Minimum-Norm Load Balancing Is (Almost) as Easy as Minimizing Makespan

Sharat Ibrahimpur
Department of Combinatorics and Optimization, University of Waterloo, Canada

Chaitanya Swamy
Department of Combinatorics and Optimization, University of Waterloo, Canada

Abstract

We consider the *minimum-norm load-balancing* (MinNormLB) problem, wherein there are n jobs, each of which needs to be assigned to one of m machines, and we are given the processing times $\{p_{ij}\}$ of the jobs on the machines. We also have a monotone, symmetric norm $f : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$. We seek an assignment σ of jobs to machines that minimizes the f-norm of the induced load vector $\overrightarrow{\mathsf{load}}_\sigma \in \mathbb{R}^m_{\geq 0}$, where $\mathsf{load}_\sigma(i) = \sum_{j:\sigma(j)=i} p_{ij}$. This problem was introduced by [4], and the current-best result for MinNormLB is a $(4+\epsilon)$-approximation [5]. In the stochastic version of MinNormLB, the job processing times are given by nonnegative random variables X_{ij}, and jobs are independent; the goal is to find an assignment σ that minimizes the *expected* f-norm of the induced random load vector.

We obtain results that (essentially) *match* the best-known guarantees for deterministic makespan minimization (MinNormLB with ℓ_∞ norm). For MinNormLB, we obtain a $(2+\epsilon)$-approximation for unrelated machines, and a PTAS for identical machines. For stochastic MinNormLB, we consider the setting where the X_{ij}s are *Poisson* random variables, denoted PoisNormLB. Our main result here is a novel and powerful reduction showing that, for any machine environment (e.g., unrelated/identical machines), *any* α-approximation algorithm for MinNormLB in that machine environment yields a randomized $\alpha(1+\epsilon)$-approximation for PoisNormLB in that machine environment. Combining this with our results for MinNormLB, we immediately obtain a $(2+\epsilon)$-approximation for PoisNormLB on unrelated machines, and a PTAS for PoisNormLB on identical machines. The latter result substantially generalizes a PTAS for makespan minimization with Poisson jobs obtained recently by [6].

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation algorithms, Load balancing, Minimum-norm optimization, LP rounding

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.81

Category Track A: Algorithms, Complexity and Games

Funding Supported in part by NSERC grant 327620-09 and an NSERC DAS Award.

1 Introduction

In the *minimum-norm load-balancing* (MinNormLB) problem, we are given a set J of n jobs, a set of m machines, and nonnegative job processing times (or sizes) $\{p_{ij}\}_{i\in[m],j\in J}$. We use $[m]$ to denote $\{1,\ldots,m\}$. We are also given a *monotone, symmetric norm* $f : \mathbb{R}^m \mapsto \mathbb{R}_{\geq 0}$. Recall that f being a norm means that: (i) $f(x) = 0$ iff $x = 0$; (ii) $f(x+y) \leq f(x) + f(y)$ for all $x, y \in \mathbb{R}^m$, and (iii) $f(\theta x) = |\theta| f(x)$ for all $x \in \mathbb{R}^m, \theta \in \mathbb{R}$. A monotone norm f satisfies $f(x) \leq f(y)$ for all $0 \leq x \leq y$, and symmetry is the property that permuting the coordinates of x does not change its norm. An assignment $\sigma : J \to [m]$ of jobs to machines induces the machine-load vector $\overrightarrow{\mathsf{load}}_\sigma = \big(\mathsf{load}_\sigma(i)\big)_{i\in[m]} \in \mathbb{R}^m_{\geq 0}$, where $\mathsf{load}_\sigma(i) := \sum_{j:\sigma(j)=i} p_{ij}$. The goal in MinNormLB is to find an assignment σ that minimizes the f-norm of $\overrightarrow{\mathsf{load}}_\sigma$.

This problem was first considered by [4], as a natural problem in the genre of minimum-norm optimization problems that they introduce. They gave a $(38 + \varepsilon)$-approximation algorithm for MinNormLB, and the approximation factor was subsequently improved to $(4 + \varepsilon)$ by [5]. Ibrahimpur and Swamy [12] introduced the genre of *stochastic* minimum-norm optimization, and considered StochNormLB, the stochastic generalization of MinNormLB, as a prominent problem in this genre. In StochNormLB, the job processing times are given by nonnegative random variables $\{X_{ij}\}_{i\in[m],j\in J}$ with specified distributions, and the jobs are independent (but X_{ij} and $X_{i'j}$ could be correlated); the goal is to find an assignment that minimizes the *expected* f-norm of the induced random load vector. As discussed in these works, one of the chief motivations and benefits of working with the rather broad class of monotone, symmetric norms is that it captures a variety of appealing objectives, including the frequently-considered (in both deterministic and stochastic settings) min-max (ℓ_∞) and min-sum (ℓ_1) objectives, general ℓ_p-norms, as also another important class of norms called Top_ℓ *norms* (for $x \geq 0$, $\mathsf{Top}_\ell(x)$ is the sum of the ℓ largest coordinates of x). Moreover, by exploiting the closure properties of monotone, symmetric norms, one can also model seemingly more general settings, such as when we have *multiple* monotone-symmetric-norm budget constraints $f_\ell(x) \leq B_\ell$ (a flexibility we leverage in Section 3).

The special case of MinNormLB where $f = \ell_\infty$ yields the classical (deterministic) *makespan-minimization* problem, which has been well studied for various machine environments. For unrelated machines, a 2-approximation is known [17, 23], and improving this factor remains a longstanding open problem, while various PTAS'es are known for identical and related machines [11, 10, 13, 14]. Similar guarantees are known for MinNormLB with (general) ℓ_p norms [2, 16, 20, 1]. Stochastic load balancing is less well-understood than its deterministic counterpart (even for the makespan objective). Constant-factor approximations are known for stochastic makespan minimization on identical [15] and unrelated [8] machines, StochNormLB with ℓ_p norms [22] and Top_ℓ norms [12], and StochNormLB with (an arbitrary f and) Bernoulli job sizes [12]. A common shortcoming of all these works is that the approximation factors obtained are quite large, at least in the 100s (although these works were not aiming to optimize the constant). With an eye towards obtaining small approximation factors, Goel and Indyk [7] considered stochastic makespan minimization on identical machines with structured distributions. They obtained (among other results) a 2-approximation for StochNormLB with Poisson job sizes – i.e., where each X_{ij} is a Poisson random variable – and very recently, this was improved to a PTAS [6].

Our contributions. As suggested by the title of the paper, we obtain results for both deterministic and stochastic load balancing with *approximation factors that (essentially) match the best-known approximation factors for the deterministic makespan-minimization problem.* Our salient results are as follows.

- For MinNormLB, we devise a $(2 + \epsilon)$-approximation for unrelated machines (Theorem 4.1), and a PTAS for identical machines (Theorem 5.1).

  For unrelated machines, this improves upon the previous-best $(4 + \epsilon)$-approximation [5].
- We consider StochNormLB with Poisson job sizes, denoted PoisNormLB, and give a novel, clean, and fairly general reduction showing that, for any machine environment (e.g., unrelated/identical machines), *any* α-approximation algorithm for MinNormLB in that machine environment can be used to obtain a randomized $\alpha(1 + \epsilon)$-approximation for PoisNormLB in that machine environment (Theorem 3.1). Combining this with our results for MinNormLB, we obtain a $(2+\epsilon)$-approximation for PoisNormLB on unrelated machines, and a PTAS for PoisNormLB on identical machines.

Our approximation factors (which are for general monotone, symmetric norms) are *considerably* smaller than the factors known (even) for stochastic makespan minimization with general distributions. Our PTAS for $\mathsf{PoisNormLB}$ on identical machines substantially generalizes the PTAS by [6] for stochastic makespan minimization with Poisson jobs. Our techniques are quite different, and our approach, based on the reduction to $\mathsf{MinNormLB}$, while being more general, is in fact also simpler and cleaner than the one in [6].

All our algorithms require only a *value oracle* for the norm f. (In contrast, the results of [4, 5] need more-sophisticated oracle access to f.)

Our techniques. We briefly discuss the key techniques underlying some of our results.

Consider first the reduction from $\mathsf{PoisNormLB}$ to $\mathsf{MinNormLB}$. Let $\mathsf{Pois}(\lambda)$ denote a Poisson random variable with parameter λ; recall that this has mean λ. The sum $\mathsf{Pois}(\lambda_1) + \mathsf{Pois}(\lambda_2)$ of independent Poisson random variables is a $\mathsf{Pois}(\lambda_1 + \lambda_2)$ random variable. So given an assignment $\sigma : J \to [m]$, the load on any machine i is a $\mathsf{Pois}(\Lambda_i^\sigma)$ random variable, where Λ_i^σ is the expected load on machine i, and the objective value of σ is $g(\Lambda^\sigma) := \mathbf{E}\big[f(\mathsf{Pois}(\Lambda_1^\sigma), \mathsf{Pois}(\Lambda_2^\sigma), \ldots, \mathsf{Pois}(\Lambda_m^\sigma))\big]$. It is known that if $\mathsf{Top}_i(y) \le \mathsf{Top}_i(y')$ for all $i \in [m]$ then $g(y) \le g(y')$ (Theorem 3.3). We argue that g is *subhomogeneous*, i.e., $g(\theta y) \le \theta \cdot g(y)$ for any $\theta \ge 1$, and so one can generalize the above statement to say that if $\mathsf{Top}_i(y) \le \theta \cdot \mathsf{Top}_i(y')$ for all $i \in [m]$, then $g(y) \le \theta \cdot g(y')$. Let Λ^* denote the Λ-vector of an optimal solution. The idea now is to "guess" $\mathsf{Top}_\ell(\Lambda^*)$ within a $(1+\varepsilon)$-factor for all ℓ in a certain sparse set $\mathsf{POS} \subseteq [m]$; let B_ℓ^* denote such an overestimate of $\mathsf{Top}_\ell(\Lambda^*)$. We now seek a solution σ such that $\mathsf{Top}_\ell(\Lambda^\sigma) \le \alpha B_\ell^*$ for all $\ell \in \mathsf{POS}$, for some $\alpha \ge 1$; this will imply that $\mathsf{Top}_i(\Lambda^\sigma) \le \alpha\big(1 + O(\varepsilon)\big)\mathsf{Top}_i(\Lambda^*)$ for all $i \in [m]$, and hence $g(\Lambda^\sigma) \le \alpha\big(1 + O(\varepsilon)\big)g(\Lambda^*)$. This is where the generality of monotone, symmetric norms comes in handy. *We can cast this multiple-Top_ℓ-norm-budgets problem as a $\mathsf{MinNormLB}$ problem, with $\{\lambda_{ij}\}$ job sizes, and monotone, symmetric norm given by $h(v) := \max_{\ell \in \mathsf{POS}} \mathsf{Top}_\ell(v)/B_\ell^*$*; we can now use an α-approximation algorithm for $\mathsf{MinNormLB}$, say $\mathcal{A}^{\mathsf{Det}}$, to find σ.

