Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Stochastic Load Balancing on Unrelated Machines

Anupam Gupta, Amit Kumar, Viswanath Nagarajan, Xiangkun Shen

informs
http://pubsonline.informs.org/journal/moor
MATHEMATICS OF OPERATIONS RESEARCH
Vol. 46, No. 1, February 2021, pp. 115–133
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Stochastic Load Balancing on Unrelated Machines

Anupam Gupta,[a] Amit Kumar,[b] Viswanath Nagarajan,[c] Xiangkun Shen[c]

[a] Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213; [b] Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi 110 016, India; [c] Industrial and Operations Engineering Department, University of Michigan, Ann Arbor, Michigan 48109

Contact: anupamg@cs.cmu.edu (AG); amitk@cse.iitd.ac.in (AK); viswa@umich.edu, https://orcid.org/0000-0002-9514-5581 (VN); xkshen@umich.edu (XS)

Received: December 28, 2018
Revised: July 22, 2019
Accepted: December 13, 2019
Published Online in Articles in Advance: August 24, 2020

MSC2000 Subject Classification: Primary: 68W25, 90B36, 90C15
OR/MS Subject Classification: Primary: Production/scheduling; secondary: analysis of algorithms

https://doi.org/10.1287/moor.2019.1049

Abstract. We consider the problem of makespan minimization on unrelated machines when job sizes are stochastic. The goal is to find a fixed assignment of jobs to machines, to minimize the expected value of the maximum load over all the machines. For the identical-machines special case when the size of a job is the same across all machines, a constant-factor approximation algorithm has long been known. Our main result is the first constant-factor approximation algorithm for the general case of unrelated machines. This is achieved by (i) formulating a lower bound using an exponential-size linear program that is efficiently computable and (ii) rounding this linear program while satisfying only a specific subset of the constraints that still suffice to bound the expected makespan. We also consider two generalizations. The first is the *budgeted* makespan minimization problem, where the goal is to minimize the expected makespan subject to scheduling a target number (or reward) of jobs. We extend our main result to obtain a constant-factor approximation algorithm for this problem. The second problem involves *q-norm* objectives, where we want to minimize the expected q-norm of the machine loads. Here we give an $O(q/\log q)$-approximation algorithm, which is a constant-factor approximation for any fixed q.

Funding: A. Gupta was supported in part by the National Science Foundation (NSF) Division of Computing and Communication Foundations [Grants CCF-1536002, CCF-1540541, and CCF-1617790], and V. Nagarajan and X. Shen were supported in part by an NSF CAREER grant [Grant CCF-1750127].

Keywords: stochastic optimization • approximation algorithms • scheduling

1. Introduction

We consider the problem of scheduling jobs on machines to minimize the maximum load (i.e., the problem of *makespan minimization*). This is a classic NP-hard problem, with Graham's [5] list scheduling algorithm for identical machines being one of the earliest approximation algorithms known. If the job sizes are deterministic, the problem is fairly well understood, with polynomial time approximation schemes (PTASs) for the cases of identical (Hochbaum and Shmoys [7]) and related machines (Hochbaum and Shmoys [8]) and a constant-factor approximation and APX-hardness (Lenstra et al. [17], Shmoys and Tardos [26]) for the case of unrelated machines. Given we understand the basic problem well, it is natural to consider settings that are less stylized, and one step closer to modeling real-world scenarios: *what can we do if there is uncertainty in the job sizes?*

We take a stochastic optimization approach where the job sizes are random variables (RVs) with known distributions. In particular, the size of each job j on machine i is given by a random variable X_{ij}. Throughout this paper, we assume that the sizes of different jobs are independent of each other. Given just this information, an algorithm has to assign these jobs to machines, resulting in, say, jobs J_i being assigned to each machine i. The expected makespan of this assignment is

$$\mathbb{E}\left[\max_{i=1}^{m} \sum_{j\in J_i} X_{ij}\right], \tag{1}$$

where m denotes the number of machines. The goal for the algorithm is to minimize this expected makespan. Observe that the entire assignment of jobs to machines is done up front without knowledge of the actual outcomes of the random variables, and hence there is no adaptivity in this problem.

Such stochastic load-balancing problems are common in real-world systems where the job sizes are indeed not known, but given the large amounts of data, one can generate reasonable estimates for the distribution. Moreover, static (nonadaptive) assignments are preferable in many applications, as they are easier to implement.

Inspired by work on scheduling and routing problems in several communities, Kleinberg et al. [15] first posed the problem of approximating the expected makespan in 1997. They gave a constant-factor approximation for the identical-machines case, that is, for the case where for each job j, the sizes $X_{ij} = X_{i'j}$ for all $i, i' \in [m]$. A key concept in their result was the *effective size* of a random variable (due to Hui [9]), which is a suitably scaled logarithm of the moment-generating function. This effective size (denoted by β_m) depended crucially on the number m of machines. Roughly speaking, the algorithm in Kleinberg et al. [15] solved the *determinisitic* makespan minimization problem by using the effective size $\beta_m(X_j)$ of each job j as its deterministic size. The main part of their analysis involved proving that the resulting schedule also has small *expected* makespan when viewed as a solution to the stochastic problem. See Section 2 for a more detailed discussion.

Later, Goel and Indyk [4] gave better approximation ratios for special classes of job size distributions, again for identical machines. Despite these improvements and refined understanding of the identical-machines case, the above stochastic load-balancing problem has remained open, even for the related-machines setting. Recall that *related machines* refers to the case where each machine i has a *speed* s_i, and the sizes for each job j satisfy $X_{ij}\, s_i = X_{i'j}\, s_{i'}$ for all $i, i' \in [m]$.

1.1. Results and Techniques

Our main result is as follows:

Theorem 1. *There is an $O(1)$-approximation algorithm for the problem of finding an assignment to minimize the expected makespan on unrelated machines.*

Our work naturally builds on the edifice of Kleinberg et al. [15]. However, we need several new ideas to achieve this. As mentioned above, the prior result for identical machines used the notion of effective size, which depends on the number m of machines available. When machines are not identical, consider just the "restricted assignment" setting, where each job needs to choose from a specific subset of machines: here it is not even clear how to define the effective size of each job. Instead of working with a single deterministic value as the effective size of any random variable X_{ij}, we use all the $\beta_k(X_{ij})$ values for $k = 1, 2, \ldots, m$.

Then, we show that in an optimal solution, for every k-subset of machines, the total β_k effective size of jobs assigned to those machines is at most some bound depending on k. Such a property for $k = m$ was also used in the algorithm for identical machines. We then formulate a linear program (LP) relaxation that enforces such a "volume" constraint for *every* subset of machines. Although our LP relaxation has an exponential number of constraints, it can be solved in polynomial time using the ellipsoid algorithm and a suitable separation oracle.

Finally, given an optimal solution to this LP, we show how to carefully choose the right parameter for the effective size of each job and use it to build an approximately optimal schedule. Although our LP relaxation has an exponential number of constraints (and it seems difficult to preserve them all), we show that it suffices to satisfy a small subset of these constraints in the integral solution. Roughly, our rounding algorithm uses the LP solution to identify the "correct" deterministic size for each job and then applies an existing algorithm for deterministic scheduling (Shmoys and Tardos [26]).

1.1.1. Budgeted Makespan Minimization. In this problem, each job j has a *reward* r_j (having no relationship to other parameters such as its size), and we are given a target reward value R. The goal is to assign some subset $S \subseteq [n]$ of jobs whose total reward $\sum_{j\in S} r_j$ is at least R, and to minimize the expected makespan of this assignment. Clearly, this generalizes the basic makespan minimization problem (by setting all rewards to one and the target $R = n$).

Theorem 2. *There is an $O(1)$-approximation algorithm for the budgeted makespan minimization problem on unrelated machines.*

To solve this, we extend the ideas for expected makespan scheduling to include an extra constraint about high reward. We again write a similar LP relaxation. Rounding this LP requires some additional ideas on top of those in Theorem 1. The new ingredient is that we need to round solutions to an assignment LP with two linear constraints. To do this without violating the budget, we utilize a "reduction" from the generalized assignment problem (GAP) to bipartite matching (Shmoys and Tardos [26]) as well as certain adjacency properties of the bipartite-matching polytope (Balinski and Russakoff [2]).

1.1.2. Minimizing ℓ_q Norms. Finally, we consider the problem of stochastic load balancing under ℓ_q norms. Given an assignment of J_i of jobs to machines, we denote the "load" on machine i by $L_i := \sum_{j \in J_i} X_{ij}$, and the "load vector" by $\mathbf{L} = (L_1, L_2, \ldots, L_m)$. The expected makespan minimization problem is to minimize $\mathbb{E}[\|\mathbf{L}\|_\infty]$. The *q-norm minimization problem* is the following: find an assignment of jobs to machines to minimize

$$\mathbb{E}\Big[\|\mathbf{L}\|_q\Big] = \mathbb{E}\left[\left(\sum_{i=1}^{m}\left(\sum_{j \in J_i} X_{ij}\right)^q\right)^{1/q}\right].$$

Our result for this setting is the following:

Theorem 3. *There is an $O(\frac{q}{\log q})$-approximation algorithm for the stochastic q-norm minimization problem on unrelated machines.*

The main idea here is to reduce this problem to a suitable instance of deterministic q-norm minimization with additional side constraints. We then show that existing techniques for deterministic q-norm minimization (Azar and Epstein [1]) can be extended to obtain a constant-factor approximation for our generalization as well. We also need to use/prove some probabilistic inequalities to relate the deterministic subproblem to the stochastic problem. We note that using general polynomial concentration inequalities (Kim and Vu [14], Schudy and Sviridenko [25]) only yields an approximation ratio that is exponential in q. We obtain a much better $O(q/\log q)$-approximation factor by utilizing the specific form of the norm function. Specifically, we use the convexity of norms, a second-moment calculation, and a concentration bound (Johnson et al. [12]) for the qth moment of sums of independent random variables.

We note that Theorem 3 implies a constant-factor approximation for any fixed $q \geq 1$. However, our techniques do not extend directly to provide an $O(1)$-approximation algorithm for all q-norms.

1.2. Other Related Work

The stochastic load-balancing problem on identical machines has also been studied for specific job-size distributions: Goel and Indyk [4] showed that Graham's [5] algorithm achieves a 2-approximation for Poisson distributions, and they obtained a PTAS for exponential distributions. Kleinberg et al. [15] also considered stochastic versions of knapsack and bin-packing problems: given an overflow probability p, a feasible single-bin packing here corresponds to any subset of jobs such that their total size exceeds one with probability at most p. Goel and Indyk [4] gave better/simpler algorithms for these problems, under special distributions.

