,1]^n$, the *lattice approximation problem* is to efficiently come up with a *lattice point* $q \in \{0,1\}^n$ such that $\|Ap - Aq\|_\infty$ is "small" [57]. When $A \in \{0,1\}^{m \times n}$, this is also called the *linear discrepancy* problem (Lovász, Spencer & Vesztergombi [50]). What we want in this problem is a lattice point q which is "close" to p in the sense of $|Ap - Aq|_i$ being small for every row i of A. If $A \in \{0,1\}^{m \times n}$, this problem has interesting connections to the *discrepancy* properties of the hypergraph represented by A [50]; we will present some elements of discrepancy theory in Section 7. Furthermore, the lattice approximation problem is closely related to the problem of deriving good approximation algorithms for classes of ILPs. If we interpret one row of A as the objective function, and the remaining rows as defining the constraint matrix of an ILP, then this and related problems arise in the "linear relaxation" approach to ILPs:
(i) Solve the linear relaxation of the ILP efficiently using some good algorithm for linear programming;
(ii) View the process of "rounding" the resultant fractional solution to a good integral solution as a lattice approximation problem and solve it.

An efficient algorithm for the lattice approximation problem with a good approximation guarantee, translates to a good approximation algorithms for some classes of ILPs.

Let $Ap = b$. The LP-based randomized rounding approach to solve lattice approximation problems and their relatives, was proposed in [58, 57]: view each p_j as a probability, and thus round it to 0 or 1 with the appropriate probability, independently for each j. Thus, our output vector q is an r.v., with all of its entries being *independent*; for each j, $\Pr[q_j = 1] = p_j$ and $\Pr[q_j = 0] = 1 - p_j$. It is immediate that $\mathbf{E}[q_j] = p_j$ and hence, by linearity of expectation, we have $\mathbf{E}[(Aq)_i] = b_i$ for each i. Furthermore since each $(Aq)_i$ is a sum of *independent* r.v.s each taking values in $[0, 1]$, we see from the CH bounds that $(Aq)_i$ is quite likely to stay "close" to its mean b_i. However, the problem, of course, is that *some* $(Aq)_i$ could stray a lot from b_i, and we now proceed to bound the maximum extent of such deviations (with high probability).

Recall the function $F(\cdot, \cdot, \cdot)$ from the bounds (7) and (8). Let $\{\delta_i > 0 : \ i \in [m]\}$ and $0 < \epsilon < 1$ be such that

$$F(n, b_i, \delta_i) \leq \epsilon/(2m) \text{ and } F(n, b_i, -\delta_i) \leq \epsilon/(2m), \text{ for each } i \in [m];$$

it is clear that such a property should hold if we make each δ_i sufficiently large. For each $i \in [m]$, let E_i be the "bad" event "$|(Aq)_i - b_i| \geq b_i\delta_i$". Thus by the CH bounds and by the definition of $F(\cdot, \cdot, \cdot)$, we have

$$\Pr[E_i] = \Pr[(Aq)_i - b_i \geq b_i\delta_i] + \Pr[(Aq)_i - b_i \leq -b_i\delta_i] \leq \epsilon/(2m) + \epsilon/(2m) = \epsilon/m, \quad (13)$$

for each $i \in [m]$.

Note that we really want to upper-bound $\Pr[\bigvee_{i\in[m]} E_i]$. However, it looks hard to analyze the correlations among the E_i. Thus, using the union bound (12) with (13), we see that $\Pr[\bigvee_{i\in[m]} E_i] \leq m \cdot (\epsilon/m) = \epsilon$ and hence, our output q satisfies

$$\|q - b\|_\infty \leq d \doteq \max_{i\in[m]} b_i\delta_i,$$

with probability at least $1 - \epsilon$. To get a handle on d, let ϵ be a constant, say $1/4$, lying in $(0, 1)$. One can show, using the observations of Remark 2.1, that for each i,

$$b_i\delta_i \leq \mathrm{O}(\sqrt{b_i \ln m}) \text{ if } b_i \geq \ln m, \text{ and } b_i\delta_i \leq \mathrm{O}(\ln m / \ln((2 \ln m)/b_i)), \text{ otherwise.} \quad (14)$$

Thus, in particular if $b_i \geq \log m$ for each i, our approximation is very good. The bound of (14) can also be achieved in *deterministic* polynomial time by closely mimicking the above analysis [57]. A generalized version of this derandomization approach is presented by Theorem 4.3.

For the lattice approximation problem presented in this generality, bound (14) is essentially the best approximation possible. Better (*non-constructive*) bounds are known in some special cases: *e.g.*, if $m = n$, $A \in \{0, 1\}^{n\times n}$ and if $p_j = 1/2$ for all j, then there exists $q \in \{0, 1\}^n$ such that $\|Ap - Aq\|_\infty \leq \mathrm{O}(\sqrt{n})$ [67].

Two main ideas above where: (i) LP-based randomized rounding and (ii) identifying a set of bad events E_i, and applying the *union bound* in conjunction with large-deviation

inequalities to show that all the bad events are avoided with high probability. These ideas can be extended as follows to the setting of *packing integer programs* [57].

Packing integer programs. In a packing integer program (henceforth PIP), the variables are $x_1, x_2, \ldots, x_N$; the objective is to maximize $\sum_j w_j x_j$ where all the w_j are *non-negative*, subject to: (i) a system of m linear constraints $Ax \leq b$ where $A \in [0,1]^{m \times N}$, and (ii) integrality constraints $x_j \in \{0, 1, \ldots, d_j\}$ for each j (some of the given integers d_j could equal infinity). Furthermore, by suitable scaling, we can assume without loss of generality that: (a) all the w_j's lie in $[0,1]$, with $\max_j \ w_j = 1$; (b) if all entries of A lie in $\{0,1\}$, then each b_i is an integer; and (c) $\max_{i,j} \ A_{i,j} = 1$ and $B \doteq \min_i \ b_i \geq 1$. The parameter B will play a key role in the approximation bounds to be described.

PIPs model several NP-hard problems in combinatorial optimization. Consider, e.g., the NP-hard *B-matching problem on hypergraphs*. Given:

- a hypergraph (V, E) where $E = \{f_1, f_2, \ldots, f_N\}$ with each f_j being a subset of V;
- a positive integer B, and
- a non-negative weight w_j for each f_j (scaled w.l.o.g. so that $\max_j w_j = 1$),

the problem is to choose a maximum-weight collection of the f_j so that each element of V is contained in at most B of the chosen f_j. A well-known PIP formulation with the $|V| \times N$ coefficient matrix A having entries zero and one, is as follows. Let $x_j \in \{0,1\}$ denote the indicator for picking f_j. Then, we wish to maximize $\sum_j w_j x_j$ subject to

$$\sum_{j: \ i \in f_j} x_j \leq B, \quad i = 1, 2, \ldots, |V|.$$

Returning to general PIPs, we now present the approach of [57] for approximating a given PIP. (We have not attempted to optimize the constants here.) We show this result as presented in the full version of [69]. Solve the PIP's LP relaxation, in which each x_j is allowed to lie in $[0,1]$. Let $\{x_1^*, x_2^*, \ldots, x_N^*\}$ be the obtained optimal solution, with $y^* = \sum_j w_j x_j^*$. Recall that $B = \min_i b_i \geq 1$; here is an $O(m^{1/B})$-approximation algorithm. We first handle an easy case. If $y^* \leq 3e(5m)^{1/B}$, we just choose any j such that $w_j = 1$; set $x_j = 1$ and $x_k = 0$ for all $k \neq j$. This is clearly an $O(m^{1/B})$-approximation algorithm.

We next move to the more interesting case where $y^* > 3e(5m)^{1/B}$. A first idea, following the lattice approximation algorithm, may be to directly conduct LP-based randomized rounding on the x_j^*. However, we do need to satisfy the constraints $Ax \leq b$ here, and such direct randomized rounding could violate some of these constraints with probability very close to 1. Recall that all entries of the matrix A are non-negative; furthermore, all the constraints are "$\leq$" constraints. Thus, in order to have a reasonable probability of satisfying $Ax \leq b$, the idea is to first *scale down* each x_j^* appropriately, and then conduct randomized rounding. More precisely, we choose an appropriate $\gamma > 1$ and set $x'_j = x_j^*/\gamma$ for each j; then, independently for each j, we set $x_j := \lceil x'_j \rceil$ with probability $1 - (\lceil x'_j \rceil - x'_j)$, and set $x_j := \lfloor x'_j \rfloor$ with probability $\lceil x'_j \rceil - x'_j$.

Let us analyze the performance of this scheme. As in the lattice approximation analysis, we define an appropriate set of bad events and upper-bound the probability of at least one of them occurring via the union bound. Define $m+1$ bad events

$$E_i \equiv ((Ax)_i > b_i),\ i = 1, 2, \ldots, m; \quad E_{m+1} \equiv (\sum_j w_j x_j < y^*/(3\gamma)). \tag{15}$$

If we avoid all these $(m+1)$ events, we would have a feasible solution with objective function value at least $y^*/(3\gamma)$. So, we wish to choose γ just large enough in order to show that $\Pr[\bigvee_{i\in[m+1]} E_i] \leq 1 - \Omega(1)$, say. As in the analysis of the lattice approximation problem, the plan will be to use the union bound to show that $\sum_{i\in[m+1]} \Pr[E_i] \leq (1 - \Omega(1))$.

Choose

$$\gamma = e(5m)^{1/B}. \tag{16}$$

We can view each x_j as a sum of $\lceil x'_j \rceil$–many independent $\{0, 1\}$ r.v.s, where the first $\lceil x'_j \rceil - 1$ are 1 with probability 1, and the last is 1 with probability $1 - (\lceil x'_j \rceil - x'_j)$. For each $i \in [m]$, $(Ax)_i$ is a sum of independent r.v.s each of which takes on values in $[0, 1]$; also,

$$\mathbf{E}[(Ax)_i] = (Ax^*)_i/\gamma \leq b_i/\gamma.$$

For each $i \in [m]$, $\Pr[E_i] = \Pr[(Ax)_i > b_i]$ can be upper-bounded by a CH bound (the function G of (9)):

$$\Pr[(Ax)_i > b_i] \leq (e/\gamma)^{b_i} = (5m)^{-b_i/B} \leq 1/(5m), \tag{17}$$

since $b_i \geq B$. Similarly, $\Pr[E_{m+1}]$ can be upper-bounded by a lower tail bound. For our purposes here, even Chebyshev's inequality will suffice. Since $y^* > 3e(5m)^{1/B}$ now by assumption, we have $\mu \doteq \mathbf{E}[\sum_j w_j x_j] = y^*/\gamma \geq 3$. So

$$\Pr[\sum_j w_j x_j < \mu/3] \leq \Pr[|\sum_j w_j x_j - \mu| > 2\mu/3] \leq 9/(4\mu) \leq 3/4, \tag{18}$$

the last inequality following from Corollary 2.1. Thus, (17) and (18) show that

$$\sum_{i\in[m+1]} \Pr[E_i] \leq 1/5 + 3/4 = 0.95.$$

As usual, the probability of failure can be reduced by repeating the algorithm. Thus we have a randomized approximation algorithm with approximation guarantee $\mathrm{O}(\gamma) = \mathrm{O}(m^{1/B})$ for general PIPs.

Theorem 4.1 ([57]) *For any given PIP, a feasible solution with objective function value $\Omega(y^*/m^{1/B})$ can be efficiently computed.*

Is there a way to improve the analysis above? One possible source of slack above is in the application of the union bound: could we exploit the special structure of packing problems (all data is non-negative, all constraints are of the "$\leq$" type) to improve the analysis? The answer is in the affirmative; we now present the work of [69, 9] in a general context to provide the answer. Our main goal for now is to prove Theorem 4.3, which shows a useful sufficient condition for efficiently avoiding certain collections of "bad" events $E_1, E_2, \ldots, E_t$.

4.2 The FKG inequality and well-behaved estimators

For all of Section 4.2, we will take $X_1, X_2, \ldots, X_\ell$ to be *independent* r.v.s, each taking values in $\{0,1\}$. We will let $\vec{X} \doteq (X_1, X_2, \ldots, X_\ell)$, and all events and r.v.s considered in this subsection are assumed to be completely determined by the value of $\vec{X}$.

The powerful FKG inequality originated in statistical physics [24], and a special case of it can be summarized as follows for our purposes. Given $\vec{a} = (a_1, a_2, \ldots, a_\ell) \in \{0,1\}^\ell$ and $\vec{b} = (b_1, b_2, \ldots, b_\ell) \in \{0,1\}^\ell$, let us say that $\vec{a} \preceq \vec{b}$ iff $a_i \leq b_i$ for all i. Define an event $\mathcal{A}$ to be *increasing* iff: for all $\vec{a} \in \{0,1\}^\ell$ such that $\mathcal{A}$ holds when $\vec{X} = \vec{a}$, $\mathcal{A}$ also holds when $\vec{X} = \vec{b}$, for any $\vec{b}$ such that $\vec{a} \preceq \vec{b}$. Analogously, event $\mathcal{A}$ is said to be *decreasing* iff: for all $\vec{a} \in \{0,1\}^\ell$ such that $\mathcal{A}$ holds when $\vec{X} = \vec{a}$, $\mathcal{A}$ also holds when $\vec{X} = \vec{b}$, for any $\vec{b} \preceq \vec{a}$.

To motivate the FKG inequality, consider, for instance, the increasing events $U \equiv (X_1 + X_5 + X_7 \geq 2)$ and $V \equiv (X_7 X_8 + X_9 \geq 1)$. It seems intuitively plausible that these events are *positively correlated* with each other, i.e., that $\Pr[U|V] \geq \Pr[U]$. One can make similar other guesses about possible correlations among certain types of events. The FKG inequality proves a class of such intuitively plausible ideas. It shows that any set of increasing events are *positively correlated* with each other; analogously for any set of decreasing events. Similarly, any increasing event is negatively correlated with any set of decreasing events; any decreasing event is negatively correlated with any set of increasing events.