Note that the versatility afforded by $\mathsf{MinNormLB}$ as a model is crucial for the reduction, and we really need $\mathcal{A}^{\mathsf{Det}}$ to work for an arbitrary monotone, symmetric norm. We need to have fine-grained control of the Λ^σ vector (as dictated by the multiple Top_ℓ-norm budget constraints), and we define a suitable (monotone, symmetric) norm h to enforce this. In contrast, De et al. [6], who devise a PTAS for stochastic makespan minimization with Poisson jobs follow a completely different approach, based on proving suitable concentration results.

Our $(2+\epsilon)$-approximation for $\mathsf{MinNormLB}$ on unrelated machines (Section 4) is based on rounding the solution to a novel LP-relaxation for the problem. Let σ^* be an optimal solution, $\vec{o}$ be the induced load vector, and $\mathsf{OPT} = f(\vec{o})$. It suffices to obtain an assignment σ such that $\mathsf{Top}_\ell(\overrightarrow{\mathsf{load}_\sigma}) \le \big(2 + O(\delta)\big)\mathsf{Top}_\ell(\vec{o})$ for all $\ell \in \mathsf{POS}$ (see Theorem 2.4 and Claim 2.6). Roughly speaking, for all $\ell \in \mathsf{POS}$, we guess the ℓ-th largest entry of $\vec{o}$, and use this to linearly encode the constraint that $\mathsf{Top}_\ell(\overrightarrow{\mathsf{load}_\sigma}) \le \mathsf{Top}_\ell(\vec{o})$. LP-rounding algorithms for the special case of makespan minimization typically return a guarantee where the load on a machine is at most its LP-load + (maximum cost (i.e., size) of a job assigned to it). In order to bound the f-norm of the portion of the load-vector, say P, arising from the most-costly jobs, we also consider the vector $\overrightarrow{\mathsf{JL}^*}$ comprising the costs of the m most-costly jobs under σ^*. In [5], it is shown that $f(\overrightarrow{\mathsf{JL}^*}) \le \mathsf{OPT}$. We indirectly encode that $f(P) \le f(\overrightarrow{\mathsf{JL}^*})$ by guessing the ℓ-th largest entry, say ζ_ℓ, of $\overrightarrow{\mathsf{JL}^*}$ and encoding that there are at most $\ell - 1$ jobs of higher cost, for all $\ell \in \mathsf{POS}$. We round a fractional solution to the resulting LP using *iterative rounding* to obtain two types of guarantees *simultaneously* (see Lemma 4.3):

(i) similar to above, the load on machine i is at most its LP-load + $(1+\delta)\times$(maximum cost of a job assigned to i); and (ii) the number of jobs having cost larger then ζ_ℓ is at most $(1+\delta)\ell-1$. The LP constraints explicitly encode that the Top_ℓ-norms of the LP-load vector are roughly speaking at most $\mathsf{Top}_\ell(\vec{o})$, for all $\ell \in \mathsf{POS}$; guarantee (ii) allows us to argue that $f(P) \leq \big(1+O(\delta)\big)\mathsf{OPT}$. Together, these imply a $(2+\epsilon)$-approximation.

We remark that the approach taken in the GAP-rounding algorithm of [23] (and also used for Top_ℓ-norm minimization in [4]), wherein the problem is essentially reduced to a bipartite matching problem, does not seem to be helpful in obtaining a $(2+\epsilon)$-approximation for MinNormLB. This approach "hard-codes" the distinction between the most-costly job assigned to a machine and the remaining jobs as a means of bounding the load on a machine by its LP-load + (maximum cost of a job assigned to it). However, bounding $f(P)$ then entails solving (to within a $(1+\epsilon)$-factor) a (bipartite) matching problem with $|\mathsf{POS}| = O(\log m)$ side-constraints, or essentially a min-norm matching problem, but neither of these has (even) an $O(1)$-approximation. Instead, iterative rounding seems crucial for simultaneously obtaining guarantees (i) and (ii) above, and (ii) allows us to obtain a sufficiently-tight bound on the Top_ℓ-norms (and hence the f-norm) of P and thereby obtain our $(2+\epsilon)$-approximation.

2 Preliminaries

For a vector $v \in \mathbb{R}^m_{\geq 0}$, we use $v^\downarrow$ to denote v with its coordinates sorted in non-increasing order; i.e., we have $v^\downarrow_i = v_{\pi(i)}$, where π is a permutation of $[m]$ such that $v_{\pi(1)} \geq v_{\pi(2)} \geq \ldots \geq v_{\pi(m)}$. We say that v is non-increasing if $v_1 \geq \ldots \geq v_m$ (i.e., $v = v^\downarrow$). Whenever we say norm in the sequel, we always mean a monotone, symmetric norm. The following claim from [12] will be useful in obtaining bounds on the optimal value.

▷ Claim 2.1 (Claim 3.2 in [12]). Let $h : \mathbb{R}^m \mapsto \mathbb{R}_{\geq 0}$ be a monotone, symmetric norm. For any $v \in \mathbb{R}^m_{\geq 0}$, we have $\frac{\sum_{i\in[m]} v_i}{m} \leq \max_{i\in[m]} v_i \leq \frac{h(v)}{h(1,0,\ldots,0)} \leq \sum_{i\in[m]} v_i$.

As established in prior work on minimum-norm optimization [4, 12], we can control the norm of a vector by controlling all its Top_ℓ norms, which are defined below. Theorem 2.4 makes this notion precise. For $x \in \mathbb{R}$, we use $(x)^+$ to denote $\max\{0, x\}$.

► **Definition 2.2.** Let $\ell \in [m]$. The Top_ℓ *norm* is defined as follows: for $v \in \mathbb{R}^m_{\geq 0}$, $\mathsf{Top}_\ell(v)$ is the sum of the ℓ largest coordinates of v, i.e., $\mathsf{Top}_\ell(v) = \sum_{i=1}^{\ell} v^\downarrow_i$.

The following ways of reformulating the Top_ℓ-norm will be useful. For a vector $v \in \mathbb{R}^m$ and $\theta \in \mathbb{R}$, define $N^{>\theta}(v) := \big|\{i \in [m] : v_i > \theta\}\big|$.

▷ Claim 2.3. Let $v \in \mathbb{R}^m_{\geq 0}$, and $\ell \in [m]$. Then

$$\mathsf{Top}_\ell(v) = \min_{t\geq 0}\Big(\ell t + \sum_{i\in[m]}(v_i - t)^+\Big) = \ell v^\downarrow_\ell + \sum_{i\in[m]}(v_i - v^\downarrow_\ell)^+ = \int_0^\infty \min\{\ell, N^{>\theta}(v)\}d\theta.$$

► **Theorem 2.4** (Follows from structural result in [4], or majorization theory of [9]). *If $x, y \in \mathbb{R}^m_{\geq 0}$ are such that $\mathsf{Top}_\ell(x) \leq \alpha\mathsf{Top}_\ell(y) + \beta$ for all $\ell \in [m]$, where $\alpha, \beta \geq 0$, then $h(x) \leq \alpha \cdot h(y) + \beta \cdot h(1, 0, \ldots, 0)$ for any monotone, symmetric norm $h : \mathbb{R}^m \mapsto \mathbb{R}_{\geq 0}$.*

Theorem 2.4 will be our chief means for reasoning about the norm of a vector. Our algorithms will "guess" (i.e., enumerate) the Top_ℓ norms (or certain associated quantities) of the load vector $\vec{o}$ induced by an optimal solution, and will aim to obtain a solution whose induced load vector $\overrightarrow{\mathsf{load}}$ satisfies $\mathsf{Top}_\ell(\overrightarrow{\mathsf{load}}) = O\big(\mathsf{Top}_\ell(\vec{o})\big)$. However, to make this approach polynomial time, we will only be able to enumerate the Top_ℓ norms for a certain *sparse* subset of indices $\mathsf{POS} \subseteq [m]$. The next few definitions and results make this precise, and show that the move to this sparse subset only incurs the loss of a small factor.

Let $\delta > 0$ be a parameter. We define $\mathsf{POS}_{m,\delta} \subseteq [m]$ iteratively as follows: include the index 1 in $\mathsf{POS}_{m,\delta}$; as long as the largest index $\ell \in \mathsf{POS}_{m,\delta}$ is such that $\lceil (1+\delta)\ell \rceil \le m$, include $\lceil (1+\delta)\ell \rceil$ (which is larger than ℓ) in $\mathsf{POS}_{m,\delta}$. (This definition is mathematically slightly more convenient to work with than the one in [4], where $\mathsf{POS}_{m,\delta}$ is defined as $\big\{\min\{\lceil (1+\delta)^s \rceil, m\} : s \in \mathbb{Z}_{\ge 0}\big\}$, but this change is not crucial.)

▷ Claim 2.5. We have $|\mathsf{POS}_{m,\delta}| \le 1 + \log_{1+\delta} m = O\big(\frac{\log m}{\delta}\big)$.

We frequently abbreviate $\mathsf{POS}_{m,\delta}$ to POS in the remainder of this section, and whenever m, δ are clear from the context. For $i \in [m]$, let $\mathsf{next}(i)$ be the smallest index in POS (strictly) larger than i; if no such index exists, then we define $\mathsf{next}(i) := m + 1$ for notational convenience. Similarly, let $\mathsf{prev}(i)$ be the largest index in POS (strictly) smaller than i; set $\mathsf{prev}(1) := 0$. It is immediate from the definition of POS that $\mathsf{next}(\ell) - 1 \le (1+\delta)\ell$ for all $\ell \in \mathsf{POS}$; it follows also that $\mathsf{next}(i) - 1 \le (1+\delta)i$ for all $i \in [m]$. Claim 2.6 and Lemma 2.7 show that focusing on only the indices in POS only results in a $(1+\delta)$-factor loss.

▷ Claim 2.6. Let $u, v \in \mathbb{R}^m_{\ge 0}$ be such that $\mathsf{Top}_\ell(u) \le \mathsf{Top}_\ell(v)$ for all $\ell \in \mathsf{POS}_{m,\delta}$. Then we have $h(u) \le (1+\delta)h(v)$ for any monotone, symmetric norm $h : \mathbb{R}^m \mapsto \mathbb{R}_{\ge 0}$.