Recently, Deshpande and Li [18] and Li and Yuan [19] considered several combinatorial optimization problems, including shortest paths and minimum spanning trees, where elements have weights (which are random variables) and one would like to find a solution (i.e., a subset of elements) whose expected utility is maximized. These results also apply to the stochastic versions of knapsack and bin-packing problems from Kleinberg et al. [15] and yield bicriteria approximations. The main technique here was a clever discretization of probability distributions. However, to the best of our knowledge, such an approach is not applicable to stochastic load balancing.

Stochastic scheduling has been studied in many different contexts, in different fields (see, e.g., Pinedo [23]). The work on approximation algorithms for these problems is more recent; see Möhring et al. [21] for some early work and many references. In this paper, we consider the (*nonadaptive*) *fixed assignment model*, where jobs have to be assigned to machines up front, and then the randomness is revealed. Hence, there is no element of adaptivity in these problems. This makes them suitable for settings where the decisions cannot be instantaneously implemented (e.g., for virtual circuit routing, or assigning customers to physically well-separated warehouses). A number of papers (Gupta et al. [6], Im et al. [11], Megow et al. [20], Möhring et al. [21]) have considered scheduling problems in the *adaptive* setting, where assignments are done online and the assignment for a job may depend on the state of the system at the time of its assignment. See Section 2 for a comparison of adaptive and nonadaptive settings in the load-balancing problem.

Very recently (after the preliminary version of this paper appeared), Molinaro [22] obtained an $O(1)$-approximation algorithm for the stochastic q-norm problem for all $q \geq 1$, which improves over Theorem 3. In addition to the techniques in our paper, the main idea in Molinaro [22] is to use a different notion of effective size, based on the L-function method (Latała [16]). We still present our algorithm/analysis for Theorem 3, as it is conceptually simpler and may provide better constant factors for small q.

2. Preliminaries

The stochastic load-balancing problem (StocMakespan) involves assigning n jobs to m machines. For each job $j \in [n]$ and machine $i \in [m]$, we are given a random variable X_{ij} that denotes the processing time (size) of job j on machine i. We assume that the random variables $X_{ij}, X_{i'j'}$ are independent when $j \neq j'$ (the sizes of job j on different machines may be correlated). We assume access to the distribution of these random variables via some (succinct) representation. A solution is a partition $\{J_i\}_{i=1}^m$ of the jobs among the machines, such that $J_i \subseteq [n]$ is the subset of jobs assigned to machine $i \in [m]$. The expected makespan of this solution is $\mathbb{E}[\max_{i=1}^m \sum_{j\in J_i} X_{ij}]$. Our goal is to find a solution that minimizes the expected makespan.

The deterministic load-balancing problem is known to be $\mathcal{NP}$-hard even on identical machines. The stochastic version introduces considerable additional complications. For example, Kleinberg et al. [15] showed that, given Bernoulli random variables $\{X_j\}$, it is #$\mathcal{P}$-hard to compute the overflow probability, that is, $\Pr[\sum_j X_j > B]$ for some target B. (Note that a Bernoulli random variable X is specified by two parameters: a size $s > 0$ and a probability q, which means that $X = s$ with probability q and $X = 0$ otherwise.) We now show that it is #$\mathcal{P}$-hard even to compute the objective value of a given assignment in the identical-machines setting.

Theorem 4. *It is #$\mathcal{P}$-hard to compute the expected makespan of a given assignment for stochastic load balancing on identical machines.*

Proof. We will reduce the overflow probability problem (Kleinberg et al. [15]) to this problem. An instance of the overflow probability problem consists of a target integer B and independent Bernoulli trials $Y_1, \ldots, Y_n$, where each Y_j has integer size s_j with probability q_j. The objective is to compute $\Pr(\sum_{j=1}^n Y_j > B)$. This problem was shown to be #$\mathcal{P}$-hard in Kleinberg et al. [15, theorem 2.1].

From an instance of the overflow probability problem, we construct two instances of stochastic load balancing. In addition to the n jobs given by $Y_1, \ldots, Y_n$, we have job Z_B of deterministic size B and job Z_{B+1} of deterministic size $B+1$. The first instance $\mathcal{I}'$ of stochastic load balancing contains $m = 2$ machines and jobs $Y_1, \ldots, Y_n$ and Z_B. The second instance $\mathcal{I}''$ contains $m = 2$ machines and jobs $Y_1, \ldots, Y_n$ and Z_{B+1}. We want to compute the expected makespan of the following assignment for both instances: assign jobs $Y_1, \ldots, Y_n$ to machine 1 and the remaining job to machine 2. We use Obj_B and Obj_{B+1} to denote the expected makespans of instances $\mathcal{I}'$ and $\mathcal{I}''$. Then we have

$$Obj_B = \mathbb{E}\left[\max\left\{B, \sum_{j=1}^n Y_j\right\}\right] = B + \mathbb{E}\left[\max\left\{0, \sum_{j=1}^n Y_j - B\right\}\right]$$
$$= B + \sum_{t \geq B+1} \Pr\left(\sum_{j=1}^n Y_j \geq t\right).$$

Similarly,

$$Obj_{B+1} = B + 1 + \sum_{t \geq B+2} \Pr\left(\sum_{j=1}^n Y_j \geq t\right).$$

It follows that $\Pr(\sum_{j=1}^n Y_j > B) = \Pr(\sum_{j=1}^n Y_j \geq B+1) = Obj_B - Obj_{B+1} + 1$. So, if we could compute the expected makespan of any given assignment for stochastic load balancing, then we can also compute the overflow probability, which is #$\mathcal{P}$-hard. This completes the proof. □

2.1. Scaling the Optimal Value

Using a standard binary search approach, in order to obtain an $O(1)$-approximation algorithm for StocMakespan, it suffices to solve the following problem. Given a bound $M > 0$, either find a solution with expected makespan $O(M)$ or establish that the optimal makespan is $\Omega(M)$. Moreover, by scaling down all random variables by factor M, we may assume that the target makespan is one.

We now provide some definitions and background results that will be used extensively in the rest of this paper.

2.2. Truncated and Exceptional Random Variables

It is convenient to divide each random variable X_{ij} into its *truncated* and *exceptional* parts, defined below:

- $X'_{ij} := X_{ij} \cdot \mathbf{I}_{(X_{ij} \leq 1)}$ (called the *truncated* part) and
- $X''_{ij} := X_{ij} \cdot \mathbf{I}_{(X_{ij} > 1)}$ (called the *exceptional* part).

Note that $X_{ij} = X'_{ij} + X''_{ij}$. The reason for doing this is that these two kinds of random variables behave very differently with respect to the expected makespan. It turns out that expectation is a good notion of deterministic size for exceptional RVs, whereas one needs a more nuanced notion (called effective size) for truncated RVs; this is discussed in detail below.

We will use the following result (which follows from Kleinberg et al. [15]) to handle exceptional RVs.

Lemma 1 (Exceptional Items' Lower Bound). *Let $X_1, X_2, \ldots, X_t$ be nonnegative discrete random variables, each taking value zero or at least L. If $\sum_j \mathbb{E}[X_j] > L$, then $\mathbb{E}[\max_j X_j] > L/2$.*

Proof. The Bernoulli case of this lemma appears as lemma 3.3 in Kleinberg et al. [15]. Although lemma 3.3 of Kleinberg et al. [15] is stated for nonstrict inequalities, it also holds for strict inequalities. The extension to the general case is easy. For each X_j, introduce independent Bernoulli random variables $\{X_{jk}\}$, where each X_{jk} corresponds to a particular instantiation s_{jk} of X_j, that is, $\Pr[X_{jk} = s_{jk}] = \Pr[X_j = s_{jk}]$. Note that $\max_k X_{jk}$ is stochastically dominated by X_j, so $\mathbb{E}[\max_j X_j] \geq \mathbb{E}[\max_{jk} X_{jk}]$. Moreover, $\sum_{jk} \mathbb{E}[X_{jk}] = \sum_j \mathbb{E}[X_j] > L$. So the lemma follows from the Bernoulli case. □

2.3. Effective Size and Its Properties

As is often the case for stochastic optimization problems, we want to find some deterministic quantity that is a good surrogate for each random variable, and then use this deterministic surrogate instead of the actual random variable. Here, we use the *effective size*, which is based on the logarithm of the (exponential) moment-generating function (Elwalid and Mitra [3], Hui [9], Kelly [13]).

Definition 1 (Effective Size). For any random variable X and integer $k \geq 2$, define

$$\beta_k(X) \;:=\; \frac{1}{\log k} \cdot \log \mathbb{E}\Big[e^{(\log k)\cdot X}\Big]. \tag{2}$$

Also define $\beta_1(X) := \mathbb{E}[X]$.

To get some intuition for this, consider independent RVs $Y_1, \ldots, Y_n$. Then if $\sum_i \beta_k(Y_i) \leq b$,

$$\Pr\Big[\sum_i Y_i \geq c\Big] = \Pr\Big[e^{\log k \sum_i Y_i} \geq e^{(\log k)c}\Big] \leq \frac{\mathbb{E}\big[e^{\log k \sum_i Y_i}\big]}{e^{(\log k)c}} = \frac{\prod_i \mathbb{E}\Big[e^{(\log k)Y_i}\Big]}{e^{(\log k)c}}.$$

Taking logarithms, we get

$$\log \Pr\Big[\sum_i Y_i \geq c\Big] \leq \log k \cdot \Big[\sum_i \beta_k(Y_i) - c\Big] \quad \Rightarrow \quad \Pr\Big[\sum_i Y_i \geq c\Big] \leq \frac{1}{k^{c-b}}.$$

The above calculation, very reminiscent of the standard Chernoff bound argument, can be summarized by the following lemma (shown, e.g., in Hui [9]).

Lemma 2 (Upper Bound). *For independent random variables $Y_1, \ldots, Y_n$, if $\sum_i \beta_k(Y_i) \leq b$, then $\Pr[\sum_i Y_i \geq c] \leq (1/k)^{c-b}$.*

The usefulness of this definition comes from a partial converse, proved in Kleinberg et al. [15, lemma 3.2]:

Lemma 3 (Lower Bound). *Consider independent Bernoulli random variables $Y_1, \ldots, Y_n$, where each Y_i has nonzero size s_i being an inverse power of 2 such that $1/(\log k) \leq s_i \leq 1$. If $\sum_i \beta_k(Y_i) \geq 7$, then $\Pr[\sum_i Y_i \geq 1] \geq 1/k$.*

2.3.1. Outline of the Algorithm for Identical Machines.

In using the effective size, it is important to set the parameter k carefully. For identical machines, Kleinberg et al. [15] used $k = m$, the total number of machines. Using the facts discussed above, we can now outline their algorithm/analysis (assuming that all RVs are truncated). If the total effective size is at most, say, $20m$, then the jobs can be assigned to m machines in a way that the effective-size load on each machine is at most 21. By Lemma 2 and a union bound, it follows that the probability of some machine exceeding load 23 is at most $m \cdot (1/m)^2 = 1/m$. On the other hand, if the total effective size is more than $20m$, then even if the solution was to balance these evenly, each machine would have effective-size load at least 7. By Lemma 3, it follows that the load on each machine exceeds one with probability $1/m$, and so with m machines, this gives a certificate that the makespan is $\Omega(1)$.