Theorem 4.2 (FKG inequality) *Let $I_1, I_2, \ldots, I_t$ be any sequence of* increasing *events and $D_1, D_2, \ldots, D_t$ be any sequence of* decreasing *events (each I_i and D_i completely determined by $\vec{X}$). Then for any $i \in [t]$ and any $S \subseteq [t]$,*
(i) $\Pr[I_i | \bigwedge_{j \in S} I_j] \geq \Pr[I_i]$ *and* $\Pr[D_i | \bigwedge_{j \in S} D_j] \geq \Pr[D_i]$*;*
(ii) $\Pr[I_i | \bigwedge_{j \in S} D_j] \leq \Pr[I_i]$ *and* $\Pr[D_i | \bigwedge_{j \in S} I_j] \leq \Pr[D_i]$.

See [26, 4] for further discussion about this and related correlation inequalities.

Well-behaved and proper estimators. Suppose $\mathcal{E}$ is some event (determined completely by $\vec{X}$, as assumed above). A random variable $g = g(\vec{X})$ is said to be a *well-behaved* estimator for $\mathcal{E}$ (w.r.t. $\vec{X}$) iff it satisfies the following properties (P1), (P2), (P3) and (P4), $\forall t \leq \ell$, $\forall T = \{i_1, i_2, \ldots, i_t\} \subseteq [\ell]$, $\forall b_1, b_2, \ldots, b_t \in \{0,1\}$; for notational convenience, we let $\mathcal{B}$ denote "$\bigwedge_{s=1}^{t}(X_{i_s} = b_s)$".

(P1) $\mathbf{E}[g|\mathcal{B}]$ is efficiently computable;

(P2) $\Pr[\mathcal{E}|\mathcal{B}] \leq \mathbf{E}[g|\mathcal{B}]$;

(P3) *if $\mathcal{E}$ is increasing*, then $\forall i_{t+1} \in ([\ell] - T)$, $\mathbf{E}[g|(X_{i_{t+1}} = 0) \wedge \mathcal{B}] \leq \mathbf{E}[g|(X_{i_{t+1}} = 1) \wedge \mathcal{B}]$; and

(P4) *if $\mathcal{E}$ is decreasing*, then $\forall i_{t+1} \in ([\ell] - T)$, $\mathbf{E}[g|(X_{i_{t+1}} = 1) \wedge \mathcal{B}] \leq \mathbf{E}[g|(X_{i_{t+1}} = 0) \wedge \mathcal{B}]$.

Taking g to be the indicator variable for $\mathcal{E}$ will satisfy (P2), (P3) and (P4), but not necessarily (P1). So the idea is that we want to approximate quantities such as $\Pr[\mathcal{E}|\mathcal{B}]$ "well" (in the sense of (P2), (P3) and (P4)) by an *efficiently computable* value ($\mathbf{E}[g|\mathcal{B}]$).

If g satisfies (P1) and (P2) (but not necessarily (P3) and (P4)), we call it a *proper* estimator for $\mathcal{E}$ w.r.t $\vec{X}$.

For any r.v. X and event $\mathcal{A}$, let $\mathbf{E'}[X]$ and $\mathbf{E'}[X|\mathcal{A}]$ respectively denote $\min\{\mathbf{E}[X], 1\}$ and $\min\{\mathbf{E}[X|\mathcal{A}], 1\}$. Let us say that a collection of events is of the *same type* if they are all increasing or are all decreasing. We start with a useful lemma from [69]:

Lemma 4.1 *Suppose $\vec{X} = (X_1, X_2, \ldots, X_\ell)$ is a sequence of* independent *r.v.s X_i, with $X_i \in \{0,1\}$ and $\Pr[X_i = 1] = p_i$ for each i. Let $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k$ be events* of the same type *with respective well-behaved estimators $h_1, h_2, \ldots, h_k$ w.r.t. $\vec{X}$ (all the A_i and h_i completely determined by $\vec{X}$). Then for any non-negative integer $t \leq \ell - 1$ and any $\vec{b} = (b_1, b_2, \ldots, b_t) \in \{0,1\}^t$,*

$$\prod_{i=1}^{k}(1 - \mathbf{E'}[h_i| \bigwedge_{j=1}^{t}(X_j = b_j)]) \leq (1 - p_{t+1}) \cdot \prod_{i=1}^{k}(1 - \mathbf{E'}[h_i|((X_{t+1} = 0) \wedge \bigwedge_{j=1}^{t}(X_j = b_j))]) + p_{t+1} \cdot \prod_{i=1}^{k}(1 - \mathbf{E'}[h_i|((X_{t+1} = 1) \wedge \bigwedge_{j=1}^{t}(X_j = b_j))]).$$

Proof: We only prove the lemma for the case where all the $\mathcal{E}_i$ are *increasing*; the proof is similar if all the $\mathcal{E}_i$ are decreasing.

For notational convenience, define, for all $i \in [k]$,

$$u_i = \mathbf{E}[h_i| \bigwedge_{j=1}^{t}(X_j = b_j)]); \quad u_i' = \min\{u_i, 1\};$$

$$v_i = \mathbf{E}[h_i|((X_{t+1} = 0) \wedge \bigwedge_{j=1}^{t}(X_j = b_j))]; \quad v_i' = \min\{v_i, 1\};$$

$$w_i = \mathbf{E}[h_i|((X_{t+1} = 1) \wedge \bigwedge_{j=1}^{t}(X_j = b_j))]; \quad w_i' = \min\{w_i, 1\}.$$

We start by showing that for all $i \in [k]$,

$$u_i' \geq (1 - p_{t+1}) \cdot v_i' + p_{t+1} \cdot w_i'. \tag{19}$$

To see this, note first that for all i,

$$0 \leq v_i \leq w_i \text{ (properties (P2) and (P3)), and} \tag{20}$$

$$u_i = (1 - p_{t+1}) \cdot v_i + p_{t+1} \cdot w_i. \tag{21}$$

If $u_i < 1$ and $w_i \leq 1$, then $v_i < 1$ by (21) and hence (19) follows from (21), with equality. If $u_i < 1$ and $w_i > 1$, note again that $v_i < 1$ and furthermore, that $w_i > w_i' = 1$; thus, (19) follows from (21). Finally if $u_i \geq 1$, then $u_i' = 1$, $w_i' = 1$ and $v_i' \leq 1$, implying (19) again.

Note that $u'_i \leq 1$ for all i. Thus, inequality (19) shows that to prove the lemma, it suffices to show that

$$\prod_{i=1}^{k}(1-(1-p_{t+1})v'_i - p_{t+1}w'_i) \leq (1-p_{t+1})\prod_{i=1}^{k}(1-v'_i) + p_{t+1}\prod_{i=1}^{k}(1-w'_i), \tag{22}$$

which we now prove by induction on k.

Equality holds in (22) for the base case $k = 1$. We now prove (22) by assuming its analog for $k-1$; i.e., we show that

$$((1-p_{t+1})\prod_{i=1}^{k-1}(1-v'_i) + p_{t+1}\prod_{i=1}^{k-1}(1-w'_i)) \cdot (1-(1-p_{t+1})v'_k - p_{t+1}w'_k)$$

is at most

$$(1-p_{t+1})\prod_{i=1}^{k}(1-v'_i) + p_{t+1}\prod_{i=1}^{k}(1-w'_i).$$

Simplifying, we need to show that

$$p_{t+1}(1-p_{t+1})(w'_k - v'_k)\left(\prod_{i=1}^{k-1}(1-v'_i) - \prod_{i=1}^{k-1}(1-w'_i)\right) \geq 0, \tag{23}$$

which is validated by (20). ∎

As seen in many situations by now, suppose $E_1, E_2, \ldots, E_t$ are all "bad" events: we would like to find an assignment for $\vec{X}$ that avoids all of the E_i. We now present an approach of [69, 9] based on the method of conditional probabilities [21, 68, 57] that shows a sufficient condition for this.

Theorem 4.3 *Suppose $\vec{X} = (X_1, X_2, \ldots, X_\ell)$ is a sequence of* independent *r.v.s X_i, with $X_i \in \{0,1\}$ for each i. Let $E_1, E_2, \ldots, E_t$ be events and r, s be non-negative integers with $r+s \leq t$ such that:*

- *$E_1, E_2, \ldots, E_r$ are all* increasing*, with respective well-behaved estimators $g_1, g_2, \ldots, g_r$ w.r.t. $\vec{X}$;*
- *$E_{r+1}, \ldots, E_{r+s}$ are all* decreasing*, with respective well-behaved estimators $g_{r+1}, \ldots, g_{r+s}$ w.r.t. $\vec{X}$;*
- *$E_{r+s+1}, \ldots, E_t$ are arbitrary events, with respective proper estimators $g_{r+s+1}, \ldots, g_t$, and*
- *all the E_i and g_i are completely determined by $\vec{X}$.*

Then if

$$1-(\prod_{i=1}^{r}(1-\mathbf{E'}[g_i]))+1-(\prod_{i=r+1}^{r+s}(1-\mathbf{E'}[g_i]))+\sum_{i=r+s+1}^{t}\mathbf{E}[g_i]<1 \tag{24}$$

holds, we can efficiently construct a deterministic assignment for $\vec{X}$ *under which none of* $E_1, E_2, \ldots, E_t$ *hold. (As usual, empty products are taken to be* 1*; e.g., if* $s = 0$*, then* $\prod_{i=r+1}^{r+s}(1-\mathbf{E'}[g_i]) \equiv 1$*.)*

Proof: Though we will not require it, let us first show that $\Pr[\exists i \in [t] : E_i] < 1$. This will serve as a warm-up and is also meant to provide some motivation about the expression in the left-hand-side of (24).

We can upper-bound $\Pr[\exists i \in [r] : E_i] = 1 - \Pr[\bigwedge_{i=1}^{r} \overline{E_i}]$ as

$$\begin{aligned}
1-\Pr[\bigwedge_{i=1}^{r}\overline{E_i}] &\leq 1-\prod_{i=1}^{r}\Pr[\overline{E_i}] \text{ (FKG inequality: } \overline{E_1},\ldots,\overline{E_r} \text{ are all decreasing)}\\
&= 1-\prod_{i=1}^{r}(1-\Pr[E_i])\\
&\leq 1-\prod_{i=1}^{r}(1-\mathbf{E'}[g_i]);
\end{aligned}$$

the last inequality is a consequence of (P2). Similarly,

$$\Pr[\bigvee_{i=r+1}^{r+s}E_i] \leq 1-\prod_{i=r+1}^{r+s}(1-\mathbf{E'}[g_i]).$$

Also, for all i, $\Pr[E_i] \leq \mathbf{E}[g_i]$ by (P2). Thus,

$$\begin{aligned}
\Pr[\exists i \in [t] : E_i] &\leq \Pr[\bigvee_{i=1}^{r}E_i]+\Pr[\bigvee_{i=r+1}^{r+s}E_i]+\sum_{i=r+s+1}^{t}\Pr[E_i]\\
&\leq 1-(\prod_{i=1}^{r}(1-\mathbf{E'}[g_i]))+1-(\prod_{i=r+1}^{r+s}(1-\mathbf{E'}[g_i]))+\sum_{i=r+s+1}^{t}\mathbf{E}[g_i]\\
&< 1,
\end{aligned}$$

by (24). Thus, there exists a value of $\vec{X}$ that avoids all of $E_1, E_2, \ldots, E_t$.

How can we *efficiently* find such a value for $\vec{X}$? Let $p_i = \Pr[X_i = 1]$. For any $u \leq \ell$ and any $\vec{b} = (b_1, b_2, \ldots, b_u) \in \{0,1\}^u$, define a "potential function" $h(u, \vec{b})$ to be

$$2-(\prod_{i=1}^{r}(1-\mathbf{E'}[g_i|\bigwedge_{j=1}^{u}(X_j=b_j)]))-(\prod_{i=r+1}^{r+s}(1-\mathbf{E'}[g_i|\bigwedge_{j=1}^{u}(X_j=b_j)]))+\sum_{i=r+s+1}^{t}\mathbf{E}[g_i|\bigwedge_{j=1}^{u}(X_j=b_j)].$$

Note that for $u \leq \ell - 1$ and for each i, $\mathbf{E}[g_i|\bigwedge_{j=1}^{u}(X_j = b_j)]$ equals

$$(1-p_{u+1})\cdot\mathbf{E}[g_i|((X_{u+1}=0)\wedge\bigwedge_{j=1}^{u}(X_j=b_j))]+p_{u+1}\mathbf{E}[g_i|((X_{u+1}=1)\wedge\bigwedge_{j=1}^{u}(X_j=b_j))].$$

Combining this with Lemma 4.1, we see that if $u \leq \ell - 1$, then

$$h(u, \vec{b}) \geq (1 - p_{u+1}) \cdot h(u+1, (b_1, b_2, \ldots, b_u, 0)) + p_{u+1} \cdot h(u+1, (b_1, b_2, \ldots, b_u, 1)).$$

Thus,

$$\min_{v \in \{0,1\}} h(u+1, (b_1, b_2, \ldots, b_u, v)) \leq h(u, \vec{b}). \tag{25}$$

Consider the following algorithm, which starts with $\vec{b}$ initialized to $\perp$ (the empty list):

for $i := 1$ **to** ℓ **do**:
$b_i := \text{argmin}_{v \in \{0,1\}} h(i, (b_1, b_2, \ldots, b_{i-1}, v))$;
$\vec{b} := (b_1, b_2, \ldots, b_i)$.

Starting with the initial condition $h(0, \perp) < 1$ (see (24)) and using (25), it is easily seen by induction on i that we maintain the property $h(i, \vec{b}) < 1$. Setting $i = \ell$, we can check using (P2) that all of the E_i are avoided when $\vec{X} = (b_1, b_2, \ldots, b_\ell)$. ■

Note that Theorem 4.3 works even if $\Pr[\bigwedge_{i=1}^{t} \overline{E_i}]$ is very small. We also remark that in many of our applications of Theorem 4.3, one of r and s will be 0, the other equaling $t - 1$.

4.3 Correlations in packing and covering integer programs

We now employ the idea of positive correlation and Theorem 4.3 in particular, to sketch the approximation algorithms for PIPs and for covering integer programs (which will be introduced at the end of this subsection) due to [69]. Positive correlation has been used for network reliability problems by Esary & Proschan [22]; see, e.g., Chapter 4.1 in Shier [64]. It has also been used for the set cover problem, an important example of covering integer programs, by Bertsimas & Vohra [15].