Proof. Let $i \in [m] \setminus \mathsf{POS}$, and $\ell = \mathsf{prev}(i)$. Then $i \le (1+\delta)\ell$, and therefore $\mathsf{Top}_i(u) \le (1+\delta)\mathsf{Top}_\ell(u) \le (1+\delta)\mathsf{Top}_\ell(v) \le (1+\delta)\mathsf{Top}_i(v)$. The claim now follows from Theorem 2.4. ◁

▶ **Lemma 2.7.** *Let $u, v \in \mathbb{R}^m_{\ge 0}$ be such that $u^\downarrow_\ell \le v^\downarrow_\ell$ for all $\ell \in \mathsf{POS}_{m,\delta}$. Then, we have $h(u) \le (1+\delta)h(v)$ for any monotone, symmetric norm $h : \mathbb{R}^m \mapsto \mathbb{R}_{\ge 0}$.*

Proof. By Theorem 2.4, it suffices to show that $\mathsf{Top}_i(u) \le (1+\delta)\mathsf{Top}_i(v)$ for all $i \in [m]$. So fix $i \in [m]$. We mimic the proof of Lemma 4.2 in [4]. Define vectors $\alpha, \beta \in \mathbb{R}^m_{\ge 0}$ as follows:

$$\alpha_k = \begin{cases} u^\downarrow_k; & \text{if } k \in \{1, \dots, i\} \\ 0 & \text{otherwise} \end{cases} \qquad \beta_k = \begin{cases} v^\downarrow_k; & \text{if } k \in \{1, \dots, i\} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, both α and β are non-increasing vectors. For notational convenience, set $\alpha_k := 0, \beta_k := 0$ for any $k > m$. We have

$$\begin{aligned} \mathsf{Top}_\ell(u) &= \sum_{k=1}^{m} \alpha_k = \sum_{\ell \in \mathsf{POS}} \sum_{k=\ell}^{\mathsf{next}(\ell)-1} \alpha_k \\ &\le \sum_{\ell \in \mathsf{POS}} \alpha_\ell \big((\mathsf{next}(\ell) - 1) - (\ell - 1) \big) = \sum_{\ell \in \mathsf{POS}} \big(\mathsf{next}(\ell) - 1\big)(\alpha_\ell - \alpha_{\mathsf{next}(\ell)}) \end{aligned}$$

where the inequality follows because α is a non-increasing vector. Now using the fact that $\mathsf{next}(\ell) - 1 \le (1+\delta)\ell$ for all $\ell \in \mathsf{POS}$, and $\alpha_\ell \le \beta_\ell$ for all $\ell \in \mathsf{POS}$, we obtain that

$$\begin{aligned} \mathsf{Top}_\ell(u) &\le (1+\delta) \sum_{\ell \in \mathsf{POS}} \ell(\alpha_\ell - \alpha_{\mathsf{next}(\ell)}) = (1+\delta) \sum_{\ell \in \mathsf{POS}} \alpha_\ell \big(\ell - \mathsf{prev}(\ell)\big) \le (1+\delta) \sum_{\ell \in \mathsf{POS}} \beta_\ell \big(\ell - \mathsf{prev}(\ell)\big) \\ &\le (1+\delta) \sum_{\ell \in \mathsf{POS}} \sum_{k=\mathsf{prev}(\ell)+1}^{\ell} \beta_k \le (1+\delta) \sum_{k=1}^{m} \beta_i = (1+\delta)\mathsf{Top}_\ell(v). \end{aligned}$$

The third inequality above follows because β is a non-increasing vector. ◀

In our algorithms, we work with estimates of $\{\vec{o}^{\downarrow}_\ell\}_{\ell\in\mathsf{POS}}$, where $\vec{o}$ is the load vector induced by an optimal solution. We show that these estimates then allow us to infer an estimate of $h(\vec{o})$ for any monotone, symmetric norm h. We need the following notation. Given a non-increasing vector $v \in \mathbb{R}^{\mathsf{POS}}_{\geq 0}$, we define its *expansion* to be the vector $v^{\mathsf{exp}} \in \mathbb{R}^m_{\geq 0}$ given by $v^{\mathsf{exp}}_i := v_i$ for $i \in \mathsf{POS}$, and $v^{\mathsf{exp}}_i = v_{\mathsf{prev}(i)}$ for $i \in [m] \setminus \mathsf{POS}$.

▶ **Lemma 2.8.** *Let $u \in \mathbb{R}^m_{\geq 0}$, and $v \in \mathbb{R}^{\mathsf{POS}}_{\geq 0}$ be a non-increasing vector. Let $h : \mathbb{R}^m \mapsto \mathbb{R}_{\geq 0}$ be a monotone, symmetric norm. Let $\varepsilon, \kappa > 0$.*

(a) *If $u^{\downarrow}_\ell \leq v_\ell$ for all $\ell \in \mathsf{POS}$, then $\mathsf{Top}_i(u) \leq \mathsf{Top}_i(v^{\mathsf{exp}})$ for all $i \in [m]$, and hence, $h(u) \leq h(v^{\mathsf{exp}})$.*

(b) *If $v_\ell \leq (1+\varepsilon)u^{\downarrow}_\ell + \kappa$ for all $\ell \in \mathsf{POS}$, then $\mathsf{Top}_i(v^{\mathsf{exp}}) \leq (1+\delta)(1+\varepsilon)\mathsf{Top}_i(u) + i\kappa$ for all $i \in [m]$, and hence, $h(v^{\mathsf{exp}}) \leq (1+\delta)(1+\varepsilon)h(u) + m\kappa \cdot h(1,0,\ldots,0)$.*

(c) *Let v be as in part (b). Let $\delta \leq 1$ (in $\mathsf{POS} = \mathsf{POS}_{m,\delta}$). Let $\alpha \in \mathbb{R}^m$ be such that $\alpha^{\downarrow}_1 \leq v_1$ and $N^{>v_\ell}(\alpha) \leq (1+\delta)\ell - 1$ for all $\ell \in \mathsf{POS}$. Then, $\mathsf{Top}_i(\alpha) \leq (1+4\delta)(1+\varepsilon)\mathsf{Top}_i(u) + 5i\kappa$ for all $i \in [m]$. Hence, $h(\alpha) \leq (1+4\delta)(1+\varepsilon)h(u) + 5m\kappa \cdot h(1,0,\ldots,0)$.*

Lemma 2.8 (c) can be seen as a generalization of Lemma 2.8 (b): the vector v^{exp} satisfies $N^{>v_\ell}(v^{\mathsf{exp}}) \leq \ell - 1$ for all $\ell \in \mathsf{POS}$, while the vector α satisfies a relaxed version of this bound.

Proof. The inequalities involving $h(.)$ in parts (a)–(c) follow from the corresponding bounds on the Top_i-norms, using Theorem 2.4. So we focus on the proving the bounds for the Top_i-norms.

Let $\gamma = v^{\mathsf{exp}}$. Part (a) follows immediately from the fact that $u^{\downarrow} \leq \gamma$. For part (b), define $\beta \in \mathbb{R}^m_{\geq 0}$ to be the expansion of $(u^{\downarrow}_\ell)_{\ell\in\mathsf{POS}}$. Then, we have $\gamma \leq (1+\varepsilon)\beta + \kappa \cdot \mathbf{1}$, where $\mathbf{1}$ is the vector of all 1s, and hence, $\mathsf{Top}_i(\gamma) \leq (1+\varepsilon)\mathsf{Top}_i(\beta) + i\kappa$, for any $i \in [m]$. Observe that $\beta_\ell \leq u^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}$. Therefore, by Lemma 2.7, we have $\mathsf{Top}_i(\beta) \leq (1+\delta)\mathsf{Top}_i(u)$ for any $i \in [m]$.

For part (c), consider an index $i \in [m]$. We use the reformulation $\mathsf{Top}_i(\alpha) = \int_0^\infty \min\{i, N^{>\theta}(\alpha)\}d\theta$ stated in Claim 2.3. Since $N^{>v_1}(\alpha) = 0$, we only need to go up to v_1 in the above integral. Let $\bar{\ell} = i$ if $i \in \mathsf{POS}$, and $\bar{\ell} = \mathsf{prev}(i)$ otherwise. Observe that $i \leq (1+\delta)\bar{\ell}$. We have the following chain of inequalities.

$$
\begin{aligned}
\mathsf{Top}_i(\alpha) &\leq \int_0^{v_{\bar{\ell}}} i d\theta + \sum_{\ell\in\mathsf{POS}:1<\ell\leq\bar{\ell}} \int_{v_\ell}^{v_{\mathsf{prev}(\ell)}} N^{>\theta}(\alpha)d\theta \leq i \cdot v_{\bar{\ell}} + \sum_{\ell\in\mathsf{POS}:1<\ell\leq\bar{\ell}} (v_{\mathsf{prev}(\ell)} - v_\ell)N^{>v_\ell}(\alpha) \\
&\leq i \cdot v_{\bar{\ell}} + \sum_{\ell\in\mathsf{POS}:1<\ell\leq\bar{\ell}} (v_{\mathsf{prev}(\ell)} - v_\ell)\big((1+\delta)\ell - 1\big) \\
&\leq i \cdot v_{\bar{\ell}} + \sum_{\ell\in\mathsf{POS}:1<\ell\leq\bar{\ell}} (v_{\mathsf{prev}(\ell)} - v_\ell)\big((1+\delta)^2\mathsf{prev}(\ell) + \delta\big). \qquad (1)
\end{aligned}
$$

The second inequality is because $N^{>\theta}(\alpha)$ is non-increasing in θ; the third is from condition (ii) in the lemma statement; and the final inequality (1) follows since $\ell - 1 \leq (1+\delta)\mathsf{prev}(\ell)$. Recall that $\mathsf{prev}(1) = 0$. Continuing, since $i \leq (1+\delta)\bar{\ell} \leq (1+\delta)^2\bar{\ell}$, the RHS of (1) is at most

$$
\begin{aligned}
&(1+\delta)^2 \sum_{\ell\in\mathsf{POS}:\ell\leq\bar{\ell}} v_\ell\big(\ell - \mathsf{prev}(\ell)\big) + \delta v_1 \\
&\leq (1+\delta)^2 \sum_{\ell\in\mathsf{POS}:\ell\leq\bar{\ell}} \big((1+\varepsilon)u^{\downarrow}_\ell + \kappa\big)\big(\ell - \mathsf{prev}(\ell)\big) + \delta(1+\varepsilon)\cdot\mathsf{Top}_i(u) + \delta\kappa
\end{aligned}
$$

$$\ldots \le (1+\delta)^2(1+\varepsilon)\sum_{i'=1}^{\bar\ell} u^{\downarrow}_{i'} + (1+\delta)^2\bar\ell\kappa + \delta(1+\varepsilon)\cdot \mathsf{Top}_i(u) + \delta\kappa$$
$$\le (1+4\delta)(1+\varepsilon)\mathsf{Top}_i(u) + 5i\kappa. \qquad \text{(since } \delta \le 1) \qquad \blacktriangleleft$$

Our algorithms will often need to estimate a non-increasing vector α (such as $(\vec{o}^{\downarrow}_\ell)_{\ell\in\mathsf{POS}}$). We show that if we have suitable bounds on the coordinates of α, then one can identify a (polynomially bounded) set containing a vector close to α.