2.3.2. Challenges with Unrelated Machines. For unrelated machines, this kind of argument breaks down even in the restricted-assignment setting, where each job can go on only some subset of machines. This is because we do not know what probability of success we want to aim for. For example, even if the machines had the same speed, but there were jobs that could go on only $\sqrt{m}$ of these machines, and others could go on the remaining $m - \sqrt{m}$ of them, we would want their effective sizes to be quite different (see the example in Section 2.3.3). And once we go to general unrelated machines, it is not clear whether any combinatorial argument would suffice. Instead, we propose an LP-based lower bound that enforces one such constraint (involving effective sizes) for every subset of machines.

2.3.3. Bad Example for Simpler Effective Sizes. For stochastic load balancing on identical machines, Kleinberg et al. [15] showed that any algorithm that maps each RV to a single real value and performs load balancing on these (deterministic) values incurs an $\Omega(\frac{\log m}{\log\log m})$ approximation ratio. This is precisely the reason they introduced the notion of truncated and exceptional RVs. For truncated RVs, their algorithm showed that it suffices to use $\beta_m(X_j)$ as the deterministic value and perform load balancing with respect to these. Exceptional RVs were handled separately (in a simpler manner). For unrelated machines, we now provide an example that shows that even when all RVs are truncated, any algorithm that maps each RV to a single real value must incur approximation ratio at least $\Omega(\frac{\log m}{\log\log m})$. This suggests that more work is needed to define the "right" effective sizes in the unrelated machine setting.

There are $m+1$ machines and $m+\sqrt{m}$ jobs. Each RV X_j takes the value 1 with probability $\frac{1}{\sqrt{m}}$ (and 0 otherwise). The first $\sqrt{m}$ jobs can only be assigned to machine 1. The remaining m jobs can be assigned to any machine. Note that $OPT \le 2$, which is obtained by assigning the first $\sqrt{m}$ jobs to machine 1, and each of the remaining m jobs in a one-to-one manner. Given any fixed mapping of RVs to reals, note that all the X_j get the same value (say, θ) as they are identically distributed. So the optimal value of the corresponding (deterministic) load-balancing instance is $\sqrt{m}\cdot\theta$. Hence, the solution which maps $\sqrt{m}$ jobs to each of the first $1+\sqrt{m}$ machines is an optimal solution to the deterministic instance. However, the expected makespan of this assignment is $\Omega(\frac{\log m}{\log\log m})$.

We will use the following specific result in dealing with truncated RVs.

Lemma 4 (Truncated Items' Lower Bound)**.** *Let $X_1, X_2, \ldots, X_n$ be independent $[0,1]$ RVs, and let $\{J_i\}_{i=1}^m$ be any partition of $[n]$. If $\sum_{j=1}^n \beta_m(X_j) \ge 17m$, then $\mathbb{E}[\max_{i=1}^m \sum_{j\in J_i} X_j] = \Omega(1)$.*

Proof. This is a slight extension of Kleinberg et al. [15, lemma 3.4], with two main differences. First, we want to consider arbitrary instead of just Bernoulli RVs. Second, we use a different definition of effective size than they do. We provide the details below.

At the loss of factor 2 in the makespan, we may assume (by rounding down) that the only values taken by the X_j RVs are inverse powers of 2. For each RV X_j, applying lemma 3.10 of Kleinberg et al. [15] yields independent Bernoulli random variables $\{Y_{jk}\}$ so that for each power-of-2 value s we have

$$\Pr\big[X_j = s\big] = \Pr\left[s \le \sum_k Y_{jk} < 2s\right].$$

Let $\overline{X}_j = \sum_k Y_{jk}$. It now follows that X_j is stochastically dominated by $\overline{X}_j$, and $\overline{X}_j$ is stochastically dominated by $2\cdot X_j$. Moreover, $\beta_m(\overline{X}_j) = \sum_k \beta_m(Y_{jk})$. Note also that $\beta_m(\overline{X}_j) \ge \beta_m(X_j)$. Hence, $\sum_{jk}\beta_m(Y_{jk}) \ge \sum_{j=1}^n \beta_m(X_j) \ge 17m$. Now, consider the assignment of the Y_{jk} RVs corresponding to $\{J_i\}_{i=1}^m$; that is, for each $i \in [m]$ and $j \in J_i$, all the $\{Y_{jk}\}$ RVs are assigned to part i. We now apply lemma 3.4 of Kleinberg et al. [15], which works for Bernoulli RVs to lower bound the expected makespan. Note that the above lemma used a different notion of effective size: $\beta'_{1/m}(X) := \min\{s, sqm^s\}$ for any Bernoulli RV X taking value s with probability q. However, as shown in Kleinberg et al. [15, proposition 2.5], $\beta_m(X) \le \beta'_{1/m}(X)$, which implies $\sum_{j=1}^n \beta'_{1/m}(X_j) \ge \sum_{j=1}^n \beta_m(X_j) \ge 17m$, as required by Kleinberg et al. [15, lemma 3.4]. So we have $\mathbb{E}[\max_{i=1}^m \sum_{j\in J_i}\sum_k Y_{jk}] = \Omega(1)$. Finally, using the fact that X_j stochastically dominates $\frac{1}{2}\overline{X}_j$,

$$\mathbb{E}\left[\max_{i=1}^m \sum_{j\in J_i} X_j\right] \ge \frac{1}{2}\mathbb{E}\left[\max_{i=1}^m \sum_{j\in J_i} \overline{X}_j\right] = \frac{1}{2}\mathbb{E}\left[\max_{i=1}^m \sum_{j\in J_i}\sum_k Y_{jk}\right] = \Omega(1),$$

which completes the proof. □

Remark 1. The constant in the $\Omega(1)$ notation of Lemma 4 is at least $\frac{1}{2}(1-\frac{1}{e}) \geq 0.31$. The factor $\frac{1}{2}$ comes from the reduction from general to Bernoulli RVs, and the factor $1 - 1/e$ comes from the proof of Kleinberg et al. [15, lemma 3.4] for Bernoulli RVs.

2.4. Nonadaptive and Adaptive Solutions

We note that our model involves computing an assignment that is fixed a priori, before observing any random instantiations. Such solutions are commonly called nonadaptive. A different class of solutions (called adaptive) involves assigning jobs to machines sequentially, observing the random instantiation of each assigned job. Designing approximation algorithms for the adaptive and nonadaptive models are mutually incomparable. For makespan minimization on identical machines, Graham's [5] list scheduling already gives a trivial 2-approximation algorithm in the *adaptive* case (in fact, it is 2-approximate on a per-instance basis), whereas the *nonadaptive* case is quite nontrivial, and the Kleinberg et al. [15] result was the first constant-factor approximation.

We now provide an instance with identical machines where there is an $\Omega(\frac{\log m}{\log\log m})$ gap between the best nonadaptive assignment (the setting of this paper) and the best adaptive assignment. The instance consists of m machines and $n = m^2$ jobs, each of which is identically distributed taking size 1 with probability $\frac{1}{m}$ (and 0 otherwise). Recall that Graham's [5] algorithm considers jobs in any order and places each job on the least loaded machine. It follows that the expected makespan of this adaptive policy is at most $1 + \frac{1}{m} \cdot \mathbb{E}[\sum_{j=1}^{m^2} X_j] = 2$. On the other hand, the best static assignment has expected makespan $\Omega(\frac{\log m}{\log\log m})$, which is obtained by assigning m jobs to each machine.

2.5. Useful Probabilistic Inequalities

Theorem 5 (Jensen's Inequality)**.** *Let $X_1, X_2, \ldots, X_t$ be random variables and $f(x_1, \ldots, x_t)$ any convex function. Then*

$$\mathbb{E}\big[f(X_1, \ldots, X_t)\big] \geq f(\mathbb{E}[X_1], \ldots, \mathbb{E}[X_t]).$$

Theorem 6 (Rosenthal Inequality; Johnson et al. [12], Latała [16], Rosenthal [24])**.** *Let $X_1, X_2, \ldots, X_t$ be independent nonnegative random variables. Let $q \geq 1$ and $K = \Theta(q/\log q)$. Then it is the case that*

$$\mathbb{E}\left[\left(\sum_j X_j\right)^q\right] \leq K^q \cdot \max\left\{\left(\sum_j \mathbb{E}\big[X_j\big]\right)^q, \sum_j \mathbb{E}\Big[X_j^q\Big]\right\}.$$

3. Makespan Minimization

The main result of this section is Theorem 1: *There is an $O(1)$-approximation algorithm for the problem of finding an assignment to minimize the expected makespan on unrelated machines.*

Using a binary search scheme and scaling, it suffices to find one of the following:

(i) an *upper bound*, a solution with expected makespan at most $O(1)$, or

(ii) a *lower bound*, a certificate that the optimal expected makespan is more than one.

Hence, we assume that the optimal solution for the instance has unit expected makespan, and try to find a solution with expected makespan $O(1)$; if we fail, we output a lower bound certificate.

At a high level, the ideas we use are the following: First, in Section 3.1, we show a more involved lower bound based on the effective sizes of jobs assigned to every subset of machines. This is captured using an exponentially sized LP that is solvable in polynomial time. Then, to show that this lower bound is a good one, we give a new rounding algorithm for this LP in Section 3.2 to get an expected makespan within a constant factor of the lower bound.

3.1. A New Lower Bound

Our starting point is a more general lower bound on the makespan. The (contrapositive of the) following lemma says that if the effective sizes are large, then the expected makespan must be large too. This is in the same spirit as Lemma 3, but for the general setting of unrelated machines. The first inequality below, (3), focuses on the exceptional parts and loosely follows from the intuition that if the sum of biases of a set of independent coin flips is large (exceeds two in this case), then you expect one of them to come up heads. The second inequality, (4), focuses on the truncated parts and applies to *every* subset $K \subseteq [m]$ of machines.