Given any sequence $\vec{X} = (X_1, X_2, \ldots, X_\ell)$ of r.v.s, we will say that an event Z is an *assignment event* w.r.t. $\vec{X}$, iff Z is of the form "$\bigwedge_{i \in S}(X_i = b_i)$", for any $S \subseteq [\ell]$ and any sequence of values b_i. Suppose the X_i's are all *independent*. Then, as mentioned in [71, 9], even conditional on any assignment event Z, we can still view the X_i's as independent: it is simply that for all $i \in S$, $X_i = b_i$ with probability 1. This simple idea, which is called the "independence view" in [71, 9], will be very useful. For instance, suppose we need to show for some r.v. $g = g(\vec{X})$ that for any assignment event $\mathcal{A}$ w.r.t. X, we can efficiently compute $\mathbf{E}[g(\vec{X}) \mid \mathcal{A}]$. In all such situations for us here, g will be a sum of r.v.s of the form $\prod_{i=1}^{v} f_{u_i}(X_{u_i})$, where all the indices u_i are distinct. Thus, even conditional on $\mathcal{A}$, the independence view lets us compute the expectation of this term as $\prod_{i=1}^{v} \mathbf{E}[f_{u_i}(X_{u_i})]$; the understanding in this computation is that for all i such that $u_i \in S$, $\mathbf{E}[f_{u_i}(X_{u_i})] = f_{u_i}(b_{u_i})$. So it will suffice if each term $\mathbf{E}[f_{u_i}(X_{u_i})]$ is efficiently computable, which will always be the case in our applications.

Let us now return to the problem of approximating a general PIP. As before, we do the "scaling down by γ followed by LP-based randomized rounding" for a suitable $\gamma > 1$; the

key point now is that Theorem 4.3 will let us choose a γ that is significantly smaller than the $\Theta(m^{1/B})$ of Section 4.1. Once again, the situation where $y^* \leq 3\gamma$, say, can be handled easily: we can easily construct a feasible solution of objective function value 1, thus resulting in an $O(\gamma)$ approximation (whatever the value of γ we choose is).

So we assume that $y^* > 3\gamma$; let the bad events $E_1, E_2, \ldots, E_{m+1}$ be as in (15). Our plan is to efficiently avoid all of these via Theorem 4.3, while getting away with a γ that is much smaller than $\Theta(m^{1/B})$. Let us start with some notation. For $i \in [m]$, define $\mu_i = \mathbf{E}[(Ax)_i]$ and define $\delta_i \geq 0$ to be $b_i/\mu_i - 1$; let $\mu_{m+1} = \mathbf{E}[\sum_j w_j x_j] = y^*/\gamma$ and $\delta_{m+1} = 2/3$. Then, by (9) and (10),

$$\Pr[E_i] \leq \frac{\prod_j \mathbf{E}[(1+\delta_i)^{A_{i,j}x_j}]}{(1+\delta_i)^{\mu_i(1+\delta_i)}} \leq G(\mu_i, \delta_i), \ i = 1, 2, \ldots, m; \tag{26}$$

$$\Pr[E_{m+1}] \leq \frac{\prod_j \mathbf{E}[(1-\delta_{m+1})^{w_j x_j}]}{(1-\delta_{m+1})^{\mu_{m+1}(1-\delta_{m+1})}} \leq H(\mu_{m+1}, \delta_{m+1}). \tag{27}$$

We now make the useful observation that $E_1, E_2, \ldots, E_m$ are all *increasing* as a function of $\vec{x} = (x_1, x_2, \ldots, x_n)$. (Also, E_{m+1} is decreasing, but this will not be very useful for us right now.) By (26) and from the above discussion regarding the "independence view" we can check that for each $i \in [m]$,

$$g_i \doteq \frac{\prod_j (1+\delta_i)^{A_{i,j}x_j}}{(1+\delta_i)^{\mu_i(1+\delta_i)}}$$

is a well-behaved estimator for E_i (w.r.t. $\vec{x}$). Also, as in (17), we have

$$\mathbf{E}[g_i] \leq (e/\gamma)^B, \quad i = 1, 2, \ldots, m. \tag{28}$$

We can also check using (27) and the independence view that

$$g_{m+1} = \frac{\prod_j (1-\delta_{m+1})^{w_j x_j}}{(1-\delta_{m+1})^{\mu_{m+1}(1-\delta_{m+1})}}$$

is a proper estimator for E_{m+1}, with

$$\mathbf{E}[g_{m+1}] \leq e^{-2y^*/9}.$$

Setting $r = m$, $s = 0$ and $t = m+1$ in Theorem 4.3, we get

$$(1 - \min\{(e/\gamma)^B, 1\})^m > e^{-2y^*/9}$$

to be a sufficient condition for efficiently avoiding all of $E_1, E_2, \ldots, E_{m+1}$. We can check that for a suitably large constant $c_1' > 0$, $\gamma = (c_1' m/y^*)^{1/(B-1)}$ suffices to satisfy this sufficient condition. Thus, we can efficiently construct a feasible solution of value

$$\Omega(y^*/\gamma) = \Omega((c_1 y^*/m^{1/B})^{B/(B-1)},$$

where $c_1 > 0$ is an absolute constant. Note that for B bounded away (and of course greater than) 1, this is a good improvement on the $\Omega(y^*/m^{1/B})$ of Theorem 4.1.

An improvement can be made above in the special (but important) case where all entries of A lie in $\{0,1\}$; this is the case, e.g., for the B-matching problem on hypergraphs. If all entries of A lie in $\{0,1\}$, the bad event "$(Ax)_i > b_i$" is equivalent to "$(Ax)_i \geq b_i + 1$". So, (28) can be improved to $\mathbf{E}[g_i] \leq (e/\gamma)^{B+1}$: hence we can choose γ to be $\Theta((m/y^*)^{1/B})$ here, leading to an efficiently computable feasible solution of value

$$\Omega((y^*/m^{1/(B+1)})^{(B+1)/B}).$$

Thus we have

Theorem 4.4 ([69]) *There is an absolute constant $c_1 > 0$ such that for any given PIP, a feasible solution with objective function value $\Omega((c_1 y^*/m^{1/B})^{B/(B-1)}$ can be efficiently computed. If all entries in the coefficient matrix A of the PIP lie in $\{0,1\}$, we can improve this to objective function value $\Omega((y^*/m^{1/(B+1)})^{(B+1)/B})$.*

In particular, if OPT denotes the optimal objective function value for a given B-matching problem on a hypergraph (V,E), a solution of value $\Omega((OPT/|V|^{1/(B+1)})^{(B+1)/B})$ can be efficiently computed. For the special case where all the w_j are 1 and $B = 1$, this bound of $\Omega(OPT^2/|V|)$ had been obtained earlier in [1].

Thus, rather than use the union bound to upper-bound $\Pr[\exists i \in [m+1] : E_i]$ as in Section 4.1, exploitation of the (desirable) *correlations* involved has led to the improved approximations of Theorem 4.4. As shown in [69], similar observations can be made for the NP-hard family of *covering integer programs*, which are in a sense dual to PIPs. In a covering integer program, we seek to *minimize* a linear objective function $\sum_j w_j x_j$, subject to $Ax \geq b$ and integrality constraints on the x_j. Once again, all the data $(A_{i,j}, b_i, c_j)$ is non-negative; also, as in PIPs, note that all the constraints "point" in the same direction. The basic idea here is to solve the LP relaxation, scale *up* all the fractional values given by the LP (to boost the probability of satisfying $Ax \geq b$), and then to conduct randomized rounding. Once again, the bad events $E_1, E_2, \ldots, E_m$ corresponding to the constraints getting violated are of the *same type*: it is just that they are all decreasing now. Thus, Theorem 4.3 can be appropriately applied here also, leading to some improved approximation bounds; the reader is referred to [69] for further details.

4.4 Low-congestion routing and related problems

As mentioned before, much algorithmic attention has been paid recently to various types of routing problems, due to the growth of high-speed integrated networks. One broad type of routing paradigm that we studied in Section 3.4 is *packet switching* or *packet routing*, where atomic packets move through the network, getting queued occasionally. Another main routing paradigm is *circuit switching*, where a connection path from source to destination is established for an appropriate duration of time, for each routing request that is admitted. Motivated by our discussion on algorithms for packing problems via Theorem 4.3, we now

present some of the LP-based approximation algorithms of [71, 9] for a family of NP-hard circuit-switching problems. Similar problems model some routing issues in ATM networks.

Given an undirected graph $G = (V, E)$, we will let $n = |V|$ and $m = |E|$. The diameter of G (the maximum length of a shortest path between any pair of vertices in G) will be denoted $\mathrm{diam}(G)$. Suppose we are given G and a (multi-)set $\mathcal{T} = \{(s_i, t_i) : 1 \leq i \leq k\}$ of pairs of vertices of G. $\mathcal{T}' \subseteq \mathcal{T}$ is said to be *realizable* iff the pairs of vertices in $\mathcal{T}'$ can be connected in G by mutually edge-disjoint paths. The classical maximum disjoint paths problem (henceforth MDP) is to find a realizable sub-(multi-)set of $\mathcal{T}$ of maximum cardinality. This is one of the most basic circuit-switching problems and is NP-hard. The reader is referred to [43] for much combinatorial work on disjoint paths and related problems.

A natural generalization of the MDP can be considered for situations where each connection request comes with a different *demand* (bandwidth request), and where link capacities may not be uniform across the network. Thus, we allow each pair (s_i, t_i) to have a demand $\rho_i > 0$, and each edge f of G to have a capacity $c_f > 0$ [38]. $\mathcal{T}' \subseteq \mathcal{T}$ is called realizable here iff each pair of vertices in $\mathcal{T}'$ can be connected by one path in G, such that the total demand using any edge f does not exceed c_f. The *unsplittable flow problem* (UFP) is to find a realizable $\mathcal{T}' \subseteq \mathcal{T}$ that maximizes $\sum_{i:(s_i,t_i)\in\mathcal{T}'} \rho_i$ [38]. (The word "unsplittable" emphasizes the requirement that if we choose to connect s_i to t_i, then a *single path* must be used for the connection: the flow from s_i to t_i should not be split across multiple (s_i, t_i)-paths.) As in [38], we assume that $\forall i \; \forall f, \; \rho_i \leq c_f$. If all capacities c_f are the same, we call the problem *uniform-capacity* UFP (UCUFP); by scaling all the capacities and demands uniformly, we will take the common edge capacity to be 1 for the UCUFP. Note that the MDP is a special case of the UCUFP.

Prior to the work of [71], the best approximation guarantee for the MDP on arbitrary graphs G was $\mathrm{O}(\max\{\sqrt{m}, \mathrm{diam}(G)\})$ [38]. Also, an $\mathrm{O}(\max\{\sqrt{m}, \mathrm{diam}(G)\} \cdot (\min_i \; \rho_i^{-1}))$-approximation bound was known for the UCUFP [38]. However, for some important special classes of graphs, recent breakthrough results have led to good approximations: see [40, 38] and the references therein.

Let us in fact work with the general *weighted* case of the UFP, where each (s_i, t_i) has a weight $w_i > 0$, with the objective being to find a realizable $\mathcal{T}' \subseteq \mathcal{T}$ that maximizes $\sum_{i:(s_i,t_i)\in\mathcal{T}'} w_i$. (As we did for PIPs, we assume without loss of generality that $\max_i w_i = 1$.) Approximation guarantees for the UFP are presented in [9]; for simplicity, we shall only present approximations for the UCUFP here, following [71, 9]. Let OPT denote the optimal objective function value for a given UCUFP instance. Theorem 4.5 presents an efficient algorithm to construct a feasible path selection of value $\Omega(\max\{OPT^2/m, OPT/\sqrt{m}\})$. Even for the MDP, this is better than the above-seen $\Omega(OPT/\max\{\sqrt{m}, \mathrm{diam}(G)\})$ bound.

The approach, as for PIPs, will be to start with an appropriate LP relaxation, do a "scaling down by γ", consider a suitable randomized rounding scheme, and then invoke Theorem 4.3. However, the details of the randomized rounding will differ somewhat from our approach for PIPs.

We start a simple lemma presented without proof. For part (iii) of the lemma, please recall the functions ψ_i of Section 2.1; for a (multi-)set $S = \{a_1, a_2, \ldots, a_N\}$, we let $\psi_i(S)$

denote $\psi_i(a_1, a_2, \ldots, a_N)$.

Lemma 4.2 **(i)** *If $\mu_1(1+\delta_1) = \mu_2(1+\delta_2)$ where $\mu_1 \leq \mu_2$ and $\mu_1, \mu_2, \delta_1, \delta_2 \geq 0$, then $G(\mu_1, \delta_1) \leq G(\mu_2, \delta_2)$.* **(ii)** *If $\mu_1 \geq \mu_2 \geq 0$ and $\delta_1, \delta_2 \in [0,1]$ are such that $\mu_1(1-\delta_1) = \mu_2(1-\delta_2)$, then $H(\mu_1, \delta_1) \leq H(\mu_2, \delta_2)$.* **(iii)** *Suppose $a_1, a_2, \ldots, a_N$ are non-negative reals summing to at most b. Then, $\psi_2(a_1, a_2, \ldots, a_N) < b^2/2$.*

There is a natural ILP formulation for a given weighted instance $(G, \mathcal{T})$ of the UCUFP. Define variables $\{x_i \in \{0,1\} : i \in [k]\}$, where x_i will be 1 iff we connect s_i to t_i by a path. Also define variables $\{y_{i,f} \in \{0,1\} : i \in [k],\ f \in E\}$, where $y_{i,f}$ is 1 iff the path chosen (if any) for (s_i, t_i), passes through f. For each $f \in E$, we have the capacity constraint $\sum_i \rho_i y_{i,f} \leq 1$. Linear equalities relating the x_i's and $y_{i,f}$'s model, in an obvious way, the fact that x_i is the indicator for choosing an (s_i, t_i)-path. Subject to the above constraints, the objective is to maximize $\sum_i w_i x_i$. Relaxing each x_i and $y_{i,f}$ to be a real lying in $[0,1]$, we get a natural LP relaxation.