▶ **Lemma 2.9.** *Let $\mathcal{L} \subseteq [m]$ be an index-set. Let $\alpha \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$ be a non-increasing vector, i.e., $\alpha_\ell \ge \alpha_{\ell'}$ for indices $\ell, \ell' \in \mathcal{L}$, $\ell < \ell'$. Let ub be such that $\alpha_\ell \le \mathsf{ub}$ for all $\ell \in \mathcal{L}$. Let $\varepsilon, \kappa > 0$, and $\varepsilon' = \min\{1, \varepsilon\}$.*

(a) *Let $N_1 := (2e)^{|\mathcal{L}|} + \left(\frac{\mathsf{ub}}{\kappa}\right)^{O(\frac{1}{\varepsilon'})}$. We can construct a set $\mathcal{T} \subseteq \mathbb{R}^{\mathcal{L}}_{\ge 0}$ with $|\mathcal{T}| \le N_1$ in $O(N_1)$ time, containing a non-increasing vector $v \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$, such that $\alpha_\ell \le v_\ell \le (1+\varepsilon)\alpha_\ell + \kappa$ for all $\ell \in \mathcal{L}$.*

(b) *Suppose that we also have $\alpha_\ell \ge \mathsf{lb}$ for all $\ell \in \mathcal{L}$, where $\mathsf{lb} > 0$. Let $N_2 := (2e)^{|\mathcal{L}|} + \left(\frac{\mathsf{ub}}{\mathsf{lb}}\right)^{O(\frac{1}{\varepsilon'})}$. We can construct $\mathcal{T} \subseteq \mathbb{R}^{\mathcal{L}}_{\ge 0}$ with $|\mathcal{T}| \le N_2$ in $O(N_2)$ time, containing a non-increasing vector $v \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$, such that $\alpha_\ell \le v_\ell \le (1+\varepsilon)\alpha_\ell$ for all $\ell \in \mathcal{L}$.*

Proof. We utilize the following standard result, lifted from [4], that gives a bound on the number of non-increasing vectors with bounded coordinates.

▷ Claim 2.10. There are at most $(2e)^{\max\{M,k\}}$ non-increasing sequences of k integers chosen from $\{0, \ldots, M\}$.

For part (a), consider the set

$$\mathcal{T} \;:=\; \Big\{\vec{t} \in \mathbb{R}^{\mathcal{L}}_{\ge 0} : \quad \vec{t} \text{ is a non-increasing vector,}$$
$$\forall \ell \in \mathcal{L}, \quad t_\ell = \tfrac{\mathsf{ub}}{(1+\varepsilon)^k}, \text{ where } k \in \mathbb{Z}_{\ge 0}, \quad t_\ell \ge \tfrac{\kappa}{1+\varepsilon}\Big\}.$$

Each $\vec{t} \in \mathcal{T}$ is a non-increasing vector, and there are $I := 1 + \left\lfloor \log_{1+\varepsilon} \frac{(1+\varepsilon)\mathsf{ub}}{\kappa} \right\rfloor = O\!\left(\frac{\log(\mathsf{ub}/\kappa)}{\varepsilon'}\right)$ choices for $\log_{1+\varepsilon} \frac{\mathsf{ub}}{t_\ell}$ for every $\ell \in \mathsf{POS}$. (Recall that $\varepsilon' = \min\{\varepsilon, 1\}$.) So Claim 2.10 implies that $|\mathcal{T}| \le (2e)^{\max\{|\mathcal{L}|, I\}}$. We have $(2e)^I \le \left(\frac{\mathsf{ub}}{\kappa}\right)^{O(\frac{1}{\varepsilon'})}$, so this yields the bound on $|\mathcal{T}|$ and the time to construct $\mathcal{T}$.

Consider the vector $v \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$, where for every $\ell \in \mathcal{L}$, v_ℓ is the smallest number of the form $\mathsf{ub}/(1+\varepsilon)^k$, $k \in \mathbb{Z}_{\ge 0}$ that is at least $\max\{\kappa/(1+\varepsilon), \alpha_\ell\}$. Then, v is a non-increasing vector, $v \in \mathcal{T}$, and $\alpha_\ell \le v_\ell \le (1+\varepsilon)\alpha_\ell + \kappa$ for all $\ell \in \mathcal{L}$.

Part (b) is proved very similarly. We now take $\mathcal{T}$ to be the set of all non-increasing vectors $\vec{t} \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$ satisfying $t_\ell \ge \mathsf{lb}$, $t_\ell = \frac{\mathsf{ub}}{(1+\varepsilon)^k}$ where $k \in \mathbb{Z}_{\ge 0}$, for all $\ell \in \mathcal{L}$. As before, one can infer that the size of $\mathcal{T}$ and the time taken to construct it are bounded by $(2e)^{|\mathcal{L}|} + \left(\frac{\mathsf{ub}}{\mathsf{lb}}\right)^{O(\frac{1}{\varepsilon'})}$. Now if $v \in \mathbb{R}^{\mathcal{L}}_{\ge 0}$ is such that, for every $\ell \in \mathcal{L}$, v_ℓ is the smallest number of the form $\mathsf{ub}/(1+\varepsilon)^k$, $k \in \mathbb{Z}_{\ge 0}$ that is at least α_ℓ, then, v is a non-increasing vector, $v \in \mathcal{T}$, and $\alpha_\ell \le v_\ell \le (1+\varepsilon)\alpha_\ell$ for all $\ell \in \mathcal{L}$. ◀

Poisson random variables. A discrete random variable Z is said to have a Poisson distribution with parameter $\lambda \ge 0$, denoted $Z \sim \mathsf{Pois}(\lambda)$, if $\mathbf{Pr}\big[Z = k\big] = e^{-\lambda}\lambda^k/k!$ for all $k \in \mathbb{Z}_{\ge 0}$.

▶ **Fact 2.11.** *The following facts about Poisson random variables are well known.*

(a) *The mean and variance of $\mathsf{Pois}(\lambda)$ are both equal to λ.*

(b) *Let $\{Z_j\}_j$ be a collection of independent Poisson variables with parameters $\{\lambda_j\}_j$. Then, $S = \sum_j Z_j$ is distributed as $\mathsf{Pois}(\sum_j \lambda_j)$.*

3 Stochastic Minimum Norm Load Balancing with Poisson Jobs

We now consider StochNormLB with Poisson job sizes, denoted PoisNormLB, wherein the processing time of a job j on machine i is a $\mathsf{Pois}(\lambda_{ij})$ random variable and we seek to minimize the expected f-norm of the load vector. Jobs are independent, but processing times of the same job could be correlated across machines. Our main result is a novel, clean, and versatile *black-box reduction* from PoisNormLB to the deterministic problem, MinNormLB, showing that, for any machine environment (i.e., unrelated/identical machines), guarantees obtained for MinNormLB translate to yield essentially the same guarantees for PoisNormLB.

▶ **Theorem 3.1.** *Let $\mathcal{I}^{\mathsf{Pois}} = (J, [m], \{\lambda_{ij}\}, f)$ be an instance of* PoisNormLB*, and $\mathcal{A}^{\mathsf{Det}}$ be an ρ-approximation algorithm for* MinNormLB*-instances with job-set J, machine-set $[m]$, and $\{\lambda_{ij}\}_{i\in[m],j\in J}$ job sizes. For any $\varepsilon, \eta > 0$, we can utilize $\mathcal{A}^{\mathsf{Det}}$ to obtain an $\rho\big(1 + O(\varepsilon)\big)$-approximate solution to* PoisNormLB *with probability at least $1 - \eta$, in time $\mathrm{poly}\big(m^{1/\varepsilon}, n, \frac{1}{\eta}\big)$. The run time also bounds the number of calls to $\mathcal{A}^{\mathsf{Det}}$ and the sample size.*

We emphasize that: (a) the above reduction preserves the machine environment: for instance, if we have identical machines ($\lambda_{ij} = \lambda_j$ for all i, j), we only need $\mathcal{A}^{\mathsf{Det}}$ to work for identical machines; and (b) algorithm $\mathcal{A}^{\mathsf{Det}}$ is required to work for an arbitrary monotone, symmetric norm (and not just the norm f): this generality is *crucial* for the above reduction and brings to the fore a prime benefit of working at the level of generality of monotone, symmetric norms. Combining the above reduction with our results for MinNormLB in Sections 4 and 5 *immediately* yields the following results as corollaries. (We do not explicitly indicate the failure probability η below; the sample size, for a fixed ε, is $\mathrm{poly}(m)/\eta$.)

▶ **Theorem 3.2.**

(a) **(Follows from Theorems 3.1 and 4.1)** *For any $\varepsilon > 0$, there is a randomized $\big(2 + O(\varepsilon)\big)$-approximation algorithm for* PoisNormLB *on unrelated machines .*

(b) **(Follows from Theorems 3.1 and 5.1)** *There is a randomized PTAS for* PoisNormLB *on identical machines.*

We discuss the chief ideas behind the reduction in Theorem 3.1, deferring some details to the full version of the paper. Since the sum of independent Poisson random variables is another Poisson random variable (Fact 2.11 (b)), the objective value of an assignment $\sigma : J \to [m]$ depends only the aggregate λ-vector $\Lambda^\sigma = (\Lambda_i^\sigma)_{i\in[m]}$, where $\Lambda_i^\sigma := \sum_{j:\sigma(j)=i} \lambda_{ij}$ for all $i \in [m]$. We drop σ in Λ^σ if the assignment σ is clear from the context. Overloading notation, for a vector $y \in \mathbb{R}^m_{\geq 0}$, we use $\mathsf{Pois}(y)$ to denote the random vector $\big(\mathsf{Pois}(y_1), \dots, \mathsf{Pois}(y_m)\big)$ of *independent* Poisson random variables. Defining $g(y) := \mathbf{E}\big[f(\mathsf{Pois}(y))\big]$ for $y \in \mathbb{R}^m_{\geq 0}$, the goal in PoisNormLB is to find an assignment σ that minimizes $g(\Lambda^\sigma)$. The function g is *not* convex, but it satisfies the following inequality [21] (see Chapter 11, Proposition E.6), which is closely related to a property called *Schur convexity* that is satisfied by all symmetric convex functions. Theorem 3.3 provides a means for controlling $g(y)$, by bounding the Top_ℓ-norms of y, and is key to our approach. We give a self-contained proof of Theorem 3.3 in the full version.