Lemma 5 (New Valid Inequalities). *Consider any feasible solution that assigns jobs J_i to each machine $i \in [m]$. If the expected makespan $\mathbb{E}[\max_{i=1}^{m} \sum_{j \in J_i} X_{ij}] \le 1$, then*

$$\sum_{i=1}^{m} \sum_{j \in J_i} \mathbb{E}\left[X''_{ij}\right] \le 2, \qquad \textit{and} \tag{3}$$

$$\sum_{i \in K} \sum_{j \in J_i} \beta_k\left(X'_{ij}\right) \;\le\; O(1) \cdot k, \;\; \textit{for all } K \subseteq [m], \quad \textit{where } k = |K|. \tag{4}$$

Proof. The first inequality, (3), follows from Lemma 1 applied to $\{X''_{ij} : j \in J_i, i \in [m]\}$.

For the second inequality, (4), consider any subset $K \subseteq [m]$ of the machines. Then, applying Lemma 4 only to the k machines in K and the truncated random variables $\{X'_{ij} : i \in K, j \in J_i\}$ corresponding to jobs assigned to these machines, we obtain the desired inequality. □

Remark 2. Using explicit constants from Lemma 4, one can also obtain an explicit constant in the $O(1)$ term in Lemma 5. In particular, if the expected makespan is at most $\frac{1}{2}(1 - \frac{1}{e})$ (instead of one), then the $O(1)$ term in (4) is 17, and the constant 2 in (3) is still valid.

Given these valid inequalities, our algorithm now seeks an assignment satisfying (3) and (4). If we fail, the lemma assures us that the expected makespan must be large. On the other hand, if we succeed, such a "good" assignment by itself is not sufficient. The challenge is to show the converse of Lemma 5, that is, that any assignment satisfying (3) and (4) gives us an expected makespan of $O(1)$.

Indeed, toward this goal, we first write an LP relaxation with an exponential number of constraints, corresponding to (4). We can solve this LP using the ellipsoid method. Then, instead of rounding the fractional solution to satisfy all constraints (which seems very hard), we show how to satisfy only a carefully chosen subset of the constraints (4) so that the expected makespan can still be bounded. Let us first give the LP relaxation.

In the integer program formulation of the above lower bound, we have binary variables y_{ij} to denote the assignment of job j to machine i, and fractional variables $z_i(k)$ to denote the total load on machine i in terms of the deterministic effective sizes β_k. Lemma 5 shows that the following feasibility LP is a valid relaxation:

$$\sum_{i=1}^{m} y_{ij} = 1, \;\; \forall j \in [n], \tag{5}$$

$$z_i(k) - \sum_{j=1}^{n} \beta_k\left(X'_{ij}\right) \cdot y_{ij} = 0, \;\; \forall i \in [m], \; \forall k = 1, 2, \ldots, m, \tag{6}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{E}\left[X''_{ij}\right] \cdot y_{ij} \le 2, \tag{7}$$

$$\sum_{i \in K} z_i(k) \le b \cdot k, \;\; \forall K \subseteq [m] \text{ with } |K| = k, \; \forall k = 1, 2, \ldots, m, \tag{8}$$

$$y_{ij}, z_i(k) \ge 0, \quad \forall i, j, k. \tag{9}$$

In the above LP, $b = O(1)$ denotes the constant multiplying k on the right-hand side of (4).

Although this LP has an exponential number of constraints (because of (8)), we can give an efficient separation oracle. Indeed, consider a candidate solution $(y_{ij}, z_i(k))$ and some integer k. Suppose we want to verify (8) for sets K with $|K| = k$. We just need to look at the k machines with the highest $z_i(k)$ values and check that the sum of $z_i(k)$ for these machines is at most bk. So, using the Ellipsoid method, we can assume that we have an optimal solution (y, z) for this LP in polynomial time. We can summarize this in the following proposition.

Proposition 1 (Lower Bound via LP). *The linear program (5)–(9) can be solved in polynomial time. Moreover, if it is infeasible, then the optimal expected makespan is more than 1.*

3.2. The Rounding

In order to get some intuition about the rounding algorithm, let us first consider the case when the assignment variables y_{ij} are either 0 or 1, that is, the LP solution assigns each job integrally to a machine. In order to bound the expected makespan of this solution, let Z_j denote the variable X_{ij}, where j is assigned to i by this solution.

First consider the exceptional parts Z_j'' of the random variables. Constraint (7) implies that $\sum_j \mathbb{E}[Z_j'']$ is at most 2. Even if the solution assigns all of these jobs to the same machine, the contribution of these jobs to the expected makespan is at most $\sum_j \mathbb{E}[Z_j'']$, and hence at most 2. Thus, we need to worry about only the truncated Z_j' variables.

Now for a machine i and integer $k \in [m]$, let $z_i(k)$ denote the sum of the effective sizes $\beta_k(Z_j')$ for the truncated RVs assigned to i. We can use Lemma 2 to infer that if $z_i(m) = \sum_{j \text{ assigned to } i} \beta_m(Z_j) \le b$, then the probability that these jobs have total size at most $b+2$ is at least $1 - 1/m^2$. Therefore, if $z_i(m) \le b$ for all machines $i \in [m]$, then by a trivial union bound, the probability that makespan is more than $b+2$ is at most $1/m$. Unfortunately, we are not done. All we know from constraint (8) is that the *average* value of $z_i(m)$ is at most b (the average being taken over the m machines). However, there is a clean solution. It follows that there is at least one machine i for which $z_i(m)$ is at most b, and so the expected load on such machines stays $O(1)$ with high probability. Now we can ignore such machines and look at the residual problem. We are left with $k < m$ machines. We recurse on this subproblem (and use the constraint (8) for the remaining set of machines). The overall probability that the load exceeds $O(1)$ on any machine can then be bounded by applying a union bound.

Next, we address the fact that y_{ij} may be not be integral. It seems very difficult to round a fractional solution while respecting all the (exponentially many) constraints in (8). Instead, we observe that the expected makespan analysis (outlined above) utilizes only a linear number of constraints in (8), although this subset is not known a priori. Moreover, for each machine i, the above analysis uses $z_i(k)$ only for a *single* value of k (say, k_i). Therefore, it suffices to find an integral assignment that bounds the load of each machine i in terms of effective sizes β_{k_i}. It turns out that this problem is precisely an instance of the generalized assignment problem (GAP), for which we utilize the algorithm from Shmoys and Tardos [26].

3.2.1. The Rounding Procedure. We now describe the iterative procedure formally. Given an LP solution $\{y_{ij}\}_{i\in[m],j\in[n]}, \{z_i(k)\}_{i,k\in[m]}$; the rounding algorithm is as follows.

Algorithm 1

1. Initialize $\ell \leftarrow m$, $L \leftarrow [m]$, $c_{ij} \leftarrow \mathbb{E}[X_{ij}'']$.
2. While $(\ell > 0)$ do:
   (a) Set $L' \leftarrow \{i \in L : z_i(\ell) \le b\}$. Machines in L' are said to be in *class* ℓ.
   (b) Set $p_{ij} \leftarrow \beta_\ell(X_{ij}')$ for all $i \in L'$ and $j \in [n]$.
   (c) Set $L \leftarrow L \setminus L'$ and $\ell = |L|$.
3. Define a deterministic instance $\mathcal{I}$ of the GAP as follows: the set of jobs and machines remains unchanged. For each job j and machine i, define p_{ij} and c_{ij} as above. The makespan bound is b. Use the algorithm of Shmoys and Tardos [26] to find an assignment of jobs to machines. Output this solution.

Recall that in an instance $\mathcal{I}$ of the GAP, we are given a set of m machines and n jobs. For each job j and machine i, we are given two quantities: p_{ij} is the processing time of j on machine i, and c_{ij} is the cost of assigning j to i. We are also given a makespan bound b. Our goal is to assign jobs to machines to minimize the total cost of assignment, subject to the total processing time of jobs assigned to each machine being at most b. If the natural LP relaxation for this problem has optimal value $C^\star$, then the algorithm in Shmoys and Tardos [26] finds in polynomial time an assignment with cost at most $C^\star$ and makespan at most $b + \max_{i,j} p_{ij}$.

3.3. The Analysis

We begin with some simple observations.

Observation 1. The above rounding procedure terminates in at most m iterations. Furthermore, for any $1 \le \ell \le m$, there are at most ℓ machines of class at most ℓ.

Proof. The first statement follows from the fact that $L' \ne \emptyset$ in each iteration. To see this, consider any iteration involving a set L of ℓ machines. The LP constraint (8) for L implies that $\sum_{i\in L} z_i(\ell) \le b \cdot \ell$, which means there is some $i \in L$ with $z_i(\ell) \le b$, that is, $L' \ne \emptyset$. The second statement follows from the rounding procedure: the machine classes only decrease over the run of the algorithm, and the class assigned to any unclassified machine equals the current number of unclassified machines. □

Observation 2. The solution y is a feasible fractional solution to the natural LP relaxation for the GAP instance $\mathcal{I}$. This solution has makespan at most b and fractional cost at most 2. The rounding algorithm of Shmoys and Tardos [26] yields an assignment with makespan at most $b+1$ and cost at most 2 for the instance $\mathcal{I}$.

Proof. Recall that the natural LP relaxation is the following:

$$\min \sum_{ij} c_{ij} y_{ij}$$

$$\sum_j p_{ij} y_{ij} \le b, \quad \forall i, \tag{10}$$

$$\sum_i y_{ij} = 1, \quad \forall j, \tag{11}$$

$$y_{ij} = 0, \quad \forall i, j \text{ s.t. } p_{ij} > 1, \tag{12}$$

$$y \ge 0.$$

First, note that by (5), y is a valid fractional assignment that assigns each job to one machine, which satisfies (11).

Next we show (10), that is, that $\max_{i=1}^m \sum_{j=1}^n p_{ij} \cdot y_{ij} \le b$. This follows from the definition of the deterministic processing times p_{ij}. Indeed, consider any machine $i \in [m]$. Let ℓ be the class of machine i, and let L be the subset of machines in the iteration when i is assigned class ℓ. This means that $p_{ij} = \beta_\ell(X'_{ij})$ for all $j \in [n]$. Also, because machine $i \in L'$, we have $z_i(\ell) = \sum_{j=1}^n \beta_\ell(X'_{ij}) \cdot y_{ij} \le b$. So we have $\sum_{j=1}^n p_{ij} \cdot y_{ij} \le b$ for each machine $i \in [m]$.