Henceforth, let $\{x_i^*, y_{i,f}^* \in [0,1] : i \in [k], f \in E\}$ denote an optimal solution to the LP relaxation, with objective function value $y^* = \sum_i w_i x_i^*$. As a first step, we transform this in polynomial time into a set of flow-paths, using the standard "flow decomposition" technique [2]. This results in a set $\{P_{i,1}, P_{i,2}, \ldots, P_{i,\ell_i}\}$ of (s_i, t_i)-paths for each i, where $\ell_i \leq m$. Each path $P_{i,j}$ carries a non-negative amount $\rho_i z_{i,j}^*$ of flow from s_i to t_i, where $x_i^* = \sum_j z_{i,j}^*$. The capacity constraints imply that for all edges f,

$$\sum_{(i,j):\ f \in P_{i,j}} \rho_i z_{i,j}^* \leq 1. \tag{29}$$

The main result now is Theorem 4.5, showing that an appropriate LP-based randomized rounding approach works well. The rounding process is related to our approach for PIPs. For an appropriate $\gamma > 1$, construct, independently for all i, j, a $\{0,1\}$ r.v. $z_{i,j}$ with $\Pr[z_{i,j} = 1] = z_{i,j}^*/\gamma$. For each pair (s_i, t_i), we will do the following. If at least one of the $z_{i,j}$'s is 1, we shall arbitrarily choose one of them, say $z_{i,k}$, and connect s_i to t_i by path $P_{i,k}$; otherwise if all the $z_{i,j}$'s are 0, we do not connect s_i to t_i. Let us analyze this process.

For any $f \in E$, the total demand requested of it is at most

$$D_f \doteq \sum_{(i,j):\ f \in P_{i,j}} \rho_i z_{i,j}. \tag{30}$$

D_f is an upper bound on the total demand finally using f, since even if some $z_{i,j}$ gets rounded to one, the path $P_{i,j}$ may not be selected if some other $z_{i,k}$ was also one. We have by (29) that

$$\mathbf{E}[D_f] = \sum_{(i,j):\ f \in P_{i,j}} \rho_i z_{i,j}^*/\gamma \leq 1/\gamma. \tag{31}$$

For any $f \in E$, let E_f denote the bad event that $D_f > 1$. We will show that by choosing γ judiciously, we can avoid all the bad events and keep the objective function reasonably high. Let $\mathcal{R} = \{z_{i,j} : i \in [k],\ j \in [\ell_i]\}$ denote the underlying collection of r.v.s. We plan to employ

Theorem 4.3; to this end, we start by constructing well-behaved and proper estimators (w.r.t. $\mathcal{R}$) for the events that concern us.

We start by constructing well-behaved estimators for the events E_f. It will often help to treat the "small" and "large" demands separately. First suppose $\rho_i \leq 1/2$ for all i. Consider any edge $f \in E$. Define $\rho'_i = 2\rho_i$ and $D'_f \doteq \sum_{(i,j):\ f \in P_{i,j}} \rho'_i z_{i,j}$. Since $\rho'_i \in [0,1]$ for all i, D'_f is a sum of independent r.v.s, each taking values in $[0,1]$. Also, $\mu \doteq \mathbf{E}[D'_f] \leq 2/\gamma$, by (31). Let $\delta = 2/\mu - 1$, and

$$u_f \doteq \frac{\prod_{(i,j):\ f \in P_{i,j}} (1+\delta)^{\rho'_i z_{i,j}}}{(1+\delta)^2}.$$

Note that $E_f \equiv (D'_f > 2)$. Using (9), part (i) of Lemma 4.2 and the independence view, we see that u_f is a well-behaved estimator for E_f (w.r.t. $\mathcal{R}$), with

$$\mathbf{E}[u_f] \leq G(2/\gamma, \gamma - 1). \tag{32}$$

We next consider the case where $\rho_i \geq 1/2$ for all i. Here, even if we condition on any assignment event w.r.t. $\mathcal{R}$, $D_f > 1$ holds only if $\sum_{(i,j):\ f \in P_{i,j}} z_{i,j} \geq 2$. Thus, defining $v_f \doteq \psi_2(\{z_{i,j} :\ f \in P_{i,j}\})$, the independence view can be used to show that v_f is a well-behaved estimator for E_f w.r.t. $\mathcal{R}$. Also, as $\rho_i \geq 1/2$ for all i, (31) implies that $\sum_{(i,j):\ f \in P_{i,j}} \mathbf{E}[z_{i,j}] \leq 2/\gamma$. This, along with part (iii) of Lemma 4.2, shows that

$$\mathbf{E}[v_f] \leq 2/\gamma^2. \tag{33}$$

Our next step is to construct a proper estimator for the bad event that the objective function value is too small. Since the objective function for us is slightly more complicated than that of the PIPs of Section 4.3, we will need a little more work. We start with some useful notation. For each $i \in [k]$, let Z_i be the r.v. that is 1 if at least one of the $z_{i,j}$'s is 1, and let Z_i be 0 if all the $z_{i,j}$'s are 0. For any $T \subseteq [k]$, define $y^*_T \doteq \sum_{i \in T} w_i x^*_i$ and $Z(T) \doteq \sum_{i \in T} w_i Z_i$.

Consider any $T \subseteq [k]$. We now construct a proper estimator p_T for the (bad) event "$Z(T) < y^*_T(1-1/e)/(2\gamma)$" (w.r.t. $\mathcal{R}$). Noting that $Z(T) = \sum_{i \in T} w_i Z_i$, we see that $Z(T)$ is a sum of independent random variables, each lying in $[0,1]$. Since

$$\mathbf{E}[Z(T)] = \sum_{i \in T} w_i \mathbf{E}[Z_i], \tag{34}$$

$\mu_1 \doteq \mathbf{E}[Z(T)]$ is efficiently computable. Let us lower-bound μ_1 first. For each $i \in T$,

$$\Pr[Z_i = 1] = 1 - \prod_j (1 - z^*_{i,j}/\gamma) \geq 1 - e^{-(\sum_j z^*_{i,j})/\gamma} = 1 - e^{-x^*_i/\gamma} \geq x^*_i(1-1/e)/\gamma;$$

the last inequality follows from the fact that for all $y \in [0,1]$, $1 - e^{-y} \geq y(1-1/e)$. Thus, by (34), $\mu_1 \geq \sum_{i \in T} w_i x^*_i (1-1/e)/\gamma = y^*_T(1-1/e)/\gamma$.

In order to construct our proper estimator p_T, we next define $\delta_1 \in (0,1)$ by $\mu_1(1-\delta_1) = y^*_T(1-1/e)/(2\gamma)$. Thus, if we define

$$p_T = \frac{\prod_{i \in T} (1-\delta_1)^{w_i Z_i}}{(1-\delta_1)^{(1-\delta_1)\mu_1}},$$

we see from (10), part (ii) of Lemma 4.2 and the independence view that p_T is indeed a proper estimator as desired. Also, since $\mu_1 \geq y^*_T(1-1/e)/\gamma$, (10) shows that

$$\mathbf{E}[p_T] \leq H(y^*_T(1-1/e)/\gamma, 1/2). \tag{35}$$

As mentioned above, we will treat the "small" and "large" demands separately. We partition $[k]$ into

$$S_0 = \{i \in [k] : \rho_i \leq 1/2\} \text{ and } S_1 = \{i \in [k] : \rho_i > 1/2\}.$$

Theorem 4.5 *For any weighted instance of the* UCUFP*, we can efficiently construct a feasible path-selection with objective function value* $\Omega(\max\{(y^*)^2/m, y^*/\sqrt{m}\})$.

Proof: We first show an $\Omega((y^*)^2/m)$ bound, and then show how it easily implies an $\Omega(y^*/\sqrt{m})$ bound.

Define $S = S_0$ if $y^*_{S_0} \geq y^*_{S_1}$, and $S = S_1$ otherwise; thus, $y^*_S \geq y^*/2$. Theorem 4.3 and the well-behaved and proper estimators that we constructed above, will now help us show how to do a feasible path selection for S with objective function value $\Omega((y^*_S)^2/m) = \Omega((y^*)^2/m)$.

First suppose $S = S_0$. By (32), $\mathbf{E}[u_f] \leq e^2/\gamma^2$ holds for all $f \in E$; also, (35) shows that $\mathbf{E}[p_S] \leq H(y^*_S(1-1/e)/\gamma, 1/2) \leq H(y^*(1-1/e)/(2\gamma), 1/2)$, since $y^*_S \geq y^*/2$. For a suitably large constant c_2, $\gamma = c_2 m/y^*$ satisfies $(1-e^2/\gamma^2)^m > H(y^*(1-1/e)/(2\gamma), 1/2)$; thus we have

$$1 - (\prod_{f \in E} (1 - \mathbf{E'}[u_f])) + \mathbf{E}[p_S] < 1. \tag{36}$$

As seen above, u_f and p_S are well-behaved and proper estimators for the events E_f and "$Z(S) < y^*_S(1-1/e)/(2\gamma)$" respectively. Also, all the events E_f are increasing. Thus, (36) and Theorem 4.3 show that if $S = S_0$, we can efficiently do a feasible path selection for S with objective function value $\Omega((y^*)^2/m)$.

The case $S = S_1$ is similar. Recall that $\mathbf{E}[v_f] \leq 2/\gamma^2$, by (33). The desired analog of (36) now is

$$1 - (\prod_{f \in E} (1 - \mathbf{E'}[s_f])) + \mathbf{E}[p_S] < 1.$$

Thus, it suffices if $(1-2/\gamma^2)^m > H(y^*(1-1/e)/(2\gamma), 1/2)$, which is again ensured by taking $\gamma = c_2 m/y^*$.

We have shown an $\Omega((y^*)^2/m)$ bound; let us now show an $\Omega(y^*/\sqrt{m})$ bound. If $y^* \geq \sqrt{m}$, it immediately follows from the $\Omega((y^*)^2/m)$ bound. If $y^* < \sqrt{m}$, we simply choose a j such that $w_j = 1$ and connect s_j to t_j. (We assume without loss of generality that there is an (s_i, t_i)-path for each i.) ■

Thus we see an application of the framework of Theorem 4.3 to a family of hard low-congestion routing problems.

5 Analyses via the Janson-Łuczak-Ruciński inequality

A major breakthrough concerning random graphs was made in the work of [34]. A key point from there of relevance in our context is the handling of results in a direction *opposite* to that of the FKG inequality. For instance, for certain types of decreasing events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k$, the probability of all the $\mathcal{E}_i$ occurring is *upper*-bounded well in [34] (a reasonable lower bound is immediate via FKG). We refer the reader to [34] and to Chapter 8 of [4] for more details. A result related to these has been given a very simple and elegant proof by Boppana & Spencer [16]; their proof approach is one of the main motivations for some of the work of this section.

The reader is referred to Section 4.4 for notation and results related to low-congestion routing. We borrow all notation from Section 4.4, with the exception that we now let $\{x_i^*, y_{i,f}^* \in [0,1] : i \in [k], f \in E\}$ be any feasible (not necessarily optimal) solution to the LP relaxation, with $y^* = \sum_i w_i x_i^*$ as before. We shall now sketch some further work of [71, 9] on approximating the UCUFP; though the same results have been obtained for the more general UFP in [9], we just handle the UCUFP with all capacities being at most half, for simplicity. Let d denote the maximum length (i.e., number of edges) of the flow-paths $P_{i,j}$ obtained from flow-decomposition as in Section 4.4; this parameter will be crucial for this section. Our main approximation algorithm here, Theorem 5.1, is parametrized by d; some reasons for d being an interesting parameter are sketched following the proof of Theorem 5.1. The work described in this section is essentially from [71, 9]. We briefly sketch some related interesting results of [42] at the end of this section.

We now work toward Theorem 5.1; as mentioned above, one of the key issues will be the handling of inequalities that go in a direction opposite to that of the FKG inequality. One further interesting aspect that will be revealed in the proof of Theorem 5.1 is as follows. In the proof of Theorem 4.3, we could round the underlying variables *in any order*. In the proof of Theorem 5.1, however, we will need to tackle situations where we may need to re-order the underlying variables on the fly.

We start by presenting a useful idea from [16]. Suppose we wish to estimate a probability of the form

$$\Pr[\mathcal{E} \mid (\bigwedge_{r \in [t]} U_r) \wedge (\bigwedge_{r' \in [\ell]} U'_r)], \tag{37}$$

where $\mathcal{E}$ is independent of any Boolean combination of the events $U'_1, U'_2, \ldots, U'_\ell$. Suppose t is "small" or, more generally, suppose the events $U_1, U_2, \ldots, U_t$ do not have "too much influence" on $\mathcal{E}$. We may then expect that by imposing some further (hopefully not very restrictive) conditions, the above probability will be approximately equal to the unconditional probability, $\Pr[\mathcal{E}]$. Indeed, this is one of the key ideas behind the Lovász Local Lemma, in situations where we need to *upper bound* probabilities such as (37). We will see more of this in Section 6.

What if we require a *lower bound* on (37)? An idea from [16] is to proceed as follows. Note the inequality $\Pr[A \mid (B \wedge C)] \geq \Pr[(A \wedge B) \mid C]$. Thus we have

$$\Pr[\mathcal{E} \mid (\bigwedge_{r \in [t]} U_r) \wedge (\bigwedge_{r' \in [\ell]} U'_r)] \;\geq\; \Pr[(\mathcal{E} \wedge (\bigwedge_{r \in [t]} U_r)) \mid \bigwedge_{r' \in [\ell]} U'_r]$$

$$\geq \Pr[\mathcal{E} \mid \bigwedge_{r' \in [\ell]} U'_r] - \sum_{r \in [t]} \Pr[(\mathcal{E} \wedge \overline{U_r}) \mid \bigwedge_{r' \in [\ell]} U'_r]$$
$$= \Pr[\mathcal{E}] - \sum_{r \in [t]} \Pr[(\mathcal{E} \wedge \overline{U_r}) \mid \bigwedge_{r' \in [\ell]} U'_r]. \quad (38)$$

Thus, in particular, if t is "small", then (37) may not be much smaller than $\Pr[\mathcal{E}]$.

Once again, the reader is asked to review the notation of Section 4.4. To place (38) in our context, suppose $\mathcal{E}$ is an event of the form "$z_{i,j} = 1$". Suppose $\mathcal{E}' \equiv ((\bigwedge_{r \in [t]} U_r) \wedge (\bigwedge_{r' \in [\ell]} U'_r))$ is that none of the capacity constraints are violated; i.e., $\mathcal{E}' \equiv (\bigwedge_{f \in E} \overline{E_f})$. Informally, in attempting to show that we can simultaneously satisfy all capacity constraints and still keep the objective function "high", we would like to lower-bound probabilities such as $\Pr[\mathcal{E} \mid \mathcal{E}']$. Unfortunately, by the FKG inequality, "negative correlation" implies an *upper* bound: $\Pr[\mathcal{E} \mid \mathcal{E}'] \leq \Pr[\mathcal{E}]$.