▶ **Theorem 3.3.** *Let $y, y' \in \mathbb{R}^m_{\geq 0}$. If $\mathsf{Top}_i(y) \leq \mathsf{Top}_i(y')$ for all $i \in [m]$, then $g(y) \leq g(y')$.*

To keep notation simple, we assume that f is normalized so that $f(1, 0, \dots, 0) = 1$; clearly, this is without loss of generality. Let σ^* be an optimal solution to the PoisNormLB-instance $\mathcal{I}^{\mathsf{Pois}}$. Let $\Lambda^* := \Lambda^{\sigma^*}$. The idea underlying our reduction is strikingly simple. Given Theorem 3.3, we aim to (ideally) find an assignment σ such that $\mathsf{Top}_i(\Lambda^\sigma) \leq \mathsf{Top}_i(\Lambda^*)$ for all $i \in [m]$. One of our chief insights is that *this amounts to solving a deterministic* *min-norm load balancing problem* with job sizes $\{\lambda_{ij}\}_{i,j}$, and the monotone, symmetric norm $h : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ given by $h(v) := \max_{i\in[m]} \mathsf{Top}_i(v)/\mathsf{Top}_i(\Lambda^*)$. Now σ^* yields a solution to this MinNormLB-instance of cost 1, and so solving this MinNormLB problem optimally, and utilizing Theorem 3.3, would yield the desired solution.

Further ingredients are needed to make this idea work. We do not know the $\mathsf{Top}_i(\Lambda^*)$ values, and cannot "guess" these values for all $i \in [m]$; moreover, we cannot solve the MinNormLB problem optimally. We utilize the sparsification tools from Section 2, and, with a small loss in approximation, move to the sparse set $\mathsf{POS} = \mathsf{POS}_{m,\delta}$ (for, say, $\delta = \min\{0.5, \varepsilon\}$) and work with estimates B_ℓ of $\mathsf{Top}_\ell(\Lambda^*)$ for all $\ell \in \mathsf{POS}$; so the norm in the MinNormLB instance is now $h(v) := \max_{\ell\in\mathsf{POS}} \mathsf{Top}_\ell(v)/B_\ell$. Now, using the algorithm $\mathcal{A}^{\mathsf{Det}}$ with the correct estimate-vector $B^* \in \mathbb{R}^{\mathsf{POS}}_{\geq 0}$ (where each B^*_ℓ overestimates $\mathsf{Top}_\ell(\Lambda^*)$ within a $(1+\delta)$-factor), we obtain an assignment σ such that $\mathsf{Top}_i(\Lambda^\sigma) \leq \alpha'\mathsf{Top}_i(\Lambda^*)$ for all $i \in [m]$, where $\alpha' = \alpha\big(1 + O(\varepsilon)\big)$. Theorem 3.3 then shows that $g(\Lambda^\sigma) \leq g(\alpha'\Lambda^*)$, but we need a bound in terms of $g(\Lambda^*)$. To this end, we prove the important property that g is *subhomogeneous* (Lemma 3.5): $g(\theta y) \leq \theta \cdot g(y)$ for any $\theta \geq 1$. Finally, we cannot quite identify the correct B^*, but we can isolate it in a polynomial-size set. We show how to estimate $g(y)$ using polynomially many samples (Lemma 3.6), and utilize this estimator to find (loosely speaking) the best solution among those computed for each candidate estimate-vector in this set. Combining these various ingredients yields Theorem 3.1.

▶ **Lemma 3.4.** *Let $y \in \mathbb{R}^m_{\geq 0}$. We have $\max\{f(y), 1 - e^{-\mathsf{Top}_m(y)}\} \leq g(y) \leq \mathsf{Top}_m(y)$.*

Proof. To prove the upper bound on g we use Claim 2.1 (recall that f is normalized) and Fact 2.11 (a): $\mathbf{E}\big[f(\mathsf{Pois}(y))\big] \leq \mathbf{E}\big[\mathsf{Top}_m(\mathsf{Pois}(y))\big] = \mathbf{E}\big[\sum_{i\in[m]} \mathsf{Pois}(y_i)\big] = \mathsf{Top}_m(y)$. The first lower bound follows from convexity of norms: $\mathbf{E}\big[f(\mathsf{Pois}(y))\big] \geq f(\mathbf{E}\big[\mathsf{Pois}(y)\big]) = f(y)$. Lastly, for the second lower bound, we use Claim 2.1:

$$\mathbf{E}\big[f(\mathsf{Pois}(y))\big] \geq \mathbf{E}\big[\mathsf{Top}_1(\mathsf{Pois}(y))\big] \geq \mathbf{Pr}\big[\mathsf{Top}_1(\mathsf{Pois}(y)) > 0\big]$$
$$= 1 - \prod_{i\in[m]} \mathbf{Pr}\big[\mathsf{Pois}(y_i) = 0\big] = 1 - e^{-\mathsf{Top}_m(y)}. \quad ◀$$

▶ **Lemma 3.5** (Subhomogeneity)**.** *For any $y \in \mathbb{R}^m_{\geq 0}$ and scalar $\theta \geq 1$, we have $g(\theta y) \leq \theta \cdot g(y)$.*

Proof. We prove this for rational θ. The proof for general θ then follows from a continuity argument, which we defer to the full version. Let $\theta = a/b$ for integers $a > b \geq 1$. (If $a = b$, there is nothing to be shown.) Observe that $g(\theta y) = g(az)$ and $g(y) = g(bz)$, where $z = y/b$. So, for the rational case, it suffices to prove that $g(az) \leq \frac{a}{b} g(bz)$ holds for all $z \in \mathbb{R}^m_{\geq 0}$ and integers $a > b \geq 1$. Fix some $z \in \mathbb{R}^m_{\geq 0}$. Let $Z^{(0)}, Z^{(1)}, \dots, Z^{(a-1)}$ be a independent random vectors that are identically distributed copies of $\mathsf{Pois}(z)$ (so each $Z^{(j)}_i$ is an independent $\mathsf{Pois}(z_i)$ random variable). For any $i \in [m]$, $\mathsf{Pois}(az_i)$ is identically distributed as $\sum_{j=0}^{a-1} Z^{(j)}_i$ (Fact b), so $g(az) = \mathbf{E}\big[f\big(\mathsf{Pois}(az)\big)\big] = \mathbf{E}\big[f\big(\sum_{j=0}^{a-1} Z^{(j)}\big)\big]$. Also, for any subset $S \subseteq \{0, 1, \dots, a\}$ with $|S| = b$, we have $\mathbf{E}\big[f\big(\sum_{j\in S} Z^{(j)}\big)\big] = g(bz)$. Define size-$b$ index sets $S_k := \{(k + j) \bmod a : j = 0, \dots, b - 1\}$, for $k = 0, 1, \dots, a - 1$. Note that each $j \in \{0, \dots, a - 1\}$ is contained in exactly b of these sets.

$$g(az) = \mathbf{E}\big[f\big(\sum_{j=0}^{a-1} Z^{(j)}\big)\big] = \mathbf{E}\big[f\big(\tfrac{1}{b} \cdot \sum_{k=0}^{a-1} \sum_{j\in S_k} Z^{(j)}\big)\big] \leq \frac{1}{b} \cdot \sum_{k=0}^{a-1} \mathbf{E}\big[f\big(\sum_{j\in S_k} Z^{(j)}\big)\big] = \frac{a}{b} \cdot g(bz). \quad ◀$$

▶ **Lemma 3.6.** *Let $\varepsilon, \eta > 0$. Let $y \in \mathbb{R}^m_{\geq 0}$ be such that $\mathsf{Top}_m(y) \geq \varepsilon$. Let $N := 2\max\{m^2, 4/\varepsilon\}/(\varepsilon^2\eta)$. Using at most N independent samples from $\mathsf{Pois}(y)$, we can compute an estimate γ satisfying $\mathbf{Pr}\big[\gamma \in [(1 - \varepsilon)g(y), (1 + \varepsilon)g(y)]\big] \geq 1 - \eta$.*

Proof. Let $x^{(1)}, \dots, x^{(N)}$ be N independent samples from $\mathsf{Pois}(y)$, and let $\gamma := \frac{1}{N}\sum_{j=1}^{N} f(x^{(j)})$ denote the sample f-average. We show that γ is the desired estimator by using Chebyshev's concentration inequality. To this end, we need a bound on the variance of the real-valued random variable $f(\mathsf{Pois}(y))$. We have $\mathrm{Var}\big[f(\mathsf{Pois}(y))\big] \le \mathbf{E}\big[f^2(\mathsf{Pois}(y))\big] \le \mathbf{E}\big[\mathsf{Top}_m^2(\mathsf{Pois}(y))\big]$ by Claim 2.1, and $\mathbf{E}\big[\mathsf{Top}_m^2(\mathsf{Pois}(y))\big] = \mathrm{Var}\big[\mathsf{Pois}(\mathsf{Top}_m(y))\big] + \big(\mathbf{E}\big[\mathsf{Pois}(\mathsf{Top}_m(y))\big]\big)^2$. By Fact b, $\mathsf{Top}_m(\mathsf{Pois}(y))$ is distributed as $\mathsf{Pois}(\mathsf{Top}_m(y))$. As the variance of a Poisson variable with mean λ is λ, we obtain that

$$\mathrm{Var}\big[f(\mathsf{Pois}(y))\big] \le \mathrm{Var}\big[\mathsf{Pois}(\mathsf{Top}_m(y))\big] + \big(\mathbf{E}\big[\mathsf{Pois}(\mathsf{Top}_m(y))\big]\big)^2 \le 2\mathsf{Top}_m(y)\max(1, \mathsf{Top}_m(y)).$$