Finally, because the random variable X'_{ij} is at most 1, we get that for any parameter $k \ge 1$, $\beta_k(X'_{ij}) \le 1$; this implies that $p_{max} := \max_{i,j} p_{ij} \le 1$, and hence the constraints (12) are vacuously true. By (7), the objective is $\sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot y_{ij} = \sum_{i=1}^m \sum_{j=1}^n \mathbb{E}[X''_{ij}] \cdot y_{ij} \le 2$. Therefore, the rounding algorithm (Shmoys and Tardos [26]) yields an assignment of makespan at most $b + p_{\max} \le b + 1$, and of cost at most 2. □

In other words, if J_i is the set of jobs assigned to machine i by our algorithm, Observation 2 shows that this assignment has the following properties (let ℓ_i denote the class of machine i):

$$\sum_{i=1}^m \sum_{j \in J_i} \mathbb{E}\left[X''_{ij}\right] \le 2, \qquad \text{and} \tag{13}$$

$$\sum_{j \in J_i} \beta_{\ell_i}\left(X'_{ij}\right) \le b + 1, \qquad \forall i \in [m]. \tag{14}$$

Note that we ideally wanted to give an assignment that satisfied (3) and (4), but instead of giving a bound for all subsets of machines, we bound the β_{ℓ_i} values of the jobs for each machine i. The key Lemma 6 shows that this is enough. First, we need another simple observation:

Observation 3. For any machine i, $\Pr[\sum_{j \in J_i} X'_{ij} > b + 1 + \alpha] \le \ell_i^{-\alpha}$ for all $\alpha \ge 0$.

Proof. Inequality (14) for machine i shows that $\sum_{j \in J_i} \beta_{\ell_i}(X'_{ij}) \le b + 1$. But recalling the definition of the effective size (Definition 1), the result follows from Lemma 2. □

Lemma 6 (Bounding the Makespan)**.** *The expected makespan of the assignment* $\{J_i\}_{i \in [m]}$ *is at most* $3b + 9$.

Proof. Let I^{hi} denote the index set of machines of class 3 or higher. Observation 1 shows that there are at most two machines that are not in I^{hi}. For a machine i, let $T_i = \sum_{j \in J_i} X'_{ij}$ denote the total load due to truncated sizes of jobs assigned to it. Clearly, the makespan is bounded by

$$\max_{i \in I^{\text{hi}}} \{T_i\} + \sum_{i \notin I^{\text{hi}}} T_i + \sum_{i=1}^m \sum_{j \in J_i} X''_{ij}.$$

The expectation of third term is at most two, using (13). We now bound the expectation of the second term above. For any random variable X and $k \ge 1$, using Jensen's inequality (Theorem 5) for the convex function $f(x) = e^{(\log k)x}$, we obtain $\beta_k(X) \ge \mathbb{E}[X]$. Then applying inequality (14) shows that $\mathbb{E}[T_i] \le b + 1$ for any machine i. Therefore, the expected makespan of our solution is at most

$$\mathbb{E}\left[\max_{i \in I^{\text{hi}}} \{T_i\}\right] + 2(b+1) + 2. \tag{15}$$

It remains to bound the first term above. To this end, we will show that

$$\Pr\left[\max_{i\in I^{\text{hi}}}\{T_i\} > b+1+\alpha\right] \le 2^{2-\alpha}/(\alpha-2), \quad \forall \alpha > 2. \tag{16}$$

Using Observation 3 for each $i \in I^{\text{hi}}$ and a union bound, we get

$$\begin{aligned}\Pr\left[\max_{i\in I^{\text{hi}}}\{T_i\} > b+1+\alpha\right] &\le \sum_{\ell=3}^{m}\sum_{i:\ell_i=\ell}\Pr[T_i > b+1+\alpha]\\ &\le \sum_{\ell=3}^{m}\ell^{-\alpha}\cdot(\text{ \# of class } \ell \text{ machines})\\ &\le \sum_{\ell=3}^{m}\ell^{-\alpha+1} \;\le\; \int_{x=2}^{\infty} x^{-\alpha+1}dx \;=\; \frac{2^{-\alpha+2}}{\alpha-2}.\end{aligned}$$

The first inequality uses a trivial union bound, the second inequality uses Observation 3 above, and the third inequality is by Observation 1. This proves (16). Finally, using (16), we get

$$\mathbb{E}\left[\max_{i\in I^{\text{hi}}}\{T_i\}\right] = (b+4) + \int_{\alpha=3}^{\infty}\Pr\left[\max_{i\in I^{\text{hi}}}\{T_i\} > b+1+\alpha\right]d\alpha \;\le\; (b+4) + \int_{\alpha=3}^{\infty} 2^{2-\alpha}d\alpha \;\le\; b+5.$$

Inequality (15) now shows that the expected makespan is at most $(b+5)+2(b+1)+2$. □

This completes the proof of Theorem 1.

3.3.1. Explicit Approximation Ratio. Recall that our algorithm runs within a binary search scheme, where each iteration involves a bound M and the algorithm either finds a solution of expected makespan $O(M)$ or finds a certificate that the optimal makespan is at least M. We now calculate the constant explicitly. We assume that the optimal makespan is at most M and proceed as follows. By scaling all RVs by $\frac{1}{2}(1-\frac{1}{e})\frac{1}{M}$ and applying Lemmas 4 and 5, it follows that the linear program (5)–(9) is feasible with $b = 17$. (If we find that this LP is infeasible, it gives a proof that the original optimal value is at least M.) Then, by Lemma 6, our rounding produces a solution of expected makespan at most $3b+9$ (after the scaling). For any constant $\epsilon > 0$, we can ensure that the value M at the end of the binary search is within a $1+\epsilon$ factor of the optimal value. So the overall approximation ratio is $\frac{2e}{e-1}(3b+9)(1+\epsilon) \le 190$. Although we have not tried to optimize the constant via our approach, we think getting a constant factor close to 2 (which is known for the deterministic problem) would require new ideas.

4. Budgeted Makespan Minimization

We now consider a generalization of the problem StocMakespan, called BudgetStocMakespan, where each job j also has reward $r_j \ge 0$. We are required to schedule some subset of jobs whose total reward is at least some target value R. The objective, again, is to minimize the expected makespan. If the target $R = \sum_{j\in[n]} r_j$, then we recover the problem StocMakespan. We show Theorem 2: *There is an $O(1)$-approximation algorithm for the budgeted makespan minimization problem on unrelated machines.*

Naturally, our algorithm/analysis will build on the ideas developed in Section 3, but we will need some new ideas to handle the fact that only a subset of jobs need to be scheduled. As in the case of StocMakespan, we can formulate a suitable LP relaxation. A similar rounding procedure reduces the stochastic problem to a deterministic problem, which we call Budgeted GAP. An instance of Budgeted GAP is similar to an instance of the GAP, besides the additional requirement that jobs have rewards and we are required to assign jobs with a total reward of at least some target R. Rounding the natural LP relaxation for Budgeted GAP turns out to be nontrivial. Indeed, using ideas from Shmoys and Tardos [26], we reduce this rounding problem to rounding a fractional matching solution with additional constraints, and solve the latter using polyhedral properties of bipartite-matching polyhedra.

As before, using a binary search scheme (and by scaling down the sizes), we can assume that we need to either (i) find a solution of expected makespan $O(1)$ or (ii) prove that the optimal value is more than 1. We use a natural LP relaxation that has variables y_{ij} for each job j and machine i. The LP includes the constraints (6)–(9) for the base problem, and in addition it has the following two constraints:

$$\sum_{i=1}^{m} y_{ij} \le 1, \qquad \forall j = 1, \ldots, n, \tag{17}$$

$$\sum_{j=1}^{n} r_j \cdot \sum_{i=1}^{m} y_{ij} \ge R. \tag{18}$$

The first constraint, (17), replaces constraint (5) and says that not all jobs need to be assigned. The second constraint, (18), ensures that the assigned jobs have total reward at least the target R. For technical reasons that will be clear later, we also perform a preprocessing step: for i, j pairs where $\mathbb{E}[X''_{ij}] > 2$, we force the associated y_{ij} variable to zero. Note that by Lemma 5, this variable fixing is valid for any integral assignment that has expected makespan at most one (in fact, we have $\sum_i \sum_j \mathbb{E}[X''_{ij}] \cdot y_{ij} \le 2$ for such an assignment). As in Section 3.1, this LP can be solved in polynomial time via the ellipsoid method. If the LP is infeasible, we get a proof (using Lemma 5) that the optimal expected makespan is more than one. Hence, we assume the LP is feasible and proceed to round the solution along the lines of Section 3.2.

Recall that the rounding algorithm in Section 3.2 reduces the fractional LP solution to an instance of the GAP. Here, we will use a further generalization of the GAP, which we call Budgeted GAP. An instance of this problem is similar to an instance of the GAP. We are given m machines and n jobs, and for each job j and machine i, we are given the processing time p_{ij} and the associated assignment cost c_{ij}. Now each job j has a reward r_j, and there are two target parameters: the reward target R and the makespan target B. We let p_{max} and c_{max} denote the maximum values of processing time and cost, respectively. A solution must assign a subset of jobs to machines such that the total reward of assigned jobs is at least R. Moreover, as in the case of the GAP, the goal is to minimize the total assignment cost subject to the condition that the makespan is at most B. Our main technical theorem of this section shows how to round an LP relaxation of this Budgeted GAP problem.

Theorem 7. *There is a polynomial-time rounding algorithm for* Budgeted GAP *that, given any fractional solution to the natural LP relaxation of cost* C^*, *produces an integer solution having total cost at most* $C^* + c_{max}$ *and makespan at most* $B + 2p_{max}$.

Before we prove this theorem, let us use it to solve BudgetStocMakespan and prove Theorem 2. Proceeding as in Section 3.2, we perform Steps 1 and 2 from the rounding procedure. This rounding gives us values p_{ij} and c_{ij} for each job/machine pair. Now, instead of reducing to an instance of the GAP, we reduce to an instance $\mathcal{I}'$ of Budgeted GAP. The instance $\mathcal{I}'$ has the same set of jobs and machines as in the original BudgetStocMakespan instance $\mathcal{I}$. For each job j and machine i, the processing time and the assignment cost are given by p_{ij} and c_{ij}, respectively. Furthermore, the reward r_j for job j and the reward target R are the same as those in $\mathcal{I}$. The makespan bound $B = b = O(1)$ (as in (8)). It is easy to check that the fractional solution y_{ij} is a feasible fractional solution to the natural LP relaxation for $\mathcal{I}'$ (given below), and the assignment cost of this fractional solution is at most 2. Applying Theorem 7 yields an assignment $\{J_i\}_{i=1}^{m}$, which has the following properties:

- The makespan is at most $b + 2 = O(1)$, that is, $\sum_{j \in J_i} p_{ij} \le b + 2p_{max} \le b + 2$ for each machine i. Here we use the fact that $p_{max} \le 1$.
- The cost of the solution, $\sum_{i=1}^{m} \sum_{j \in J_i} c_{ij}$, is at most 4. This uses the fact that the LP cost $C^* = \sum c_{ij} \cdot y_{ij} \le 2$ and $c_{max} \le 2$ by the preprocessing on the $\mathbb{E}[X''_{ij}]$ values.
- The total reward for the assigned jobs, $\sum_{j \in \cup_i J_i} r_j$, is at least R.