Let us see why bounds such as (38) help in such situations. Recall that $z_{i,j}$ is the indicator variable for path $P_{i,j}$ being chosen, and that $P_{i,j}$ has at most d edges by definition of d. Think of d as "small". Now, $\mathcal{E}'$ is a conjunction of m events, one each for the capacity constraint of each edge $f \in E$. Since all the flow-paths are rounded independently, our event $\mathcal{E}$ is independent of all but at most d of the events that constitute $\mathcal{E}'$. As seen above, this is a type of situation where (38) could be a good lower-bound on (37). The proof of Lemma 5.1 expands on this intuition in our UCUFP setting.

We return to the UCUFP. Recall that by scaling, we can assume that all capacities are 1. Also, as mentioned above, we only work with the case where all demands ρ_i are at most a half, here; the reader is referred to [71, 9] for the general case. Our main result now will be Theorem 5.1, which shows how to do a feasible path selection with objective function value $\Omega(y^*/d)$.

Define, for all (f, i, j) such that $f \in P_{i,j}$,

$$D_f(i,j) \doteq D_f - \rho_i z_{i,j} = \sum_{(r,s) \neq (i,j):\ f \in P_{r,s}} \rho_r z_{r,s}. \quad (39)$$

Recall that k denotes the number of connection requests (s_i, t_i). Also, as before, for each $i \in [k]$, Z_i is the indicator r.v. for the event "$(\exists j :\ z_{i,j} = 1)$"; $Z([k])$ is the random variable denoting the objective function value.

Lemma 5.1 *For any set X of pairs (i,j) and any set of $\{0,1\}$ values $\{b_{i,j} \in \{0,1\} : (i,j) \in X\}$, let $\mathcal{A}$ denote the assignment event that "$\forall (i,j) \in X,\ z_{i,j} = b_{i,j}$". Suppose $\Pr[\bigwedge_{f \in E} \overline{E_f} \mid \mathcal{A}] > 0$. Then:*

(a) For any i, $\mathbf{E}[Z_i \mid (\mathcal{A} \wedge (\bigwedge_{f \in E} \overline{E_f}))]$ is at least

$$(\sum_j \Pr[z_{i,j} = 1 \mid \mathcal{A}](1 - \sum_{f \in P_{i,j}} \Pr[D_f(i,j) > (1 - \rho_i) \mid \mathcal{A}])) - \sum_{j < j'} \Pr[(z_{i,j} = z_{i,j'} = 1) \mid \mathcal{A}].$$

(b) In particular, suppose $\gamma \geq 2$ and that $\forall i\ \forall j\ \forall f \in P_{i,j},\ \Pr[D_f(i,j) > (1 - \rho_i)] \leq 1/(2d)$. Then,

$$\Pr[\bigwedge_{f \in E} \overline{E_f} \mid \mathcal{A}] > 0 \text{ and } \mathbf{E}[Z([k]) \mid \bigwedge_{f \in E} \overline{E_f}] \geq y^*/(4\gamma).$$

Proof: (a) Fix i. Define, for each j,

$$g(j) \doteq \Pr[z_{i,j} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}))] - \sum_{f \in P_{i,j}} \Pr[(z_{i,j} = 1) \wedge (D_f > 1) \mid (\mathcal{A} \wedge (\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}))]. \quad (40)$$

As before, even conditional on $\mathcal{A}$, the "independence view" lets us take all the $z_{i,j}$'s as *independent*. Now, $(Z_i = 1) \equiv (\exists j : \ z_{i,j} = 1)$. We have

$$\begin{aligned} \mathbf{E}[Z_i \mid (\mathcal{A} \wedge (\bigwedge_{f \in E} \overline{E_f}))] &\geq \sum_j \Pr[z_{i,j} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}))] - \\ &\quad \sum_{j<j'} \Pr[z_{i,j} = z_{i,j'} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \in E} \overline{E_{f'}}))] \\ &\geq (\sum_j g(j)) - \sum_{j<j'} \Pr[z_{i,j} = z_{i,j'} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \in E} \overline{E_{f'}}))]. \quad (41) \end{aligned}$$

The first bound is a consequence of inclusion-exclusion; the reader is invited to check that (41) follows from the independence view and the discussion above about (38) and its applications.

We now lower-bound the expression in (41). Fix j. By the independence view, we can check that

$$\Pr[z_{i,j} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}))] = \Pr[z_{i,j} = 1 \mid \mathcal{A}]. \quad (42)$$

All remaining applications of the FKG inequality will be favorable to us. Consider any $f \in P_{i,j}$. By the independence view, even conditional on $\mathcal{A}$: (i) all the underlying random variables are still independent, and (ii) the event "$(z_{i,j} = 1) \wedge (D_f > 1)$" is increasing as a function of the underlying random variables, while "$\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}$" is decreasing. Thus, by the FKG inequality,

$$\begin{aligned} \Pr[(z_{i,j} = 1) \wedge (D_f > 1) \mid (\mathcal{A} \wedge (\bigwedge_{f' \notin P_{i,j}} \overline{E_{f'}}))] &\leq \Pr[(z_{i,j} = 1) \wedge (D_f > 1) \mid \mathcal{A}] \\ &= \Pr[(z_{i,j} = 1) \wedge D_f(i,j) > 1 - \rho_i \mid \mathcal{A}] \quad (43) \end{aligned}$$

Since $D_f(i,j)$ involves only variables other than $z_{i,j}$, the independence view yields

$$\Pr[(z_{i,j} = 1) \wedge (D_f(i,j) > (1 - \rho_i)) \mid \mathcal{A}] = \Pr[z_{i,j} = 1 \mid \mathcal{A}] \cdot \Pr[D_f(i,j) > (1 - \rho_i) \mid \mathcal{A}]. \quad (44)$$

Next, as above, the independence view and the FKG inequality combine to show, for $j \neq j'$, that

$$\Pr[z_{i,j} = z_{i,j'} = 1 \mid (\mathcal{A} \wedge (\bigwedge_{f' \in E} \overline{E_{f'}}))] \leq \Pr[z_{i,j} = z_{i,j'} = 1 \mid \mathcal{A}]. \quad (45)$$

Substituting (40), (42), (43), (44) and (45) into (41) completes the proof of part (a).

(b) First, since $\gamma > 1$, there is a nonzero probability that all the $z_{i,j}$ are 0. So $\Pr[\bigwedge_{f \in E} \overline{E_f}] > 0$. Recall next that $\forall i, \ \sum_j \Pr[z_{i,j} = 1] = x_i^*/\gamma$. Thus, by part (iii) of Lemma 4.2,

$$\sum_{j<j'} \Pr[z_{i,j} = z_{i,j'} = 1] \leq (x_i^*/\gamma)^2/2 \leq x_i^*/(4\gamma),$$

the last inequality following since $x_i^* \in [0,1]$ and $\gamma \geq 2$. Also, since each $P_{i,j}$ has at most d edges, we see, by setting $X = \emptyset$ and $\mathcal{A}$ to be the tautology in part (a), that for all i,

$$\mathbf{E}[Z_i \mid \bigwedge_{f \in E} \overline{E_f}] \geq (\sum_j \Pr[z_{i,j} = 1](1 - d/(2d))) - x_i^*/(4\gamma) = x_i^*/(2\gamma) - x_i^*/(4\gamma) = x_i^*/(4\gamma).$$

Thus, $\mathbf{E}[Z([k]) \mid \bigwedge_{f \in E} \overline{E_f}] \geq (\sum_i w_i x_i^*)/(4\gamma)$. ■

Recall our approximation algorithm of Section 4.4, which was motivated in part by the methods of Section 4.3. One key point about this approach is that the objective function is treated separately from the constraints, by applying a union bound. However, we now have Lemma 5.1, which shows lower bounds on the expected objective function value *even conditional on*, e.g., the event that all capacity constraints are obeyed. This will let us handle the capacity constraints and objective function together, which is a key driver of our approximation guarantee that is parametrized by d. We demonstrate this through our next main theorem:

Theorem 5.1 ([71, 9]) *Suppose all the demands are at most* $1/2$ *and all capacities are* 1 *in a given* UCUFP *instance. We can round the* $z_{i,j}$*'s in deterministic polynomial time to produce a feasible path selection that has objective function value* $\Omega(y^*/d)$.

Proof: Choose $\gamma = 4d$. For any f, i, j,

$$\begin{aligned}
\Pr[D_f(i,j) > (1 - \rho_i)] &\leq \Pr[D_f(i,j) > 1/2] \\
&\leq 2 \cdot \sum_{(r,s) \neq (i,j):\ f \in P_{r,s}} \rho_r \Pr[z_{r,s} = 1] \text{ (Markov's inequality)} \quad (46) \\
&\leq 2 \cdot \mathbf{E}[D_f] \leq 2/\gamma = 1/(2d). \quad (47)
\end{aligned}$$

Thus, by part (b) of Lemma 5.1, there exists a rounding that does a feasible path selection with objective function value $\Omega(y^*/d)$. We now show how to turn this into an efficient algorithm, by rounding the $z_{i,j}$'s one-by-one.

Assume inductively, for some set X of pairs (i,j) and some set $Y = \{b_{i,j} \in \{0,1\} : (i,j) \in X\}$, that we have already rounded $z_{i,j}$ to $b_{i,j}$, $\forall (i,j) \in X$. Let $\mathcal{A}$ denote the event that "$\forall (i,j) \in X,\ z_{i,j} = b_{i,j}$". ($X = Y = \emptyset$ when we start our algorithm.) As in the proof of Lemma 5.1, even conditional on $\mathcal{A}$, we can take the independence view that all the $z_{i,j}$'s are independent. Suppose we define a potential function $\Phi(\mathcal{A})$ to be

$$\sum_i w_i ((\sum_j \Pr[z_{i,j} = 1](1 - 2 \sum_{f \in P_{i,j}} \sum_{(r,s) \neq (i,j):\ f \in P_{r,s}} \rho_r \Pr[z_{r,s} = 1])) - \sum_{j < j'} \Pr[z_{i,j} = 1] \cdot \Pr[z_{i,j'} = 1]), \quad (48)$$

with the convention that "$\forall (i,j) \in X,\ \Pr[z_{i,j} = b_{i,j}] = 1$". Note that (46) holds even when the probabilities in its l.h.s. and r.h.s. are computed conditional on $\mathcal{A}$. Thus, Lemma 5.1(a) and our independence view show that

$$\mathbf{E}[Z([k]) \mid (\mathcal{A} \wedge (\bigwedge_{f \in E} \overline{E_f}))] \geq \Phi(\mathcal{A}). \quad (49)$$

Our inductive requirements on X and $\mathcal{A}$ are: **(Q1)** $\Pr[\bigwedge_{f\in E}\overline{E_f} \mid \mathcal{A}] > 0$, and **(Q2)** $\Phi(\mathcal{A}) \geq y^*/(4\gamma)$. By (46), (47) and part (b) of Lemma 5.1, (Q1) and (Q2) are satisfied initially, when $X = \emptyset$. Given that some $\mathcal{A}$ satisfies (Q1) and (Q2), we now show how to add one or more new elements to X and update $\mathcal{A}$ appropriately, to maintain (Q1) and (Q2). If we have rounded all the $z_{i,j}$'s and still maintain (Q1) and (Q2), we see by (49) that we have made a feasible path selection of value at least $y^*/(4\gamma)$.

Let $(i,j) \notin X$ be arbitrary. We wish to decide which way to round $z_{i,j}$. Define $\mathcal{A}_0$ to be "$\mathcal{A} \wedge (z_{i,j} = 0)$", and $\mathcal{A}_1$ to be "$\mathcal{A} \wedge (z_{i,j} = 1)$". For each $f \in E$, define

$$cd_{\mathcal{A}}(f) = \sum_{(r,s)\in X:\ f\in P_{r,s}} \rho_r b_{r,s};$$

since we have rounded $z_{r,s}$ to $b_{r,s}$ for each $(r,s) \in X$, this is the demand that we have so far committed on f. Now, if there is an edge f belonging to $P_{i,j}$ such that $cd_{\mathcal{A}}(f) > 1 - \rho_i$, then rounding $z_{i,j}$ to 1 will violate the capacity constraint on f:

$$\Pr[\bigwedge_{f\in E} \overline{E_f} \mid \mathcal{A}_1] = 0. \tag{50}$$

Otherwise, suppose that $\forall f \in P_{i,j},\ cd_{\mathcal{A}}(f) \leq 1 - \rho_i$. Since $\gamma < 1$, there is a nonzero probability that all the yet-unrounded variables get rounded to 0; so,

$$\Pr[\bigwedge_{f\in E} \overline{E_f} \mid \mathcal{A}_0] > 0, \text{ and } \Pr[\bigwedge_{f\in E} \overline{E_f} \mid \mathcal{A}_1] > 0. \tag{51}$$

Our algorithm is thus as follows. There are two possibilities:

(C1): $\exists (i,j) \notin X :\ \forall f \in P_{i,j},\ cd_{\mathcal{A}}(f) \leq 1 - \rho_i$, and
(C2): $\forall (i,j) \notin X :\ \exists f \in P_{i,j},\ cd_{\mathcal{A}}(f) > 1 - \rho_i$.

Suppose (C1) holds. We have

$$\Phi(\mathcal{A}) = \mathbf{E}[(\sum_u w_u \sum_v z_{u,v}(1 - 2 \sum_{f\in P_{u,v}} \sum_{(r,s)\neq(u,v):\ f\in P_{r,s}} \rho_r z_{r,s})) - \sum_u w_u \sum_{v<v'} z_{u,v} \cdot z_{u,v'}] \tag{52}$$

by the independence view. Thus, $\Phi(\mathcal{A})$ is a convex combination of $\Phi(\mathcal{A}_0)$ and $\Phi(\mathcal{A}_1)$: so there exists a $b \in \{0,1\}$ such that $\Phi(\mathcal{A}_b) \geq \Phi(\mathcal{A})$. Furthermore, $\Pr[\bigwedge_{f\in E}\overline{E_f} \mid \mathcal{A}_b] > 0$, from (51). Importantly, b can be computed efficiently since the potential function is efficiently computable. Thus, adding (i,j) to X and rounding $z_{i,j}$ to b (i.e., setting $\mathcal{A} := \mathcal{A}_b$), will maintain (Q1) and (Q2).