Since γ is an average of $f(\mathsf{Pois}(y))$ over N independent samples, we have $\mathrm{Var}\big[\gamma\big] = \mathrm{Var}\big[f(\mathsf{Pois}(y))\big]/N$. By Chebyshev's inequality,

$$\mathbf{Pr}\big[|\gamma - g(y)| > \varepsilon g(y)\big] \le \frac{\mathrm{Var}\big[\gamma\big]}{\varepsilon^2 g^2(y)} \le \frac{2}{N\varepsilon^2}\cdot\frac{\mathsf{Top}_m(y)\max(1, \mathsf{Top}_m(y))}{g^2(y)} \le \frac{2\max\{m^2, 4/\varepsilon\}}{N\epsilon^2} \le \eta$$

It remains to justify the penultimate inequality used above. First, observe that $\mathsf{Top}_m(y)/g(y) \le m$ for any $y \in \mathbb{R}^m_{\ge 0}$, so the inequality holds when $\mathsf{Top}_m(y) \ge 1$. Suppose $\mathsf{Top}_m(y) \in [\varepsilon, 1]$. We have $g(y) \ge 1 - e^{-\mathsf{Top}_m(y)}$ (by Lemma 3.4), which is at least $\mathsf{Top}_m(y)(1 - \mathsf{Top}_m(y)/2) \ge \mathsf{Top}_m(y)/2$. This finishes the proof of the lemma. ◀

Proof of Theorem 3.1. Set $\epsilon' = \delta = \min\{0.5, \varepsilon\}$. Let σ^{sum} be the assignment that minimizes the expected sum of machine loads, so $\sigma^{\mathsf{sum}}(j) = \mathrm{argmin}_i \lambda_{ij}$ for all jobs j. Let $\Lambda^{\mathsf{sum}} := \Lambda^{\sigma^{\mathsf{sum}}}$, and $\mathsf{UB} := \mathsf{Top}_m(\Lambda^{\mathsf{sum}})$. If $\mathsf{UB} \le \epsilon'$, then we claim that σ^{sum} is a $(1+\epsilon')$-approximate solution to $\mathsf{PoisNormLB}$. This is because we have $\mathsf{Top}_m(\Lambda^*) \ge \mathsf{UB}$, and so by Lemma 3.4, we have that $g(\Lambda^{\mathsf{sum}}) \le \mathsf{UB}$ and $g(\Lambda^*) \ge 1 - e^{-\mathsf{UB}} \ge \mathsf{UB}(1 - \mathsf{UB}/2) \ge \mathsf{UB}/(1+\epsilon')$, where we use that $\mathsf{UB} \le \epsilon' \le 1$.

So suppose that $\mathsf{UB} \ge \epsilon'$ (and so $\mathsf{Top}_m(\Lambda^\sigma) \ge \epsilon'$ for every assignment σ). We have $\mathsf{Top}_1(\Lambda^*) \ge \mathsf{Top}_m(\Lambda^*)/m \ge \mathsf{UB}/m$. Also, $\mathsf{Top}_m(\Lambda^*) \le m\mathsf{Top}_1(\Lambda^*)$, and we have $\mathsf{Top}_1(\Lambda^*) \le f(\Lambda^*) \le g(\Lambda^*) \le g(\Lambda^{\mathsf{sum}}) \le \mathsf{UB}$; here, we have used Claim 2.1, Lemma 3.4 (twice), and the optimality of σ^*. So we obtain that $\mathsf{Top}_m(\Lambda^*) \le m \cdot \mathsf{UB}$.

Now consider the non-increasing vector u which is $(\Lambda^*_\ell)_{\ell\in\mathsf{POS}}$ with its coordinates listed in *decreasing* order of ℓ. We apply Lemma 2.9 (b) on u, taking $\mathcal{L} = \mathsf{POS}$, and upper and lower bounds $m \cdot \mathsf{UB}$ and UB/m respectively, to obtain a $\mathrm{poly}(m^{1/\delta})$-size set $\mathcal{T} \subseteq \mathbb{R}^{\mathsf{POS}}_{\ge 0}$ containing a vector B^* such that $\mathsf{Top}_\ell(\Lambda^*) \le B^*_\ell \le (1+\delta)\mathsf{Top}_\ell(\Lambda^*)$ for all $\ell \in \mathsf{POS}$.

For each $B \in \mathcal{T}$, we do the following. Let $h_B : \mathbb{R}^m \to \mathbb{R}_{\ge 0}$ be the monotone, symmetric norm defined as $h_B(v) := \max_{\ell\in\mathsf{POS}} \mathsf{Top}_\ell(v)/B_\ell$. We run $\mathcal{A}^{\mathsf{Det}}$ on the $\mathsf{MinNormLB}$ instance with $\{\lambda_{ij}\}_{i\in[m], j\in J}$ job sizes and norm h_B, to obtain an assignment σ^B. Let $\Lambda^B = \Lambda^{\sigma^B}$. We use Lemma 3.6 to compute an estimate γ^B such that $\mathbf{Pr}\big[\gamma^B \in [(1-\epsilon')g(\Lambda^B), (1+\epsilon')g(\Lambda^B)]\big] \ge 1 - \frac{\eta}{|\mathcal{T}|}$. We output the assignment $\sigma^{B'}$ with smallest $\gamma^{B'}$ value among all $B' \in \mathcal{T}$.

We argue that this is an $\rho\big(1+O(\varepsilon)\big)$-approximation with probability at least $1-\eta$. By the union bound, with probability at least $1-\eta$, we have that $\gamma^B \in [(1-\epsilon')g(\Lambda^B), (1+\epsilon')g(\Lambda^B)]$ for all $B \in \mathcal{T}$; we assume that this event happens. Consider the correct guess $B^* \in \mathcal{T}$. We have $h_{B^*}(\Lambda^*) \le 1$, and so the solution σ^{B^*} satisfies $\mathsf{Top}_\ell(\Lambda^{B^*}) \le \rho B^*_\ell \le \rho(1+\delta)\mathsf{Top}_\ell(\Lambda^*)$ for all $\ell \in \mathsf{POS}$. Using Claim 2.6, we have $\mathsf{Top}_i(\Lambda^{B^*}) \le \rho(1+\delta)^2\mathsf{Top}_i(\Lambda^*)$ for all $i \in [m]$. By Theorem 3.3 and Lemma 3.5, we then obtain that $g(\Lambda^{B^*}) \le \rho(1+\delta)^2 g(\Lambda^*)$. Accounting for the error due to the γ^B estimates, yields $g(\Lambda^{B'}) \le \frac{1+\epsilon'}{1-\epsilon'} \cdot g(\Lambda^{B^*}) \le \rho(1+15\epsilon')g(\Lambda^*)$. ◀

4 A $(2+\epsilon)$-approximation for MinNormLB on unrelated machines

▶ **Theorem 4.1.** *For any $\varepsilon > 0$, there is a $\big(2+O(\varepsilon)\big)$-approximation algorithm for* MinNormLB *on unrelated machines.*

Our algorithm yielding the above theorem is based on LP-rounding. Our relaxation is different from the LP used in [4] and the convex program used in [5] for MinNormLB. The latter relaxation, which was used to obtain the previous best $(4+\varepsilon)$-approximation has an integrality gap of (roughly) 4 [3], and therefore we need new ideas to obtain our $(2+\varepsilon)$-approximation algorithm. Indeed, our LP and rounding algorithm leverage and build upon the ideas used in the above works in a novel fashion.

Let $n = |J|$ be the number of jobs. Let σ^* denote an optimal solution, $\vec{o} := \big(\overrightarrow{\mathsf{load}_{\sigma^*}}\big)$ be the load vector induced by σ^*, and let $\mathsf{OPT} = f(\vec{o})$ denote the optimal value. Define the cost of a job j under σ^* to be $p_{\sigma^*(j)j}$. For any set $S \subseteq J$ of jobs, define the job-cost vector $\overrightarrow{P^*_S} = \overrightarrow{P^{\sigma^*}_S} := \big(p_{\sigma^*(j)j}\big)_{j\in S}$. Let $\varepsilon > 0$ be a parameter, and let $\delta = \min\{\varepsilon, 1\}$. Recall that we abbreviate $\mathsf{POS}_{m,\delta}$ to POS. We may assume that $n \geq \lceil 2(1+\delta)/\delta^3 \rceil$ as otherwise we can simply exhaustively search for the optimal assignment.

LP relaxation. As is standard, our LP has variables x_{ij} for every machine i and job j denoting if job j is assigned to machine i (or, fractionally, the extent of j assigned to machine i). Constraints (2) enforce that every job is assigned to a machine.

We are guided by Theorem 2.4, which shows that to obtain a solution whose load vector has f-norm equal to $O(\mathsf{OPT})$, it suffices to show that the Top_ℓ-norm of the load vector is $O\big(\mathsf{Top}_\ell(\vec{o})\big)$ for all $\ell \in [m]$; also, by Claim 2.6, one can focus on indices $\ell \in \mathsf{POS}$. We work with "guesses" (i.e., estimates) t_ℓ of $\vec{o}^{\downarrow}_\ell$, for all $\ell \in \mathsf{POS}$. Our LP seeks a fractional solution such that the resulting load vector has Top_ℓ-norm at most $\mathsf{Top}_\ell(\vec{o})$ for all $\ell \in \mathsf{POS}$. Using Claim 2.3 and our estimates, we encode this via the constraint $\ell t_\ell + \sum_i \big(\sum_j p_{ij}x_{ij} - t_\ell\big)^+ \leq \mathsf{Top}_\ell(\overrightarrow{t^{\mathsf{exp}}})$ for all $\ell \in \mathsf{POS}$; constraints (3), (4) linearize these. (Recall that $\overrightarrow{t^{\mathsf{exp}}} \in \mathbb{R}^m_{\geq 0}$ is defined by $t^{\mathsf{exp}}_i = t_i$ for $i \in \mathsf{POS}$, and $t^{\mathsf{exp}}_i = t_{\mathsf{prev}(i)}$ for $i \in [m] \setminus \mathsf{POS}$.)