Now, arguing exactly as in Lemma 6, the first two properties imply that the expected makespan is at most $3b + 14$ (instead of $3b + 9$). The third property implies that the total reward of assigned jobs is at least R and completes the proof of Theorem 2.

Remark. Based on the same arguments as at the end of Section 3.3, we can calculate an explicit constant factor of $\frac{2e}{e-1}(3b + 14)(1 + \epsilon) \le 206$.

Proof of Theorem 7. Let $\mathcal{I}$ be an instance of Budgeted GAP as described above. The natural LP relaxation for this problem is as follows:

$$\min \sum_{ij} c_{ij} y_{ij}$$

$$\sum_j p_{ij} y_{ij} \leq B, \quad \forall i, \tag{19}$$

$$\sum_i y_{ij} \leq 1, \quad \forall j, \tag{20}$$

$$\sum_{i,j} y_{ij} r_j \geq R, \tag{21}$$

$$y_{ij} = 0, \quad \forall i, j \text{ s.t. } p_{ij} > b, \tag{22}$$

$$y \geq 0.$$

Let $\{y_{ij}\}$ denote an optimal fractional solution to this LP. For each machine i, let $t_i := \lceil \sum_j y_{ij} \rceil$ be the (rounded) fractional assignment to machine i. Using the algorithm in theorem 2.1 of Shmoys and Tardos [26], we obtain a bipartite graph $G = (V_1 \cup V_2, E)$ and a fractional matching y' in G, where we have the following:

- $V_1 = [n]$ is the set of jobs, and V_2 (indexed by $i' = 1, \ldots, m'$) consists of t_i copies for each machine $i \in [m]$. The cost $c_{i'j} = c_{ij}$ for any job $j \in [n]$ and any machine copy i' of machine $i \in [m]$.
- For each job $j \in [n]$, we have $\sum_{i=1}^{m'} y'_{ij} = \sum_{i=1}^{m} y_{ij} \leq 1$ for all $j \in [n]$.
- The reward $\sum_{j=1}^{n} r_j \sum_{i=1}^{m'} y'_{ij} \geq R$ and the cost $\sum_{i'=1}^{m'} \sum_{j=1}^{n} c_{i'j} y'_{ij} = C^*$ are the same as for y.
- The jobs of V_1 incident to copies of any machine $i \in [m]$ can be divided into (possibly overlapping) groups $H_{i,1}, \ldots, H_{i,t_i}$, where

$$\sum_{j \in H_{i,g}} y'_{ij} = 1 \text{ for all } 1 \leq g \leq t_i - 1 \quad \text{and} \quad \sum_{j \in H_{i,t_i}} y'_{ij} \leq 1,$$

and for any two consecutive groups $H_{i,g}$ and $H_{i,g+1}$, we have $p_{ij} \geq p_{ij'}$ for all $j \in H_{i,g}$ and $j' \in H_{i,g+1}$. Informally, this is achieved by sorting the jobs in nonincreasing order of p_{ij} and assigning the kth unit of $\sum_j y_{ij}$ to the kth machine copy for each $1 \leq k \leq t_i$.

A crucial property of this construction shown in Shmoys and Tardos [26] is that any assignment that places at most one job on each machine copy has makespan at most $B + p_{max}$ in the original instance $\mathcal{I}$ (where for every machine i, we assign to it all the jobs that are assigned to a copy of i in this integral assignment). We will use the following simple extension of this property: if the assignment places *two* jobs on one machine copy and at most one job on all other machine copies, then it has makespan at most $B + 2p_{max}$ in the instance $\mathcal{I}$.

Observe that the solution y' is a feasible solution to the following LP with variables $\{z_{ij}\}_{(i,j) \in E}$:

$$\min \sum_{ij} c_{ij} z_{ij} \tag{23}$$

$$\sum_{i \in [m']:(ij) \in E} z_{ij} \leq 1, \quad \forall j \in [n], \tag{24}$$

$$\sum_{j \in [n]:(ij) \in E} z_{ij} \leq 1, \quad \forall i \in [m'], \tag{25}$$

$$\sum_{(ij) \in E} r_j \cdot z_{ij} \geq R, \tag{26}$$

$$z \geq 0. \tag{27}$$

So the optimal value of this auxiliary LP is at most C^*. We note that its integrality gap is unbounded even when c_{max} is small; see the example below. So this differs from Shmoys and Tardos [26] for the usual GAP where the corresponding LP (without (26)) is actually integral. However, we show below how to obtain a good integral solution that violates the matching constraint for just a *single* machine copy in V_2.

Indeed, let $\mathbf{z}$ be an optimal extreme point solution to this LP, so $\mathbf{c}^\mathrm{T}\mathbf{z} \leq C^*$. Note that the feasible region of this LP is just the bipartite-matching polytope on G intersected with one extra linear constraint (26) that corresponds to the total reward being at least R. So $\mathbf{z}$ must be a convex combination of two adjacent extreme points of the bipartite-matching polytope. Using the integrality and adjacency properties (see Balinski and Russakoff [2]) of the bipartite-matching polytope, it follows that $\mathbf{z} = \lambda_1 \cdot \mathbf{1}_{M_1} + \lambda_2 \cdot \mathbf{1}_{M_2}$, where

- $\lambda_1 + \lambda_2 = 1$ and $\lambda_1, \lambda_2 \geq 0$,
- M_1 and M_2 are integral matchings in G, and
- the symmetric difference $M_1 \oplus M_2$ is a single cycle or path.

For any matching M, let $c(M)$ and $r(M)$ denote its total cost and reward, respectively. Without loss of generality, we assume that $r(M_1) \geq r(M_2)$. If $c(M_1) \leq c(M_2)$, then M_1 is itself a solution with reward at least R and cost at most C^*. So we assume $c(M_1) > c(M_2)$ below.

If $M_1 \oplus M_2$ is a cycle, then we output M_2 as the solution. Note that the cycle must be an even cycle, so the sets of jobs assigned by M_1 and M_2 are identical. As the reward function is dependent only on the assigned jobs (and not the machines used in the assignment), it follows that $r(M_2) = r(M_1) \geq R$. So M_2 is indeed a feasible solution and has cost $c(M_2) \leq \mathbf{c}^{\mathrm{T}}\mathbf{z} \leq C^*$.

Now consider the case where $M_1 \oplus M_2$ is a path. If the sets of jobs assigned by M_1 and M_2 are the same, then M_2 is an optimal integral solution (as above). The only remaining case is that M_1 assigns one additional job (say, j^* to i^*) over the jobs in M_2. Then we return the solution $M_2 \cup \{(j^*, i^*)\}$. Note that this is not a feasible matching. But the only infeasibility is at machine copy i^*, which may have two jobs assigned; all other machine copies have at most one job. The reward of this solution is $r(M_1) \geq R$. Moreover, its cost is at most $c(M_2) + c_{i^* j^*} \leq C^* + c_{max}$.

Now, using this (near-feasible) assignment gives us the desired cost and makespan bounds and completes the proof of Theorem 7. □

4.2. Integrality Gap for Budgeted Matching LP

Here we show that the LP (23)–(27) used in the algorithm for Budgeted GAP has an unbounded integrality gap, even if we assume that $c_{max} \ll OPT$. The instance consists of n jobs and $m = n - 1$ machines. For each machine $i \in [m]$, there are two incident edges in E: one to job i (with cost 1) and the other to job $i + 1$ (with cost n). So E is the disjoint union of two machine perfect matchings M_1 (of total cost m) and M_2 (of total cost mn). The rewards are

$$r_j = \begin{cases} 1 & \text{if } j = 1, \\ 4 & \text{if } 2 \leq j \leq n - 1, \\ 2 & \text{if } j = n, \end{cases}$$

and the target is $R = 4(n - 2) + 1 + \epsilon$, where $\epsilon \to 0$. Note that the only (minimal) integral solution involves assigning the jobs $\{2, 3, \ldots, n\}$, which has total reward $4(n - 2) + 2$. This solution has cost $OPT = mn$ and corresponds to matching M_2. On the other hand, consider the fractional solution $z = \epsilon \mathbf{1}_{M_2} + (1 - \epsilon)\mathbf{1}_{M_1}$. This is clearly feasible for the matching constraints, and its reward is $\epsilon(4(n - 2) + 2) + (1 - \epsilon)(4(n - 2) + 1) = R$. So z is a feasible fractional solution. The cost of this fractional solution is at most $m + \epsilon(mn) \ll OPT$.

5. ℓ_q-Norm Objectives

In this section, we consider the stochastic load-balancing problem with q-norm objectives. Given an assignment of jobs to machines $\{J_i\}_{i=1}^m$, the load L_i on machine i is the RV $L_i := \sum_{j \in J_i} X_{ij}$. Our goal is to find an assignment to minimize the expected q-norm of the load vector $\mathbf{L} := (L_1, L_2, \ldots, L_m)$. Recall that the makespan is $\|\mathbf{L}\|_\infty$, which is approximated within constant factors by $\|\mathbf{L}\|_{\log m}$. So the q-norm problem is a generalization of StocMakespan. Our main result here is Theorem 3: *There is an $O(\frac{q}{\log q})$-approximation algorithm for the stochastic q-norm minimization problem on unrelated machines.*

We begin by assuming that we know the optimal value M of the q-norm. Our approach parallels that for the case of minimizing the expected makespan, with some changes. In particular, the main steps are as follows: (i) find valid inequalities satisfied by any assignment for which $\mathbb{E}[\|\mathbf{L}\|_q] \leq M$, (ii) reduce the problem to a deterministic assignment problem for which any feasible solution satisfies the valid inequalities above, (iii) solve the deterministic problem by writing a convex programming relaxation and give a rounding procedure for a fractional solution to this convex program, and (iv) prove that the resulting assignment of jobs to machines has a small expected q-norm of the load vector.

5.1. Useful Bounds

We start with stating some valid inequalities satisfied by any assignment $\{J_i\}_{i=1}^m$. For each $j \in [n]$, define $Y_j = X_{ij}$, where $j \in J_i$. By definition of M, we know that

$$\mathbb{E}\left[\left(\sum_{i=1}^m \left(\sum_{j\in J_i} Y_j\right)^q\right)^{1/q}\right] \le M. \tag{28}$$

As in Section 3, we split each random variable Y_j into two parts: truncated, $Y'_j = Y_j \cdot \mathbf{I}_{Y_j \le M}$, and exceptional, $Y''_j = Y_j \cdot \mathbf{I}_{Y_j > M}$. The lemma below is analogous to (3) and states that the total expected size of the exceptional parts cannot be too large.