On the other hand, suppose (C2) is true. Then, by (50), rounding any yet-unrounded (i,j) to 1 will make E_f true for some $f \in E$. But, we are assured by (49), (Q1) and (Q2) that there exists a rounding of the yet-unrounded variables, such that $\overline{E_f}$ holds for all $f \in E$ and such that $Z([k]) \geq y^*/(4\gamma)$. Thus, the only possible valid rounding for all remaining variables is to set them to 0, and this is guaranteed to be a feasible path selection ($\overline{E_f}$ holds for all $f \in E$) with objective function value at least $y^*/(4\gamma)$.

Thus, in both cases (C1) and (C2), we can successfully keep rounding the variables. So we have a deterministic polynomial-time algorithm that outputs a feasible path selection of value $\Omega(y^*/d)$. ■

As mentioned before, one other way in which the above approach differs from the previous such approaches, is that the underlying variables cannot be rounded in any arbitrary pre-imposed order, but may have to be re-ordered dynamically.

We now sketch some applications of Theorem 5.1. Recall a classical graph invariant, the (edge-)expansion. Given an undirected graph $G = (V, E)$ and any $S \subseteq V$, let $\delta(S)$ denote the set of edges of G that have precisely one end-point in S. The edge-expansion of G, $\beta = \beta(G)$, is defined to be

$$\min_{S \subseteq V : |S| \leq n/2} |\delta(S)|/|S|.$$

Expansion is well-known to be of much importance in routing. For instance, graphs with a "high" value of β have provably good routability and fault-tolerance properties; the reader is referred to [45]. Let $\Delta = \Delta(G)$ denote the maximum degree of the vertices of G.

Given an instance $(G, \mathcal{T})$ of the UCUFP, let $\alpha^*(\mathcal{T})$ denote the optimum value of the LP relaxation studied above. Then, a nice result of [39] shows the following: for any constant $\epsilon > 0$, there exists a feasible solution to the LP relaxation with:

- objective function value at least $\alpha^*(\mathcal{T})/(1+\epsilon)$, and such that
- all flow-paths have length at most $d_0 = \mathrm{O}(\Delta^2 \beta^{-2} \log^3 n)$.

We can easily add the *linear* constraints that "all flow paths are of length at most d_0" to our LP relaxation; thus, by Theorem 5.1, we have an $\mathrm{O}(\Delta^2 \beta^{-2} \log^3 n)$-approximation algorithm for the UCUFP. The case of "small" (say, constant) Δ and "reasonably large" (say, $\Omega(1/\mathrm{polylog}(n))$) β is a typical property of well-known interconnection networks such as the butterfly. Thus, for such graphs in particular, we get good approximations for the UCUFP. Furthermore, the flow-paths are sometimes *required* to be "short"; in such cases, if d is bounded by a constant, Theorem 5.1 shows a constant-factor approximation algorithm for the UCUFP.

In recent interesting work, Kolliopoulos & Stein [42] have matched the result of Theorem 5.1 and nearly (i.e., to within a small polylogarithmic factor) matched Theorem 4.5 for weighted instances of the UFP. In work done independently of [42], the bounds of Theorems 4.5 and 5.1 have been generalized in [9] to weighted instances of the UFP.

Also, a class of integer programs (*column-restricted packing integer programs*) related to PIPs and to the UFP have been proposed and analyzed in [42]. See [42, 9] for approximation algorithms for this class, and applications to certain families of packing integer programs.

Finally, we mention that in recent work on approximating the group Steiner tree problem [25], a "correlations opposite to FKG" situation has been handled via an inequality related to [34].

6 Applications of an extension of the Lovász Local Lemma

The main theme of most of Sections 4 and 5 has been the use of appropriate *correlation inequalities* to present improved analyses of LP-based randomized rounding in various contexts. We continue this thread here. We present an extension of the Lovász Local Lemma and some of its applications, following [70]. Also, the parametrization of the UCUFP by d in Section 5 can basically be seen as a *column sparsity* condition on the corresponding ILP: all columns have at most d nonzero entries. We also follow Section 5 in this sense: the primary results of this section will be improved analyses of LP-based randomized rounding for a family of column-sparse ILPs.

To motivate the combinatorial optimization problems considered, let us return to the MDP. In the MDP, we attempt to route a maximum-weight sub-collection of the given (multi-)set of requests $\mathcal{T} = \{(s_i, t_i) : i \in [k]\}$, subject to the capacity constraints. But what if we indeed need to route *all* the pairs (s_i, t_i)? One natural question here is how to do the routing to minimize *congestion*–the maximum number of chosen paths that use any given edge. This NP-hard problem has also been studied earlier in the context of global routing in VLSI gate arrays [58]. This problem is a member of a class of integer programs that we introduce in Section 6.1 using the name *minimax integer programs*. Section 6.1 presents the union-bound based analysis of LP-based randomized rounding for MIPs, due to [58]. We then present the extended LLL in Section 6.2 and sketch how it can be employed to analyze the same randomized rounding scheme better.

6.1 Minimax integer programs and union-bound based analysis

We start by defining the minimax integer programs (MIPs) as in [70].

Definition 6.1 *An MIP has variables W and $\{x_{i,j} : i \in [n], j \in [\ell_i]\}$, for some integers $\{\ell_i\}$. Let $N = \sum_{i \in [n]} \ell_i$ and let x denote the N-dimensional vector of the variables $x_{i,j}$ (arranged in any fixed order). An MIP seeks to minimize a real W, subject to:*

(i) Equality constraints: $\forall i \in [n]$ $\sum_{j \in [\ell_i]} x_{i,j} = 1$;

(ii) a system of linear inequalities $Ax \leq \vec{W}$, where $A \in [0,1]^{m \times N}$ and $\vec{W}$ is the m-dimensional vector with the variable W in each component, and

(iii) Integrality constraints: $x_{i,j} \in \{0,1\}$ $\forall i, j$.

To see what problems MIPs model, note, from constraints (i) and (iii), that for all i, any feasible solution will make the set $\{x_{i,j} : j \in [\ell_i]\}$ have precisely one 1, with all other elements being 0. Thus, MIPs model many "choice" scenarios. For instance, consider the "congestion minimization" version of the edge-disjoint paths problem. We have an undirected graph G and a (multi-)set of vertex-pairs $\mathcal{T} = \{(s_i, t_i) : i \in [k]\}$. We wish to find one (s_i, t_i)-path in G for each i, such that the maximum congestion on any edge f in G (the number of paths

using f) is minimized. Consider the "natural" LP relaxation of this problem, which is closely related to the one studied for the MDP and UCUFP: we want to ship one unit of flow from s_i to t_i (using potentially several paths), such that the maximum flow on any edge is minimized. Start with an optimal solution to this LP relaxation and do a flow-decomposition as done for the UCUFP. This leads to a set of flow-paths $\{P_{i,j}\}$ for each (s_i, t_i), and the problem of picking exactly one of them for each i in order to minimize congestion, is readily seen to be an instance of MIPs. (Variable $x_{i,j}$ is the indicator for choosing path $P_{i,j}$, and there is one constraint for each edge.) An application of MIPs to a hypergraph-partitioning problem will be considered in Section 6.2.

A natural LP relaxation for a given MIP is to allow the entries of x to be non-negative reals subject to the family of constraints (i) and (ii). Let x^* and y^* denote, resp., an optimal solution to, and the optimum value of, this LP relaxation. Thus, letting OPT denote the optimum objective function value for a given MIP, we have $y^* \leq OPT$.

A natural randomized rounding approach, keeping in mind the set (i) of constraints of MIPs, is as follows: independently for each i, randomly round *exactly one* $x_{i,j}$ to 1, guided by the "probabilities" $\{x^*_{i,j}\}$. We will now present an analysis of this scheme due to [58]. An improved analysis will be shown in Section 6.2.

Recall the CH bound function $G(\mu, \delta)$ of (9). We start with a simple lemma to quantify the approximation results that we shall present for MIPs.

Lemma 6.1 *a.* $\forall \mu > 0\ \forall p \in (0,1)\ \exists \delta = D(\mu, p) > 0$ *such that: (i)* $G(\mu, \delta) \leq p$, *and (ii)* $D(\mu, p) = \Theta(\frac{\log(p^{-1})}{\mu \log(\log(p^{-1})/\mu)})$ *if* $\mu \leq \log(p^{-1})/2$, *and is* $\Theta(\sqrt{\frac{\log(p^{-1})}{\mu}})$ *otherwise.*

b. $\forall \mu > 0\ \forall p \in (0,1),\ \exists \delta = L(\mu, p) > 0$ *such that: (i)* $\lceil \mu\delta \rceil \cdot G(\mu, \delta) \leq p$, *and (ii)* $L(\mu, p) = \Theta(\frac{\log(p^{-1})}{\mu \log(\log(p^{-1})/\mu)})$ *if* $\mu \leq \log(p^{-1})/2$, *and is* $\Theta(\sqrt{\frac{\log(\mu + p^{-1})}{\mu}})$ *otherwise.*

c. If $0 < \mu_1 \leq \mu_2$, *then for any* $\delta > 0$, $G(\mu_1, \mu_2\delta/\mu_1) \leq G(\mu_2, \delta)$.

Proof: Parts (a) and (b) can be derived using Remark 2.1. For part (c), we need to show that

$$(1+\delta)^{(1+\delta)\mu_2} \leq \left(1 + \frac{\mu_2\delta}{\mu_1}\right)^{(1+\frac{\mu_2\delta}{\mu_1})\mu_1},$$

i.e., that $h(v) = (1+v\delta)\ln(1+v\delta) - v(1+\delta)\ln(1+\delta) \geq 0$ for all $v \geq 1$. Note that $h(1) = 0$ and that $h'(v) = \delta + \delta\ln(1+v\delta) - (1+\delta)\ln(1+\delta)$. For $v \geq 1$, $h'(v) \geq \delta - \ln(1+\delta) \geq 0$, completing the proof. ■

Theorem 6.1 ([58]) *Given an MIP conforming to Definition 6.1, randomized rounding produces a feasible solution of value at most* $y^* + min\{y^*, m\} \cdot D(min\{y^*, m\}, 1/(2m))$, *with probability at least* $1/2$.

Proof: We may assume that $\{x^*_{i,j}\}$ is a *basic* feasible solution to the LP relaxation. Hence, at most m of the $\{x^*_{i,j}\}$ will be neither zero nor one, and only these variables will participate

in the rounding. Thus, since all the entries of A are in $[0,1]$, we assume w.l.o.g. henceforth that $y^* \leq m$ (and that $\max_{i\in[n]} \ell_i \leq m$); this explains the $\min\{y^*, m\}$ term in the statement of Theorem 6.1.

Conduct randomized rounding. If $z \in \{0,1\}^N$ denotes the randomly rounded vector, then $\mathbf{E}[(Az)_i] \leq y^*$ for each i, by linearity of expectation. Let $b_i = \mathbf{E}[(Az)_i]$. Define $k = y^* D(\min\{y^*, m\}, 1/(2m))$ and define events $E_1, E_2, \ldots, E_m$ by $E_i \equiv$ "$(Az)_i \geq b_i + k$". We now show that for each i, $\Pr[E_i] \leq 1/(2m)$; as in our approach for lattice approximation, a union bound over all i will then complete the proof.

For each $i \in [m]$, rewrite the ith constraint of the MIP as

$$\sum_{r\in[n]} X_{i,r} \leq W, \text{ where } X_{i,r} = \sum_{s\in[\ell_r]} A_{i,(r,s)} x_{r,s}; \tag{53}$$

in using the notation $A_{i,(r,s)}$, we are assuming that the pairs $\{(r,s) : r \in [n], s \in [\ell_r]\}$ have been mapped bijectively to $[N]$, in some fixed way. Defining the r.v.

$$Z_{i,r} = \sum_{s\in[\ell_r]} A_{i,(r,s)} z_{r,s}, \tag{54}$$

we note that for each i, the r.v.s $\{Z_{i,r} : r \in [n]\}$ lie in $[0,1]$ and are *independent*. Also, $E_i \equiv$ "$\sum_{r\in[n]} Z_{i,r} \geq b_i + k$". For any $i \in [m]$, let $\delta_i = k/b_i$. Since $b_i \leq y^*$, we have, for each $i \in [m]$,

$$\Pr[E_i] \leq G(b_i, \delta_i) \leq G(y^*, k/y^*) = G(y^*, D(y^*, 1/(2m))) \leq 1/(2m),$$

where the second inequality follows from part (c) of Lemma 6.1 and the final inequality holds by the definition of D. As mentioned above, a union bound over all $i \in [m]$ now completes the proof. ■

So suppose, e.g., that for a given congestion minimization problem, $y^* \leq 1$. Thus, the congestion produced is $O(\log m / \log\log m)$ with high probability.

There seem to be some avenues for improvement in the above proof; for instance, if we just want $\Pr[\bigwedge_{i\in[m]} \overline{E_i}] > 0$, can we exploit the particular structure of MIPs in conjunction with some correlation inequalities to show that a smaller value of k will suffice? Recall from Section 5 that UCUFP instances where all flow-paths $P_{i,j}$ are short can be approximated well. Now, in the above-seen modeling of the congestion minimization problem by an MIP, the length of a longest $P_{i,j}$ will correspond to the largest number of nonzero entries in any column of A. Motivated by this, we can ask: if the coefficient matrix of a given MIP is *column-sparse*, can it be approximated well? To make this more concrete, let us define

Definition 6.2 *For a given MIP, let a denote the maximum number of non-zero entries in any column of its coefficient matrix A. Also define $g \leq a$ to be the maximum column-sum of A, i.e.,* $\max_j \sum_i A_{i,j}$.

So, a natural question is to derive approximations (or at least integrality gaps) for MIPs parametrized by a or g. One such example is Theorem 7.2, which uses a key rounding

theorem of [36] to efficiently construct an integral solution of value at most $y^* + g$. While very useful if g is "small", this may not be so useful if g is somewhat large compared to y^*. Theorem 6.3 shows how Theorem 6.2 can help, for sparse MIPs. We will then bootstrap Theorem 6.3 to get the further improved Theorem 6.4.