The last set of constraints of our LP is motivated by an insight in [5]. They show that if $\overrightarrow{\mathsf{JL}^*} \in \mathbb{R}^m_{\geq 0}$ is the vector comprising the costs of the m most costly jobs under σ^*, then we have $f(\overrightarrow{\mathsf{JL}^*}) \leq \mathsf{OPT}$, which also implies that $f(\overrightarrow{P^*_S}) \leq \mathsf{OPT}$ for any set S of m jobs. Chakrabarty and Swamy [5] include this (convex) constraint directly in their convex program, and it plays a crucial role in obtaining their $(4+\varepsilon)$-approximation for MinNormLB. We also utilize this constraint, but incorporate it (loosely speaking) in our LP in a more subtle fashion. We work with guesses ζ_ℓ of $\overrightarrow{\mathsf{JL}^*}^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}$, and encode (see constraints (5)) that there are at most $\ell - 1$ jobs whose cost is larger than ζ_ℓ, for all $\ell \in \mathsf{POS}$. This can be seen as an indirect way of capturing $f(\overrightarrow{\mathsf{JL}^*}) \leq \mathsf{OPT}$, and this indirect way is crucial for us because we show that we can round a fractional solution x with only a $(1+\delta)$-factor violation of these constraints. For technical reasons, to facilitate this, we use another partial enumeration step. Let ℓ_0 be the smallest index in POS that is at least $\frac{2}{\delta^3}$. Note that $\ell_0 \leq \lceil 2(1+\delta)/\delta^3 \rceil \leq n$. We will guess the ℓ_0 most costly jobs under σ^* and their σ^*-assignments. Let $C = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{\ell_0}, j_{\ell_0})\}$ denote our guess of these jobs and their σ^*-assignments. Note that given C, we know that $\overrightarrow{\mathsf{JL}^*}^{\downarrow}_\ell$ is the ℓ-th largest value in $\{p_{ij} : (i,j) \in C\}$ for all $\ell \in [\ell_0]$, and we therefore set $\zeta_\ell = \overrightarrow{\mathsf{JL}^*}^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}$ with $\ell \leq \ell_0$. An important consequence of this enumeration step, which will be crucial in the analysis (see the proof of Lemma 4.3), is that this *fixes* the assignment of all jobs whose cost under σ^* is larger than ζ_{ℓ_0}; we encode this in our LP via constraints (6).

This yields the following LP, which is a feasibility LP depending on $\vec{t}$, $\vec{\zeta}$, and C. Throughout, we use i to index $[m]$, and j to index J.

$$(\mathsf{LP}(\vec{t},\vec{\zeta},C)) \qquad \begin{aligned} &x \geq 0, \quad \sum_i x_{ij} = 1 && \forall j \in J && (2)\\ &\ell t_\ell + \sum_i W_i^\ell \leq \mathsf{Top}_\ell(\overrightarrow{t^{\mathsf{exp}}}) && \forall \ell \in \mathsf{POS} && (3)\\ &W_i^\ell \geq 0, \quad W_i^\ell \geq \sum_j p_{ij}x_{ij} - t_\ell && \forall i \in [m],\ \ell \in \mathsf{POS} && (4)\\ &\sum_{i,j:p_{ij}>\zeta_\ell} x_{ij} \leq \ell - 1 && \forall \ell \in \mathsf{POS} && (5)\\ &x_{ij} = \begin{cases} 1 & \text{if } (i,j) \in C \\ 0 & \text{otherwise.}\end{cases} && \forall i \in [m], j \in J \text{ s.t. } p_{ij} > \zeta_{\ell_0} && (6)\end{aligned}$$

▷ Claim 4.2. $(\mathsf{LP}(\vec{t},\vec{\zeta},C))$ is feasible whenever: (i) $t_\ell \geq \vec{o}^{\downarrow}_\ell$, $\zeta_\ell \geq \overrightarrow{\mathsf{JL}^*}^{\downarrow}_\ell$ for all $\ell \in \mathsf{POS}$; and (ii) C is the correct guess of the ℓ_0 most costly jobs under σ^* and their σ^*-assignments.

Rounding algorithm. Assume that $(\mathsf{LP}(\vec{t},\vec{\zeta},C))$ is feasible and $(\overline{x},\overline{W})$ is a feasible solution. We consider the following auxiliary LP (Aux) for rounding $\overline{x}$.

$$x \geq 0, \quad \sum_i x_{ij} = 1 \ \ \forall j \in J, \quad \sum_j p_{ij}x_{ij} \leq \sum_j p_{ij}\overline{x}_{ij} \ \ \forall i \in [m], \quad \sum_{i,j:p_{ij}>\zeta_\ell} x_{ij} \leq \ell - 1 \quad \forall \ell \in \mathsf{POS}.$$

Clearly, $\overline{x}$ is a feasible solution to (Aux). We show that this can be rounded to an integer solution $\widetilde{x}$ so that $\sum_i \widetilde{x}_{ij} = 1$ for every job j, the load on each machine is at most its $\overline{x}$-load + $(1+\delta)\times$(cost of the most-costly job assigned to it), and the remaining constraints are violated by a small factor. More precisely, in the analysis, we prove the following result.

► **Lemma 4.3.** *We can round $\overline{x}$ efficiently to an integer solution $\widetilde{x}$ satisfying the following.*
(a) $\widetilde{x}_{ij} = \overline{x}_{ij}$ *for all i, j such that* $\overline{x}_{ij} \in \{0,1\}$;
(b) $\sum_i \widetilde{x}_{ij} = 1$ *for each job j;*
(c) $\sum_j p_{ij}\widetilde{x}_{ij} \leq \sum_i p_{ij}\overline{x}_{ij} + (1+\delta)\max_{j:\widetilde{x}_{ij}=1} p_{ij}$ *for every machine i;*
(d) $\sum_{i,j:p_{ij}>\zeta_\ell} \widetilde{x}_{ij} \leq \ell - 1$ *for all* $\ell \in \mathsf{POS}$ *with* $\ell \leq \ell_0$; *and*
(e) $\sum_{i,j:p_{ij}>\zeta_\ell} \widetilde{x}_{ij} \leq (1+\delta)\ell - 1$ *for all* $\ell \in \mathsf{POS}$ *with* $\ell > \ell_0$.

We apply the above lemma, and due to Lemma 4.3 (a), the integer solution $\widetilde{x}$ yields an assignment $\widetilde{\sigma}$ of jobs to machines; we return this assignment.

Analysis. We assume again that f is normalized. Lemma 4.3 is the main technical result that we need to prove. Its proof utilizes an *iterative-rounding* result of independent interest (Theorem 4.6) that is very similar to Corollary 11 in [19], wherein iterative rounding is used to round a point that lies in the base polytope of one matroid $\mathcal{M}_0$ and satisfies various other matroid-independence and knapsack constraints, to a basis of $\mathcal{M}_0$ that is "approximately independent" in the other matroids and violates the knapsack constraints by a certain additive factor. In (Aux), the job assignment constraints correspond to the base-polytope constraints (of a partition matroid), and the remaining constraints correspond to knapsack constraints. First, we establish that, assuming Lemma 4.3, we obtain the stated guarantee.

Suppose that we have the correct set C, and that $\vec{t}$ and $\vec{\zeta}$ are "good" estimates of $(\vec{o}^{\downarrow}_{\ell})_{\ell \in \mathsf{POS}}$ and $(\overrightarrow{\mathsf{JL}^*}^{\downarrow}_{\ell})_{\ell \in \mathsf{POS}}$ respectively (we make this precise later). We analyze the load vector $\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}}$ by considering two vectors $L = (L_i)_{i \in [m]}$ and $P = (P_i)_{i \in [m]}$. For each $i \in [m]$, define $L_i = \sum_j p_{ij}\overline{x}_{ij}$, and $P_i = \max_{j:\widetilde{\sigma}(j)=i} p_{ij}$; note that $P_i = 0$ if there is no job assigned by $\widetilde{\sigma}$ to machine i. By Lemma 4.3 (c), we have that $\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}} \leq L + (1+\delta)P$, and we bound both $f(L)$ and $f(P)$ by roughly OPT. Constraints (3), (4) of our LP immediately allow us to bound $\mathsf{Top}_{\ell}(L)$ by $\mathsf{Top}_{\ell}(\overrightarrow{t^{\mathsf{exp}}})$ for all $\ell \in \mathsf{POS}$, which suffices since $\vec{t}$ well estimates $(\vec{o}^{\downarrow}_{\ell})_{\ell \in \mathsf{POS}}$ (see Lemma 4.4). For the vector P, parts (d) and (e) of Lemma 4.3 yield the desired bound on $f(P)$, due to Lemma 2.8 (c) (see Lemma 4.5). Before proving Lemmas 4.4 and 4.5, we justify, and make precise, the assumption that we have the correct set C (specifying the ℓ_0 most costly jobs under σ^* and their σ^*-assignments), and good estimates $\vec{t}$, $\overrightarrow{\mathsf{JL}^*}$. There are at most $\binom{n}{\ell_0} \cdot m^{\ell_0}$ choices for the set C. As noted earlier, for all $\ell \in \mathsf{POS}$ with $\ell \leq \ell_0$, this then fixes $\zeta_{\ell} = \zeta^*_{\ell} = \overrightarrow{\mathsf{JL}^*}^{\downarrow}_{\ell}$ to be the ℓ-th largest entry in $\{p_{ij} : (i,j) \in C\}$. Using Claim 2.1, if we consider the assignment σ^{sum} that minimizes the sum of the machine loads – so $\sigma^{\mathsf{sum}}(j) = \operatorname{argmin}_i p_{ij}$ for all jobs j – and set $\mathsf{UB} := \sum_i \mathsf{load}_{\sigma^{\mathsf{sum}}}(i)$, then we obtain that $\frac{\mathsf{UB}}{m} \leq \mathsf{OPT} \leq \mathsf{UB}$, and so $\vec{o}_1, \overrightarrow{\mathsf{JL}^*}_1 \leq \mathsf{UB}$. Set $\kappa = \varepsilon\mathsf{UB}/m^2$. Let $\mathsf{POS}_{>} := \{\ell \in \mathsf{POS} : \ell > \ell_0\}$. So using Lemma 2.9 (a) (and since $|\mathsf{POS}| = O(\log m/\delta)$), we can identify polynomial-size sets containing vectors $\vec{t} = \vec{t^*}$ and $(\zeta_{\ell})_{\ell \in \mathsf{POS}_>} = (\zeta^*_{\ell})_{\ell \in \mathsf{POS}_>}$ such that:

(1) $\vec{o}^{\downarrow}_{\ell} \leq t^*_{\ell} \leq (1+\varepsilon)\vec{o}^{\downarrow}_{\ell} + \kappa$ for all $\ell \in \mathsf{POS}$, and $\vec{t^*}$ is a non-increasing vector; and

(2) $\overrightarrow{\mathsf{JL}^*}^{\downarrow}_{\ell} \leq \zeta^*_{\ell} \leq (1+\varepsilon)\overrightarrow{\mathsf{JL}^*}^{\downarrow}_{\ell} + \kappa$ for all $\ell \in \mathsf{POS}_>$, and $\zeta^* = (\zeta^*_{\ell})_{\ell \in \mathsf{POS}}$ is non-increasing.