Lemma 7. *For any schedule satisfying* (28), *we have* $\sum_{j=1}^n \mathbb{E}[Y''_j] \le 2M$.

Proof. Suppose for a contradiction that $\sum_{j=1}^n \mathbb{E}[Y''_j] > 2M$. Lemma 1 implies $\mathbb{E}[\max_{j=1}^n Y''_j] > M$. Now, using the monotonicity of norms and the fact that $Y''_j \le Y_j$, we have

$$\max_{j=1}^n Y''_j \le \|(Y''_1, \dots, Y''_n)\|_q \le \left(\sum_{i=1}^m \left(\sum_{j\in J_i} Y_j\right)^q\right)^{1/q},$$

which contradicts (28). □

Our next two bounds deal with the truncated RVs Y'_j. The first one states that if we replace Y'_j by its expectation $\mathbb{E}[Y'_j]$, the q-norm of this load vector of expectations cannot exceed M. The second bound states that the expected qth moment of the vector $(Y'_j)_{j=1}^n$ is bounded by a constant times M^q.

Lemma 8. *For any schedule satisfying* (28), *we have*

$$\sum_{i=1}^m \left(\sum_{j\in J_i} \mathbb{E}\left[Y'_j\right]\right)^q \le M^q.$$

Proof. Because the function

$$f(Y'_1, \dots, Y'_n) := \left(\sum_{i=1}^m \left(\sum_{j\in J_i} Y'_j\right)^q\right)^{1/q}$$

is a norm and hence convex, Jensen's inequality (Theorem 5) implies $\mathbb{E}[f(Y'_1, \dots, Y'_n)] \ge f(\mathbb{E}[Y'_1], \dots, \mathbb{E}[Y'_n])$. Raising both sides to the qth power and using (28), the lemma follows. □

Lemma 9. *Let* $\alpha = 2^{q+1} + 8$. *For any schedule satisfying* (28), *we have*

$$\sum_{j=1}^n \mathbb{E}\left[\left(Y'_j\right)^q\right] \le \alpha \cdot M^q.$$

Proof. Define $Z := \sum_{j=1}^n (Y'_j)^q$ as the quantity of interest. Observe that it is the sum of independent $[0, M^q]$ bounded random variables. Because $q \ge 1$ and the RVs are nonnegative, $Z \le \sum_{i=1}^m (\sum_{j\in J_i} Y'_j)^q$. Thus, (28) implies $\mathbb{E}[Z^{1/q}] \le M$. However, now Jensen's inequality cannot help upper bound $\mathbb{E}[Z]$.

Instead, we use a second-moment calculation. To reduce notation, let $Z_j := (Y'_j)^q$, so $Z = \sum_{j=1}^n Z_j$. The variance of Z is $\mathrm{var}(Z) = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 \le \sum_{j=1}^n \mathbb{E}[Z_j^2] \le M^q \cdot \mathbb{E}[Z]$, as each Z_j is $[0, M^q]$ bounded. By Chebyshev's inequality,

$$\Pr\left[Z < \frac{\mathbb{E}[Z]}{2} - 4M^q\right] \le \frac{\mathrm{var}(Z)}{(\mathbb{E}[Z]/2 + 4M^q)^2} \le \frac{\mathrm{var}(Z)}{(\mathbb{E}[Z]/2) \cdot 4M^q} \le \frac{2M^q \cdot E[Z]}{\mathbb{E}[Z] \cdot 4M^q} \le \frac{1}{2}.$$

This implies

$$E[Z^{1/q}] \ge \frac{1}{2}\left(\frac{\mathbb{E}[Z]}{2} - 4M^q\right)^{1/q}.$$

Using the bound $\mathbb{E}[Z^{1/q}] \le M$ from above, we now obtain $\mathbb{E}[Z] \le 2 \cdot ((2M)^q + 4M^q)$ as desired. ◻

In the next subsections, we show that the three bounds above are enough to get a meaningful lower bound on the optimal q-norm of loads.

5.2. Reduction to a Deterministic Scheduling Problem

We now formulate a surrogate deterministic scheduling problem, which we call q-DetSched. An instance of this problem has n jobs and m machines. For each job j and machine i, there is a processing time p_{ij} and two costs c_{ij} and d_{ij}. There are also bounds C and D on the two cost functions, respectively. The goal is to find an assignment of jobs to machines that minimizes the q-norm of the machine loads subject to the constraint that the total c-cost and d-cost of the assignments are at most C and D, respectively. We now show how to convert an instance $\mathcal{I}_{stoc}$ of the (stochastic) expected q-norm minimization problem to an instance $\mathcal{I}_{det}$ of the (deterministic) problem q-DetSched.

Suppose $\mathcal{I}_{stoc}$ has m machines and n jobs, with random variables X_{ij} for each machine i and job j. As before, let $X'_{ij} = X_{ij} \cdot \mathbf{I}_{X_{ij} \le M}$ and $X''_{ij} = X_{ij} \cdot \mathbf{I}_{X_{ij} > M}$ denote the truncated and exceptional parts of each random variable X_{ij}, respectively. Then, instance $\mathcal{I}_{det}$ has the same set of jobs and machines as those in $\mathcal{I}$. Furthermore, define

- the processing time $p_{ij} = \mathbb{E}[X'_{ij}]$,
- the c-cost $c_{ij} = \mathbb{E}[X''_{ij}]$ with bound $C = 2M$, and
- the d-cost $d_{ij} = \mathbb{E}[(X'_{ij})^q]$ with bound $D = \alpha \cdot M^q$, where $\alpha = 2^{q+1} + 8$.

Observation 4. If there is any schedule of expected q-norm at most M in the instance $\mathcal{I}_{stoc}$, then the optimal value of the instance $\mathcal{I}_{det}$ is at most M.

Proof. The proof follows directly from Lemmas 7, 8, and 9. ◻

5.3. Approximation Algorithm for q-DetSched

Our approximation algorithm for q-DetSched is closely based on the algorithm for unrelated machine scheduling to minimize ℓ_q-norms (Azar and Epstein [1]). We show the following:

Theorem 8. *There is a polynomial-time algorithm that, given any instance $\mathcal{I}_{det}$ of q-DetSched, finds a schedule with* (i) *q-norm of processing times at most $2^{1+2/q} \cdot OPT(\mathcal{I}_{det})$*, (ii) *$c$-cost at most $3C$, and* (iii) *d-cost at most $3D$.*

Proof. We provide only a sketch, as many of these ideas parallel those from Azar and Epstein [1]. Start with a convex programming "relaxation" with variables x_{ij} (for assigning job j to machine i):

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} \ell_i^q + \sum_{ij} p_{ij}^q \cdot x_{ij} \\
\text{s.t.} \quad & \ell_i = \sum_j p_{ij} \cdot x_{ij}, \quad \forall i, \\
& \sum_i x_{ij} = 1, \quad \forall j, \\
& \sum_{ij} c_{ij} \cdot x_{ij} \le C, \\
& \sum_{ij} d_{ij} \cdot x_{ij} \le D, \\
& x \ge 0.
\end{aligned}
$$

This convex program can be solved to arbitrary accuracy, and its optimal value is $V \le 2 \cdot OPT(\mathcal{I}_{det})^q$. Let (x, ℓ) denote the optimal fractional solution below.

We now further reduce this q-norm problem to the generalized assignment problem. The GAP instance $\mathcal{I}_{gap}$ has the same set of jobs and machines as those in $\mathcal{I}_{det}$. For a job j and machine i, the processing time remains p_{ij}. However, the cost of assigning j to i is now $\gamma_{ij} := \frac{c_{ij}}{C} + \frac{d_{ij}}{D} + \frac{p_{ij}^q}{V}$. Furthermore, we impose a bound of ℓ_i on the total processing time of jobs assigned to each machine i (i.e., the makespan on i is constrained to be at most ℓ_i). Note that the solution x to the convex program is also a feasible fractional solution to the natural LP relaxation for the GAP with an objective function value of $\sum_{ij} \gamma_{ij} \cdot x_{ij} \le 3$. The rounding algorithm in Shmoys and Tardos [26] can now be used to round x into an integral assignment $\{A_{ij}\}$ with γ-cost also at most 3, and load on each machine i being $L_i \le \ell_i + m_i$, where m_i denotes the maximum processing time of any job assigned to machine i

by this algorithm. The definition of γ and the bound on the γ-cost imply that the c-cost and d-cost of this assignment are at most $3C$ and $3D$, respectively. To bound the q-norm of processing times,

$$\sum_{i=1}^{m} L_i^q \le 2^{q-1}\left(\sum_i \ell_i^q + \sum_i m_i^q\right) \le 2^{q-1}\left(V + \sum_{ij} p_{ij}^q \cdot A_{ij}\right) \le 2^{q-1}(V + 3V) = 2^{q+1}\cdot V.$$

Above, the first inequality uses $(a+b)^q \le 2^{q-1}(a^q + b^q)$, and the third inequality uses the fact that $\sum_{i,j} p_{ij}^q A_{ij} \le V\cdot \sum_{i,j}\gamma_{ij}A_{ij} \le 3V$ by the bound on the γ-cost. The proof is now completed by using $V \le 2\cdot OPT(\mathcal{I}_{det})^q$. □

5.4. Interpreting the Rounded Solution

Starting from an instance $\mathcal{I}_{stoc}$ of the expected q-norm minimization problem, we first constructed an instance $\mathcal{I}_{det}$ of q-DetSched. Let $\mathcal{J} = (J_1,\ldots,J_m)$ denote the solution found by applying Theorem 8 to the instance $\mathcal{I}_{det}$. If the q-norm of processing times of this assignment (as a solution for $\mathcal{I}_{det}$) is more than $2^{1+2/q}M$, then, using Observation 4 and Theorem 8, we obtain a certificate that the optimal value of $\mathcal{I}_{stoc}$ is more than M. So we assume that $\mathcal{J}$ has objective at most $2^{1+2/q}M$ (as a solution to $\mathcal{I}_{det}$). We use exactly this assignment as a solution for the stochastic problem as well. It remains to bound the expected q-norm of this assignment.