6.2 The extended LLL and some of its applications

The work described in Section 6.2 is from [70]. We start with an extension of the LLL, Theorem 6.2. The main application of this extension here will be to Theorem 6.4.

Recall that for any event $\mathcal{E}$, its indicator r.v. is denoted by $\chi(\mathcal{E})$. Suppose we have bad events $E_1, \ldots, E_m$ with a "dependency" d' (in the sense of Lemma 3.2) that is "large". Theorem 6.2 shows some conditions under which d' can effectively be replaced by a potentially much smaller d. It generalizes Lemma 3.2 (define one r.v., $C_{i,1} = \chi(E_i)$, for each i, to get Lemma 3.2), and its proof is very similar to the classical proof of Lemma 3.2. The main motivation for Theorem 6.2 is that it leads to an improved analysis of randomized rounding for column-sparse MIPs via Theorems 6.3 and 6.4.

Theorem 6.2 *Given events $E_1, \ldots, E_m$ and any $I \subseteq [m]$, let $Z(I) \doteq \bigwedge_{i \in I} \overline{E_i}$. Suppose that for some positive integer d, we can define, for each $i \in [m]$, a finite number of r.v.s $C_{i,1}, C_{i,2}, \ldots$ each taking on only* non-negative *values such that:*

(i) any $C_{i,j}$ is mutually independent of all but at most d of the events E_k, $k \neq i$, and

(ii) $\forall I \subseteq ([m] - \{i\})$, $\Pr[E_i \mid Z(I)] \leq \sum_j \mathbf{E}[C_{i,j} \mid Z(I)]$.

Let p_i denote $\sum_j \mathbf{E}[C_{i,j}]$; clearly, $\Pr[E_i] \leq p_i$ (set $I = \emptyset$ in (ii)). Suppose that for all $i \in [m]$ we have $ep_i(d+1) \leq 1$. Then $\Pr[\bigwedge_i \overline{E_i}] \geq (d/(d+1))^m > 0$.

Proof: We prove by induction on $|I|$ that if $i \notin I$ then $\Pr[E_i \mid Z(I)] \leq ep_i$, which suffices to prove the theorem since

$$\Pr[\bigwedge_i \overline{E_i}] = \prod_{i \in [m]} (1 - \Pr[E_i \mid Z([i-1])]).$$

For the base case $I = \emptyset$, $\Pr[E_i \mid Z(I)] = \Pr[E_i] \leq p_i$. For the inductive step, let $S_{i,j,I} \doteq \{k \in I \mid C_{i,j} \text{ depends on } E_k\}$, and $S'_{i,j,I} = I - S_{i,j,I}$; note that $|S_{i,j,I}| \leq d$. If $S_{i,j,I} = \emptyset$, then $\mathbf{E}[C_{i,j} \mid Z(I)] = \mathbf{E}[C_{i,j}]$. Otherwise, letting $S_{i,j,I} = \{\ell_1, \ldots, \ell_r\}$,

$$\mathbf{E}[C_{i,j} \mid Z(I)] = \frac{\mathbf{E}[C_{i,j} \cdot \chi(Z(S_{i,j,I})) \mid Z(S'_{i,j,I})]}{\Pr[Z(S_{i,j,I}) \mid Z(S'_{i,j,I})]} \leq \frac{\mathbf{E}[C_{i,j} \mid Z(S'_{i,j,I})]}{\Pr[Z(S_{i,j,I}) \mid Z(S'_{i,j,I})]}, \tag{55}$$

since $C_{i,j}$ is non-negative. The numerator of the last term is $\mathbf{E}[C_{i,j}]$ by the definition of $S'_{i,j,I}$. The denominator is

$$\prod_{s \in [r]} (1 - \Pr[E_{\ell_s} \mid Z(\{\ell_1, \ell_2, \ldots, \ell_{s-1}\} \cup S'_{i,j,I})]) \quad \geq \quad \prod_{s \in [r]} (1 - ep_{\ell_s}) \text{ (induction hypothesis)}$$

$$\begin{aligned} &\geq (1-1/(d+1))^r \\ &\geq (d/(d+1))^d \\ &> 1/e. \end{aligned}$$

So by (55), $\mathbf{E}[C_{i,j} \mid Z(I)] \leq e\mathbf{E}[C_{i,j}]$ and hence,

$$\Pr[E_i \mid Z(I)] \leq \sum_j \mathbf{E}[C_{i,j} \mid Z(I)] \leq ep_i \leq 1/(d+1).$$

This establishes the inductive step. ■

Note that $C_{i,j}$ and $C_{i,j'}$ can "depend" on *different* subsets of $\{E_k | k \neq i\}$; the only restriction is that these subsets be of size at most d. Moreover, the dependency *among the r.v.s* $C_{i,j}$ could be much higher than d: all we upper-bound is the number of E_k that any $C_{i,j}$ depends on.

Thus, the crucial point is that the events E_i could have a large dependency d' among themselves, in the sense of the classical Lemma 3.2; but Theorem 6.2 essentially reduces this dependency to just d ($ep_i(d+1) \leq 1$ suffices). So, the main utility of Theorem 6.2 is in situations where we can "decompose" each E_i into r.v.s $C_{i,j}$ that satisfy the conditions of the theorem. This is indeed what we now set about to show for MIPs.

The tools behind the MIP application are Theorems 6.2 and 2.1. Suppose we are given an MIP conforming to Definition 6.1. Let $t \doteq \max_{i\in[n]} NZ_i$, where NZ_i is the number of rows of A which have a non-zero coefficient corresponding to at least one variable among $\{x_{i,j} : j \in [\ell_i]\}$. Note from Definition 6.2 that

$$g \leq a \leq t \leq \min\{m, a \cdot \max_{i\in[n]} \ell_i\}. \tag{56}$$

Our first approximation for MIPs is given by

Theorem 6.3 *Given an MIP conforming to Definition 6.1, randomized rounding produces a feasible solution of value at most* $y^* + min\{y^*, m\} \cdot L(min\{y^*, m\}, 1/(et))$, *with non-zero probability.*

Proof: Conduct randomized rounding. As in the proof of Theorem 6.1, we

- can assume w.l.o.g. that $y^* \leq m$ and that $\max_{i\in[n]}\ell_i \leq m$;
- let $z \in \{0,1\}^N$ denote the randomly rounded vector;
- have for each $i \in [m]$ that $b_i \doteq \mathbf{E}[(Az)_i] \leq y^*$; and
- rewrite each of the constraints as in (53) and define the r.v.s $Z_{i,r}$ as in (54).

As before, for each i the r.v.s $\{Z_{i,r} : r \in [n]\}$ lie in $[0,1]$ and are independent. Define $k = \lceil y^* \cdot L(y^*, 1/(et)) \rceil$. The bad events $E_1, E_2, \ldots, E_m$ now are given by $E_i \equiv ((Az)_i \geq b_i + k)$. Hence, $E_i \equiv (\sum_{r \in [n]} Z_{i,r} \geq b_i + k)$. We now show that $\Pr[\bigwedge_{i \in [m]} \overline{E_i}] > 0$ using Theorem 6.2.

Theorem 2.1 suggests a suitable choice for the crucial r.v.s $C_{i,j}$ in order to apply Theorem 6.2. Let $u = \binom{n}{k}$; we now define the r.v.s $\{C_{i,j} : i \in [m], j \in [u]\}$ as follows. Fix any $i \in [m]$. Identify each $j \in [u]$ with some distinct k-element subset $S(j)$ of $[n]$, and let

$$C_{i,j} \doteq \frac{\prod_{v \in S(j)} Z_{i,v}}{\binom{b_i + k}{k}}.$$

(Note that some of the "random variables" $Z_{i,v}$ are actually identically zero. If, for some given (i,j), there is such a $Z_{i,v}$ for some $v \in S(j)$, then of course $C_{i,j}$ is also identically zero.)

We will now show that the r.v.s $C_{i,j}$ satisfy the conditions of Theorem 6.2. For any $i \in [m]$, let $\delta_i = k/b_i$. Since $b_i \leq y^*$, we have, for each $i \in [m]$,

$$\begin{aligned} G(b_i, \delta_i) &\leq G(y^*, k/y^*) \text{ (Lemma 6.1, part (c))} \\ &\leq G(y^*, L(y^*, 1/(et))) \text{ (definition of } k) \\ &\leq 1/(ekt) \text{ (definition of } L). \end{aligned}$$

Now by Theorem 2.1, we get

Proposition 6.1 *For all $i \in [m]$ and for all nonempty events Z, $\Pr[E_i \mid Z] \leq \sum_{j \in [u]} \mathbf{E}[C_{i,j} \mid Z]$. Also, $p_i \doteq \sum_{j \in [u]} \mathbf{E}[C_{i,j}] \leq G(b_i, \delta_i) \leq 1/(ekt)$.*

Crucially, every $C_{i,j}$ involves (a product of) k terms, each of which "depends" on at most $(t-1)$ of the events $\{E_v : v \in ([m] - \{i\})\}$ by definition of t. So we get

Proposition 6.2 *$\forall i \in [m]$ $\forall j \in [u]$, $C_{i,j} \in [0,1]$ and $C_{i,j}$ "depends" on at most $d = k(t-1)$ of the set of events $\{E_v : v \in ([m] - \{i\})\}$.*

From Propositions 6.1 and 6.2 and by noting that $ep_i(d+1) \leq e(kt - k + 1)/(ekt) \leq 1$, we invoke Theorem 6.2 to see that $\Pr[\bigwedge_{i \in [m]} \overline{E_i}] > 0$. ∎

As shown in [70], an idea suggested by Éva Tardos can help bootstrap Theorem 6.3 to give

Theorem 6.4 ([70]) *For MIPs, there is an integral solution of value at most $y^* + \mathrm{O}(min\{y^*, m\} \cdot L(min\{y^*, m\}, 1/a)) + \mathrm{O}(1)$.*

The reader is referred to [70] for a proof of Theorem 6.4. Theorem 6.4 is a good improvement over Theorem 6.1 if $a \ll m$. Consider, e.g., the congestion-minimization problem and its MIP formulation, sketched above; m here is the number of edges in G, and a is the maximum length of any of the flow-paths $P_{i,j}$. (This a was denoted d in Section 5.) To focus on a specific interesting case, suppose y^*, the fractional congestion, is at most one. Then while

the above-cited results (Theorems 6.1 and 7.2, resp.) give bounds of $O(\log m/\log\log m)$ and $O(a)$ on an integral solution, we get the improved bound of $O(\log a/\log\log a)$. Similar improvements are easily seen for other ranges of y^* also.

Thus, routing along *short* paths is very beneficial in keeping the congestion low. Indeed, specific useful classes of graphs such as *expanders* have been shown to be rich in the structure of such short paths [48, 39]. Recalling the graph-theoretic parameters such as Δ and β from Section 5, the work of [39] can essentially be used to ensure that $a = O(\Delta^2\beta^{-2}\log^3 n)$. So if, for instance, Δ and β^{-1} are $O(\text{polylog}(n))$, then a also can be bounded by $O(\text{polylog}(n))$. Hence, if $y^* \leq 1$, there exists an integral solution of congestion at most $O(\log\log n/\log\log\log n)$ in such cases, a good improvement over the $O(\log n/\log\log n)$ bound guaranteed by Theorem 6.1. See [70] for some further applications.

An outstanding open question in low-congestion routing is whether the integral optimum is always $O(1)$ if $y^* \leq 1$. See [38] for references and discussion related to this. As seen above, the integral optimum is $O(\log a/\log\log a)$, which is $O(\log n/\log\log n)$ in the worst case. Any improvement here will be very interesting.

7 A very short tour of discrepancy theory

Much of our discussion on LP-based randomized rounding has involved problems and results on "approximating" a given real vector x by a "rounded" (lattice) vector y, w.r.t. a linear system (matrix) A. Discrepancy theory studies general questions of this type. The survey [13] is an excellent presentation of discrepancy theory; we very briefly touch upon this vast area here.

For an $m \times n$ matrix A, its *discrepancy* is defined to be

$$\text{disc}(A) \doteq \min\{\|A\chi\|_\infty : \chi \in \{-1,1\}^n\}.$$

It has a close relationship to the lattice approximation problem [50]. Discrepancy theory is useful, in particular, in devising divide-and-conquer algorithms and for approximating NP-hard integer programs as discussed for the lattice approximation problem.

Recall that Sections 5 and 6 basically dealt with such rounding issues in the context of column-sparse integer programs. We continue the study of column-sparse integer programs here in the general context of discrepancy theory. A fundamental result of Beck & Fiala is that if each column of A has L_1 norm at most t, then $\text{disc}(A) < 2t$ *independent of* m *and* n [12]. This is an elegant constructive result based on linear algebra. A related famous conjecture of [12] is that if $A \in \{0,1\}^{m\times n}$ has at most t nonzero entries in each column, then $\text{disc}(A) = O(\sqrt{t})$; this, if true, would be best-possible.

A theorem closely related to, but incomparable with, the above Beck-Fiala theorem is

Theorem 7.1 ([36]) *Let A be a real valued $r \times s$ matrix, and y be a real-valued s-vector. Let b be a real valued vector such that $Ay = b$, and t be a positive real number such that in* **every column** *of A,* `(i)` *the sum of all the positive entries is $\leq t$,* **and** `(ii)` *the sum of all*

the negative entries is $\geq -t$. Then we can compute an integral vector $\overline{y}$ such that for every i, either $\overline{y}_i = \lfloor y_i \rfloor$ or $\overline{y}_i = \lceil y_i \rceil$, and $A\overline{y} = \overline{b}$ where $\overline{b}_i - b_i < t$ for all i. Furthermore, if y contains d distinct components, the integral approximation can be obtained in time $O(r^3 \log(1+s/r) + r^3 + d^2 r + sr)$.

Recall the parameters a and g of MIPs from Definition 6.2. Let us now see a simple application of Theorem 7.1 to MIPs:

Theorem 7.2 *Suppose a given feasible solution x^* to a given MIP's LP relaxation has objective function value some y^*. Then we can efficiently construct an integral solution to the MIP with objective function value less than $y^* + g$.*

Proof: The vector x^* satisfies:

(i) $\forall i \in [n]$ $\sum_j x^*_{i,j} = 1$, and

(ii) $(Ax^*)_i \leq y^*$ for all $i \in [m]$.