By the reduction from $\mathcal{I}_{stoc}$ to $\mathcal{I}_{det}$, and the statement of Theorem 8, we know that

$$\sum_{i=1}^{m}\sum_{j\in J_i}\mathbb{E}\left[X''_{ij}\right] = \sum_{i=1}^{m}\sum_{j\in J_i} c_{ij} \le 6M, \tag{29}$$

$$\sum_{i=1}^{m}\left(\sum_{j\in J_i}\mathbb{E}\left[X'_{ij}\right]\right)^q = \sum_{i=1}^{m}\left(\sum_{j\in J_i} p_{ij}\right)^q \le 2^{q+2}\cdot M^q, \tag{30}$$

$$\sum_{i=1}^{m}\sum_{j\in J_i}\mathbb{E}\left[\left(X'_{ij}\right)^q\right] = \sum_{i=1}^{m}\sum_{j\in J_i} d_{ij} \le 3\alpha M^q. \tag{31}$$

We now derive properties of this assignment as a solution for $\mathcal{I}_{stoc}$.

Lemma 10. *The expected q-norm of exceptional jobs* $\mathbb{E}[(\sum_{i=1}^m(\sum_{j\in J_i} X''_{ij})^q)^{1/q}] \le 6M$.

Proof. This follows from (29), because the ℓ_q-norm of a vector is at most its ℓ_1-norm. □

Lemma 11. *The expected q-norm of truncated jobs* $\mathbb{E}[(\sum_{i=1}^m(\sum_{j\in J_i} X'_{ij})^q)^{1/q}] \le O(\frac{q}{\log q})M$.

Proof. Define random variables $Q_i := (\sum_{j\in J_i} X'_{ij})^q$, so that the q-norm of the loads is

$$Q := \left(\sum_{i=1}^{m} Q_i\right)^{1/q} = \left(\sum_{i=1}^{m}\left(\sum_{j\in J_i} X'_{ij}\right)^q\right)^{1/q}.$$

Because $f(Q_1,\ldots,Q_m) = (\sum_{i=1}^m Q_i)^{1/q}$ is a concave function for $q\ge 1$, using Jensen's inequality (Theorem 5) on $-f(Q_1,\ldots,Q_m)$, which is a convex function,

$$\mathbb{E}[Q] \le \left(\sum_{i=1}^{m}\mathbb{E}[Q_i]\right)^{1/q}. \tag{32}$$

We can bound each $\mathbb{E}[Q_i]$ separately using Rosenthal's inequality (Theorem 6):

$$\mathbb{E}[Q_i] = \mathbb{E}\left[\left(\sum_{j\in J_i} X'_{ij}\right)^q\right] \le K^q\cdot\left(\left(\sum_{j\in J_i}\mathbb{E}\left[X'_{ij}\right]\right)^q + \sum_{j\in J_i}\mathbb{E}\left[\left(X'_{ij}\right)^q\right]\right),$$

where $K = O(q/\log q)$. Summing this over all $i = 1,\ldots,m$ and using (30) and (31), we get

$$\sum_{i=1}^{m}\mathbb{E}[Q_i] \le K^q\cdot(2^{q+2} + 3\alpha)M^q. \tag{33}$$

Recall that $\alpha = 2^{q+1}+8$. Now, plugging this into (32), we obtain $\mathbb{E}[Q] \le O(K)\cdot M$. □

Finally, using Lemmas 10 and 11 and the triangle inequality, the expected q-norm of solution $\mathcal{J}$ is $O(\frac{q}{\log q}) \cdot M$, which completes the proof of Theorem 3.

5.4.1. Explicit Approximation Ratio. Here we state our approximation ratios explicitly. By Equations (32) and (33), the expected q-norm of truncated jobs is

$$\mathbb{E}[Q] \leq \left(K^q \cdot \left(2^{q+2} + 3\alpha\right)M^q\right)^{1/q} = \left(2^{q+2} + 3\left(2^{q+1} + 8\right)\right)\right)^{1/q} \cdot KM = 2\left(10 + 3 \cdot 2^{3-q}\right)^{1/q} \cdot KM.$$

And the expected q-norm of exceptional jobs is at most $6M$ by Lemma 10. By the triangle inequality, the expected q-norm of solution $\mathcal{J}$ is at most $(6 + 2(10 + 3 \cdot 2^{3-q})^{1/q} \cdot K)M$. Note that for any constant $\epsilon > 0$, we can ensure that M is within a $1 + \epsilon$ factor of the optimal value (by the binary search approach). Hence, the overall approximation ratio for the q-norm is $(6 + 2(10 + 3 \cdot 2^{3-q})^{1/q}K)(1 + \epsilon)$, for any $\epsilon > 0$, where K is the parameter in Theorem 6 (Rosenthal's inequality).

The following known result provides a bound on the parameter K^q in Theorem 6.

Theorem 9 (Ibragimov and Sharakhmetov [10]). *Let Z denote a random variable with Poisson distribution with parameter* 1, *that is,* $\Pr[Z = k] = e^{-1}/k!$ *for integer $k \geq 0$. The parameter of Theorem 6 (Rosenthal inequality) is* $K = (\mathbb{E}Z^q)^{1/q}$.

We note that Theorem 6 holds for any set of RVs, and only the constant K depends on the Poisson rate 1 random variable Z.

Example 1. For the ℓ_2-norm, $K^2 \leq \mathbb{E}Z^2 = 2 \Rightarrow K \leq \sqrt{2}$. The overall approximation ratio is $(6 + (10 + 3 \cdot 2^1)^{1/2} 2\sqrt{2})(1 + \epsilon) = (6 + 8\sqrt{2})(1 + \epsilon) \approx 17.31(1 + \epsilon)$. For the ℓ_3-norm, $K^3 \leq 5 \Rightarrow K \leq \sqrt[3]{5}$, and the approximation ratio is $(6 + (10 + 3)^{1/3} 2 \sqrt[3]{5}) \cdot (1 + \epsilon) \approx 14.04(1 + \epsilon)$.

6. Conclusion

We obtained the first constant-factor approximation algorithm for stochastic makespan minimization on unrelated machines. We also extended this result to a budgeted version of the problem. Finally, we considered the stochastic problem with q-norm objectives and obtained a constant approximation for fixed q. An interesting open problem is to extend these techniques to deal with other stochastic discrete optimization problems with makespan objectives.

Acknowledgments

A preliminary version of this paper appeared as "Stochastic load balancing on unrelated machines" in the *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms* (pp. 1274–1285), 2018. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

[1] Azar Y, Epstein A (2005) Convex programming for scheduling unrelated parallel machines. *Proc. 37th Annual ACM Sympos. Theory Comput.* (ACM, New York), 331–337.
[2] Balinski ML, Russakoff A (1974) On the assignment polytope. *SIAM Rev.* 16(4):516–525.
[3] Elwalid AI, Mitra D (1993) Effective bandwidth of general Markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Networking* 1(3):329–343.
[4] Goel A, Indyk P (1999) Stochastic load balancing and related problems. *Proc. 40th Annual Sympos. Foundations Comput.* (IEEE, Washington, DC), 579–586.
[5] Graham RL (1966) Bounds for certain multiprocessing anomalies. *Bell System Tech. J.* 45(9):1563–1581.
[6] Gupta V, Moseley B, Uetz M, Xie Q (2017) Stochastic online scheduling on unrelated machines. Eisenbrand F, Koenemann J, eds. *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 10328 (Springer International Publishing, Cham, Switzerland), 228–240.
[7] Hochbaum DS, Shmoys DB (1987) Using dual approximation algorithms for scheduling problems theoretical and practical results. *J. ACM* 34(1):144–162.
[8] Hochbaum DS, Shmoys DB (1988) A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM J. Comput.* 17(3):539–551.
[9] Hui JY (1988) Resource allocation for broadband networks. *IEEE J. Selected Areas Comm.* 6(3):1598–1608.
[10] Ibragimov R, Sharakhmetov S (2001) The best constant in the Rosenthal inequality for nonnegative random variables. *Statist. Probab. Lett.* 55(4):367–376.
[11] Im S, Moseley B, Pruhs K (2015) Stochastic scheduling of heavy-tailed jobs. *32nd Internat. Sympos. Theoret. Aspects Comput. Sci.*, Leibniz International Proceedings in Informatics, vol. 30 (Schloss Dagstuhl–Leibniz Center for Informatics, Wadern, Germany), 474–486.
[12] Johnson WB, Schechtman G, Zinn J (1985) Best constants in moment inequalities for linear combinations of independent and exchangeable random variables. *Ann. Probab.* 13(1):234–253.
[13] Kelly FP (1996) Notes on effective bandwidths. Kelly FP, Zachary S, Ziedins IB, eds. *Stochastic Networks: Theory and Applications* (Oxford University Press, Oxford, UK), 141–168.

[14] Kim JH, Vu VH (2000) Concentration of multivariate polynomials and its applications. *Combinatorica* 20(3):417–434.
[15] Kleinberg J, Rabani Y, Tardos E (2000) Allocating bandwidth for bursty connections. *SIAM J. Comput.* 30(1):191–217.
[16] Latała R (1997) Estimation of moments of sums of independent real random variables. *Ann. Probab.* 25(3):1502–1513.
[17] Lenstra JK, Shmoys DB, Tardos É (1990) Approximation algorithms for scheduling unrelated parallel machines. *Math. Programming* 46(1–3):259–271.
[18] Li J, Deshpande A (2011) Maximizing expected utility for stochastic combinatorial optimization problems. *Proc. IEEE 52nd Annual Sympos. Foundations Comput. Sci.* (IEEE Computer Society, Washington, DC), 797–806.
[19] Li J, Yuan W (2013) Stochastic combinatorial optimization via Poisson approximation. *Proc. 45th Annual ACM Sympos. Theory Comput.* (ACM, New York), 971–980.
[20] Megow N, Uetz M, Vredeveld T (2006) Models and algorithms for stochastic online scheduling. *Math. Oper. Res.* 31(3):513–525.
[21] Möhring RH, Schulz AS, Uetz M (1999) Approximation in stochastic scheduling: the power of LP-based priority policies. *J. ACM* 46(6):924–942.
[22] Molinaro M (2019) Stochastic l_p load balancing and moment problems via the *L*-function method. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 343–354.
[23] Pinedo M (2004) Offline deterministic scheduling, stochastic scheduling, and online deterministic scheduling. Leung JYT, ed. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (Chapman and Hall/CRC, Boca Raton, FL).
[24] Rosenthal HP (1970) On the subspaces of L^p ($p > 2$) spanned by sequences of independent random variables. *Israel J. Math.* 8:273–303.
[25] Schudy W, Sviridenko M (2012) Concentration and moment inequalities for polynomials of independent random variables. *Proc. 23rd Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 437–446.
[26] Shmoys DB, Tardos É (1993) An approximation algorithm for the generalized assignment problem. *Math. Programming* 62(1–3):461–474.