Also, $x^*_{i,j} \in [0,1]$ for all i, j. Rewrite (i) as "$\forall i \in [n]$ $(-g \sum_j x^*_{i,j}) = -g$". Consider the linear system given by this rewritten set of inequalities and (ii). It can be verified that the parameter t of Theorem 7.1 can be taken to be g here. Thus, by Theorem 7.1, we can efficiently round each $x^*_{i,j}$ to $x_{i,j} \in \{0,1\}$ in such a way that:

(a) $\forall i \in [n]$ $(-g \sum_j x_{i,j}) < -g + g$, and

(b) $(Ax)_i < y^* + g$ for all $i \in [m]$.

But since $x_{i,j} \in \{0,1\}$ for all i, j, (a) implies that for each i, $\sum_j x_{i,j} \geq 1$. This, in conjunction with (b), completes the proof. ■

Recall that for the packet-routing problem studied in Section 3.4, the routing paths were prespecified. But what if they are not, as is often the case? This issue has been handled recently in an approximation algorithm of [72]. Our proof of Theorem 7.2 is based on some of the ideas used there.

Returning to the Beck-Fiala conjecture, beautiful work of Beck and Spencer based on uniform randomized rounding, entropy and the pigeonhole principle has led to a non-constructive bound of $O(\sqrt{t} \log n \log t)$; the reader is referred to [10, 53] and to the second edition of [68] for a description of this nice method. In a recent breakthrough, the $O(\sqrt{t} \log n \log t)$ bound has been improved to $O(\sqrt{t \log n})$ by Banaszczyk [8].

8 Conclusion

We have seen a family of applications of randomized rounding to approximation algorithms: a central theme has been the use of correlation inequalities in the analysis and design of such algorithms.

In our view, one aspect of the results of Sections 4.3, 4.4 and 5 is noteworthy. It is not hard to show that for many of the underlying randomized rounding algorithms there, their probability of constructing a feasible solution of the desired quality is *exponentially small* in the worst-case. So, in themselves, they do not imply efficient (randomized) algorithms. However, the nature of the arguments involved helped us construct suitable potential functions that led to constructive versions. Recall that a similar situation exists for many applications of the LLL [11, 3, 54]. Thus, "probabilistic thinking" can lead to the only known efficient deterministic algorithms for many classes of problems. Another well-known point [48, 39] reiterated by our discussion of routing algorithms is the rich structure of multi-commodity flow on *short* paths for various classes of graphs. We expect continuing impetus to the general area studied here through the further development of, e.g., correlation inequalities.

Relaxations such as *semidefinite relaxations* that are more powerful than linear relaxations, have been used to obtain breakthrough approximation algorithms for some important problems; see, e.g., [28, 35]. We have not discussed these here. Also, rounding of linear and other relaxations is a vast area, many nice examples of which are out of our scope or have been left untouched here. In the context of scheduling, let us just mention one recent breakthrough result that has not been discussed here: the work of [31] and related recent papers on scheduling to minimize average job completion time under various scheduling models.

We conclude by presenting some open problems:

1. Resolve the approximation complexity of job-shop scheduling. In particular, is there a constant-factor approximation algorithm for this problem?

2. As asked in the last paragraph of Section 6, is the integral optimum OPT always O(1) in low-congestion routing if the fractional optimum y^* is at most 1? Simple examples show that OPT can sometimes be 2 even if $y^* = 1$, but nothing better is known. Even a family of examples in which $y^* \leq 1$ and OPT is at least 3, will be a good start.

3. As presented, Theorem 6.4 is *non-constructive*: it does not imply efficient randomized or deterministic algorithms. While constructive versions have been obtained for some important special cases (Leighton, Rao & Srinivasan [49], Lu [51]), the general problem remains open.

4. The classical Beck-Fiala conjecture is an outstanding problem to resolve: even a $o(t)$ discrepancy bound there would be a good start. In fact, even a discrepancy bound of $(2-\epsilon)t$ for some fixed $\epsilon > 0$ that holds for all sufficiently large t, may be a useful first step in this direction. The algorithmic question of efficiently constructing low-discrepancy lattice points for column-sparse matrices, is also of much interest.

Acknowledgements. I am grateful to Michał Karoński and Hans Jürgen Prömel for their kind invitation to give a short course in the Summer School on Randomized Algorithms, Antonin, Poland, 1997, and for their helpful comments on an earlier version. I would like to thank Andrzej Ruciński and Edyta Szymańska for their hospitality, and the participants at the summer school for their comments and suggestions. My gratitude also to David Grable and Alessandro Panconesi for their enthusiastic help on matters related to the summer school.

References

[1] R. Aharoni, P. Erdős, and N. Linial. Optima of dual integer linear programs. *Combinatorica*, 8:13–20, 1988.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] N. Alon. A parallel algorithmic version of the Local Lemma. *Random Structures & Algorithms*, 2:367–378, 1991.

[4] N. Alon, J. H. Spencer, and P. Erdős. *The Probabilistic Method*. Wiley–Interscience Series, John Wiley & Sons, Inc., New York, 1992.

[5] D. Angluin and L.G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155–193, 1979.

[6] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal of Computing*, 3:149–156, 1991.

[7] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced allocations. In *Proc. ACM Symposium on Theory of Computing*, pages 593–602, 1994.

[8] W. Banaszczyk. Balancing vectors and Gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12:351–360, 1998.

[9] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. Available in two parts as Technical Reports TRA1/98 and TRB1/98, Dept. of Information Systems & Computer Science, National University of Singapore, January 1998.

[10] J. Beck. Roth's estimate on the discrepancy of integer sequences is nearly sharp. *Combinatorica*, 1:319–325, 1981.

[11] J. Beck. An algorithmic approach to the Lovász Local Lemma. *Random Structures & Algorithms*, 2:343–365, 1991.

[12] J. Beck and T. Fiala. "Integer-making" theorems. *Discrete Applied Mathematics*, 3:1–8, 1981.

[13] J. Beck and V. T. Sós. Discrepancy theory. In *Handbook of combinatorics, Volume II*, chapter 26, pages 1405–1446. Elsevier Science B.V. and the MIT Press, 1995.

[14] J. Beck and J. H. Spencer. Integral approximation sequences. *Mathematical Programming*, 30:88–98, 1984.

[15] D. Bertsimas and R. Vohra. Rounding algorithms for covering problems. *Mathematical Programming*, 80:63–89, 1998.

[16] R. B. Boppana and J. H. Spencer. A useful elementary correlation inequality. *Journal of Combinatorial Theory, Series A*, 50:305–307, 1989.

[17] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35:164–176, 1989.

[18] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[19] D. Dubhashi and D. Ranjan. Balls and bins: a study in negative dependence. *Random Structures & Algorithms*, 13:99–124, 1998.

[20] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and Finite Sets*, A. Hajnal et. al., editors, Colloq. Math. Soc. J. Bolyai 11, North Holland, Amsterdam, pages 609–627, 1975.

[21] P. Erdős and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14:298–301, 1973.

[22] J. D. Esary and F. Proschan. Coherent structures of non-identical components. *Technometrics*, 5:191–209, 1963.

[23] U. Feige and C. Scheideler. Improved bounds for acyclic job shop scheduling. In *Proc. ACM Symposium on Theory of Computing*, pages 624–633, 1998.

[24] C. M. Fortuin, J. Ginibre, and P. N. Kasteleyn. Correlational inequalities for partially ordered sets. *Communications of Mathematical Physics*, 22:89–103, 1971.

[25] N. Garg, G. Konjevod and R. Ravi. A polylogarithmic approximation algorithm for the Group Steiner Tree problem. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259, 1998.

[26] R. L. Graham. Application of the FKG inequality and its relatives. In A. Bachem, M. Grötschel and B. Korte Ed., *Mathematical Programming: The State of the Art*, Springer–Verlag, 1983.

[27] M. X. Goemans and D. P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7:656–666, 1994.

[28] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[29] L. A. Goldberg, M. S. Paterson, A. Srinivasan, and E. Sweedyk. Better approximation guarantees for job-shop scheduling. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 599–608, 1997.

[30] G. H. Gonnet. Expected length of the longest probe sequence in hash code searching. *Journal of the ACM*, 28:289–304, 1981.

[31] L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein. Scheduling to minimize average completion time: off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22:513–544, 1997.

[32] D. S. Hochbaum, Editor. *Approximation algorithms for NP-hard problems.* PWS Press, 1997.

[33] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

[34] S. Janson, T. Łuczak and A. Ruciński. An exponential bound for the probability of nonexistence of a specified subgraph in a random graph. In *Random Graphs '87* (M. Karoński, J. Jaworski and A. Ruciński, eds.), John Wiley & Sons, Chichester, pages 73–87, 1990.

[35] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45:246–265, 1998.

[36] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.

[37] R. M. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. In *Proc. ACM Symposium on Theory of Computing*, pages 318–326, 1992.

[38] J. Kleinberg. *Approximation algorithms for disjoint paths problems.* PhD Thesis, Department of EECS, MIT, 1996.

[39] J. Kleinberg and R. Rubinfeld. Short paths in expander graphs. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 86–95, 1996.

[40] J. Kleinberg and É. Tardos. Disjoint paths in densely embedded networks. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 52–61, 1995.

[41] V. F. Kolchin, B. A. Sevastyanov, and V. P. Chistyakov. *Random Allocations.* John Wiley & Sons, 1978.

[42] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proc. MPS Conference on Integer Programming and Combinatorial Optimization*, pages 153–168, 1998. Lecture Notes in Computer Science 1412, Springer-Verlag.

[43] B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Editors. *Paths, Flows, and VLSI-Layout*. Springer-Verlag, Berlin, 1990.

[44] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: algorithms and complexity. In *Handbooks in Operations Research and Management Science*, Volume 4: Logistics of Production and Inventory, S. C. Graves et al., editors, Elsevier, pages 445–522, 1993.

[45] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays • Trees • Hypercubes*, Morgan Kaufmann, San Mateo, California, 1992.

[46] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and jobshop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14:167–186, 1994.

[47] F. T. Leighton, B. M. Maggs, and A. Richa. Fast algorithms for finding O(congestion + dilation) packet routing schedules. *Technical Report CMU-CS-96-152*, School of Computer Science, Carnegie-Mellon University, 1996. To appear in *Combinatorica*.

[48] F. T. Leighton and S. B. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[49] F. T. Leighton, S. B. Rao, and A. Srinivasan. New algorithmic aspects of the Local Lemma with applications to routing and partitioning. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[50] L. Lovász, J. H. Spencer, and K. Vesztergombi. Discrepancy of set systems and matrices. *European Journal of Combinatorics*, 7:151–160, 1986.

[51] C.-J. Lu. A deterministic approximation algorithm for a minmax integer programming problem. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[52] P. D. MacKenzie, C. G. Plaxton, and R. Rajaraman. On contention resolution protocols and associated probabilistic phenomena. *Journal of the ACM*, 45:324–378, 1998.

[53] J. Matoušek and J. H. Spencer. Discrepancy in arithmetic progressions. *Journal of the American Mathematical Society*, 9:195–204, 1996.

[54] M. Molloy and B. Reed. Further algorithmic aspects of the Local Lemma. In *Proc. ACM Symposium on Theory of Computing*, pages 524–529, 1998.

[55] R. Motwani, J. Naor, and P. Raghavan. Randomized approximation algorithms in combinatorial optimization. In *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, editor, PWS Press, 1997.

[56] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[57] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System Sciences*, 37:130–143, 1988.

[58] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[59] R. Ostrovsky and Y. Rabani. Universal O(congestion + dilation + $\log^{1+\epsilon} N$) local control packet switching algorithms. In *Proc. ACM Symposium on the Theory of Computing*, pages 644–653, 1997.

[60] G. Rayzman. Approximation techniques for job-shop scheduling problems. *MSc Thesis*, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Israel, July 1996.

[61] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8:223–250, 1995.

[62] S. V. Sevast'yanov. Efficient construction of schedules close to optimal for the cases of arbitrary and alternative routes of parts. *Soviet Math. Dokl.*, 29:447–450, 1984.

[63] S. V. Sevast'yanov. Bounding algorithm for the routing problem with arbitrary paths and alternative servers. *Kibernetika* 22:74–79, 1986 (translation in *Cybernetics* 22, pages 773–780).

[64] D. R. Shier. *Network Reliability and Algebraic Structures*. Oxford Science Publications, Oxford University Press, 1991.

[65] D. B. Shmoys. Computing near-optimal solutions to combinatorial optimization problems. In *Combinatorial Optimization* (W. Cook, L. Lovász and P. D. Seymour, eds.), American Mathematical Society, pages 355–397, 1995.

[66] D. B. Shmoys, C. Stein, and J. Wein. Improved approximation algorithms for shop scheduling problems. *SIAM J. Comput.*, 23:617–632, 1994.

[67] J. H. Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289:679–706, 1985.

[68] J. H. Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, Philadelphia, 1987. (Second edition published in 1994.)

[69] A. Srinivasan. Improved approximations of packing and covering problems. In *Proc. ACM Symposium on Theory of Computing*, pages 268–276, 1995. Full version to appear in *SIAM J. Comput.*, as "Improved approximation guarantees for packing and covering integer programs".

[70] A. Srinivasan. An extension of the Lovász Local Lemma, and its applications to integer programming. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 6–15, 1996.

[71] A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 416–425, 1997.

[72] A. Srinivasan and C.-P. Teo. A constant-factor approximation algorithm for packet routing, and balancing local vs. global criteria. In *Proc. ACM Symposium on Theory of Computing*, pages 636–643, 1997.

[73] C.-P. Teo. *Constructing approximation algorithms via linear programming relaxations: primal dual and randomized rounding techniques.* Ph.D. Thesis, Sloan School of Management, MIT, September 1996.

[74] C.-P. Teo and J. Sethuraman. LP based approach to optimal stable matchings. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 710–719, 1997.

[75] L. G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11:350–361, 1982.

[76] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proc. ACM Symposium on the Theory of Computing*, pages 263–277, 1981.

[77] M. Yannakakis. On the approximation of maximum satisfiability. *Journal of Algorithms*, 17:475–502, 1994.