For (1), we take $\mathcal{L} = \mathsf{POS}$, $\mathsf{ub} = \mathsf{UB}$, and κ as above; for (2), we take $\mathcal{L} = \mathsf{POS}_>$, $\mathsf{ub} = \overrightarrow{\mathsf{JL}^*}_{\ell_0}$, and κ as above.) Recall that $\delta = \min\{\varepsilon, 1\}$.

▶ **Lemma 4.4.** *We have* $f(L) \leq (1+\delta)(1+\varepsilon)\mathsf{OPT} + m\kappa \leq (1+\delta+3\varepsilon)\mathsf{OPT}$.

Proof. We show that $\mathsf{Top}_i(L) \leq \mathsf{Top}_i(\overrightarrow{t^{*\mathsf{exp}}})$ for *all* $i \in [m]$. For any $\ell \in \mathsf{POS}$, this follows quite directly from constraints (3), (4) of our LP $(\mathsf{LP}(\vec{t},\vec{\zeta},C))$: we have $\mathsf{Top}_{\ell}(L) \leq \ell t^*_{\ell} + \sum_{i' \in [m]}(L_{i'} - t^*_{\ell})^+ \leq \mathsf{Top}_{\ell}(\overrightarrow{t^{*\mathsf{exp}}})$. For any $i \in [m] \setminus \mathsf{POS}$, taking $\ell = \mathsf{prev}(i)$, this then implies that

$$\mathsf{Top}_i(L) \leq i t^*_{\ell} + \sum_{i' \in [m]} (L_{i'} - t^*_{\ell})^+ \leq \mathsf{Top}_{\ell}(\overrightarrow{t^{*\mathsf{exp}}}) + (i-\ell)t^*_{\ell} = \mathsf{Top}_i(\overrightarrow{t^{*\mathsf{exp}}}),$$

where the final equality follows because ${t^*}^{\mathsf{exp}}_{i'} = t^*_{\ell}$ for all $i' \in \{\ell, \ell+1, \ldots, \mathsf{next}(\ell)-1\}$. Hence, by Theorem 2.4, we have that $f(L) \leq f(\overrightarrow{t^{*\mathsf{exp}}})$.

Since $t^*_{\ell} \leq (1+\varepsilon)\vec{o}^{\downarrow}_{\ell} + \kappa$ for all $\ell \in \mathsf{POS}$, by Lemma 2.8 (b), taking $u = \vec{o}$ and $v = \vec{t^*}$, we obtain that $f(\overrightarrow{t^{*\mathsf{exp}}}) \leq (1+\delta)(1+\varepsilon)f(\vec{o}) + m\kappa$. The final inequality in the lemma statement follows because $m\kappa \leq \varepsilon\mathsf{OPT}$ and $(1+\delta)(1+\varepsilon) \leq (1+\delta+2\varepsilon)$ as $\delta \leq 1$. ◀

▶ **Lemma 4.5.** *We have* $f(P) \leq (1+4\delta+5\varepsilon)f(\overrightarrow{\mathsf{JL}^*}) + 5m\kappa \leq (1+4\delta+10\varepsilon)\mathsf{OPT}$.

Proof. We apply Lemma 2.8 (c), taking $u = \overrightarrow{\mathsf{JL}^*}$, $\alpha = P$, and $v = (\zeta^*_{\ell})_{\ell \in \mathsf{POS}}$. The conditions in Lemma 2.8 (c) hold due to parts (d) and (e) of Lemma 4.3. This yields that $f(P) \leq (1+4\delta)(1+\varepsilon)f(\overrightarrow{\mathsf{JL}^*}) + 5m\kappa$. The lemma follows since $\delta \leq 1$, $m\kappa \leq \varepsilon\mathsf{OPT}$, and $f(\overrightarrow{\mathsf{JL}^*}) \leq \mathsf{OPT}$. ◀

Proof of Theorem 4.1. Recall that $\widetilde{\sigma}$ is the assignment returned by the algorithm. Lemmas 4.4 and 4.5 together with the fact that $\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}} \leq L + (1+\delta)P$ immediately yield that

$$f(\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}}) \leq (1+\delta+3\varepsilon)\mathsf{OPT} + (1+\delta)(1+4\delta+10\varepsilon)\mathsf{OPT} \leq (2+10\delta+23\varepsilon)\mathsf{OPT}. \quad ◀$$

Proof of Lemma 4.3

We reproduce the system (Aux) below for easy reference.

$$\text{(Aux)}\qquad \begin{aligned} & x \geq 0, \quad \sum_i x_{ij} = 1 && \forall j \in J && (7)\\ & \sum_j p_{ij}x_{ij} \leq \sum_j p_{ij}\overline{x}_{ij} && \forall i \in [m] && (8)\\ & \sum_{i,j:p_{ij}>\zeta_\ell} x_{ij} \leq \ell - 1 && \forall \ell \in \mathsf{POS} && (9) \end{aligned}$$

We utilize the following iterative-rounding result that is quite similar to Corollary 11 in [19].

▶ **Theorem 4.6** (Similar to Corollary 11 in [19]). *Let $\mathcal{M}_s = (N_s, \mathcal{I}_s)$, for $s = 0, \ldots, k$ be $(k+1)$ matroids, where each N_s is a subset of $N := N_0$. Let r_s be the rank function of matroid $\mathcal{M}_s$, for $s = 0, \ldots, k$. Let $w \in \mathbb{R}^N$, $A \in \mathbb{R}^{d\times N}_{\geq 0}$, $b \in \mathbb{R}^d_{\geq 0}$. Consider the following LP.*

$$\begin{aligned} \max \quad & w^T x && && (\mathrm{LP}_{\mathrm{matkn}})\\ s.t. \quad & x(N) = r_0(N), \quad x(T) \leq r_0(T) && \forall T \subseteq N\\ & x(T) \leq r_s(T) && \forall s = 1, \ldots, k,\ \forall T \subseteq N_i\\ & Ax \leq b, \quad x \geq 0. \end{aligned}$$

Let $q_1, \ldots, q_k \geq 1$ be integers, and $\mu_1, \ldots, \mu_d \in \mathbb{R}_{\geq 0}$ be such that:

$$\sum_{s\in[k]:e\in N_s} \frac{1}{q_s} + \sum_{s\in[d]:A_{se}>0} \frac{1}{\mu_s} \leq 1 \qquad \forall e \in N. \tag{10}$$

If $(\mathrm{LP}_{\mathrm{matkn}})$ *is feasible, then we can efficiently obtain a basis R of $\mathcal{M}_0$ such that:*

(i) *$|T| \leq q_s r_s(T)$ for all $s \in [k]$ and $T \subseteq R \cap N_s$;*

(ii) *$\sum_{e\in R} A_{se} \leq b_s + \mu_s \cdot \max_{e\in R} A_{se}$ for all $s \in [d]$; and*

(iii) *$w(R) \geq \mathsf{OPT}_{LP_{\mathrm{matkn}}}$.*

▶ Remark 4.7. The statement of Theorem 4.6 differs from that of Corollary 11 in [19] in three respects. In [19]: (1) $\mu_1, \ldots, \mu_d$ are assumed to be positive integers (but their proof does not need the integrality requirement); (2) condition (10) is replaced by the *weaker* condition, $\sum_{s\in[k]:e\in N_s} \frac{1}{q_s} + \sum_{s\in[d]} \frac{A_{se}}{(\max_{e\in N} A_{se})\mu_s} \leq 1$ for all $e \in N$; and (3) one obtains the slightly *weaker* additive error of $\mu_s \cdot \max_{e\in N} A_{se}$ in (ii) for the knapsack constraint for index $s \in [d]$. The proof of Corollary 11 in [19] is however easily adapted to yield the above statement; we include the proof of Theorem 4.6 in the full version.

For our purposes, we only need a simpler version of Theorem 4.6, where we have only one matroid $\mathcal{M}_0$ and other knapsack constraints. We include the more general statement above (with its subtly stronger guarantee for the knapsack constraints, which turns out to be crucial for us; see Remark 4.8) because we believe that this is of independent interest.

Before delving into the details of how we apply Theorem 4.6 to prove our lemma, we briefly discuss the main idea, which will also elucidate why we need the partial enumeration step of "guessing" the set C specifying the ℓ_0 most costly jobs under σ^* and their σ^*-assignments.

We can cast (Aux) as an instance of $(\mathrm{LP}_{\mathrm{matkn}})$ as follows. Let $U = [m] \times J$. The weight vector w is irrelevant. The matroid $\mathcal{M}_0$ is the partition matroid over U encoding that each job j is assigned to at most one machine. It is immediate that (7) shows that

$\overline{x}$ lies in the base polytope of $\mathcal{M}_0$. Constraints (8) correspond to knapsack constraints in ($\mathrm{LP}_{\mathrm{matkn}}$). Constraints (9) can be incorporated into ($\mathrm{LP}_{\mathrm{matkn}}$) in three ways: (i) as matroid-independence constraints in the (single) laminar matroid formed by the nested family of sets $\{(i,j) \in U : p_{ij} > \zeta_\ell\}_{\ell \in \mathsf{POS}}$; (ii) as multiple matroid-independence constraints, where for each $\ell \in \mathsf{POS}$, we have a matroid encoding the cardinality bound from $\{(i,j) \in U : p_{ij} > \zeta_\ell\}$; or (iii) as additional knapsack constraints.

But utilizing Theorem 4.6 directly on the above system does not quite work for us. To elaborate, in order to obtain the guarantee in 4.3 (c), we need to set the μ_is for the knapsack constraints (8) to $(1+\delta)$. But then if we view (9) as matroid independence constraints (in one laminar matroid, or multiple matroids), we need to set the q_s value for the corresponding matroid(s) to be at least 2 in order to satisfy (10) resulting in (at least) a multiplicative factor-2 violation of constraints (9), which is too large a violation. Suppose we incorporate (9) as knapsack constraints and, for some constant ρ, take $\mu_\ell = \rho\ell$ for constraint (9) for index ℓ. But to satisfy (10), we need ρ to be at least $\left(1-\frac{1}{1+\delta}\right)^{-1} \cdot \sum_{\ell' \in \mathsf{POS}:\ell'>1} \frac{1}{\ell'} = \Omega\left(\frac{1}{\delta^2}\right)$, which again yields too large a violation of (9). (A more accurate estimate of $\sum_{\ell' \in \mathsf{POS}:\ell'>1}$ is $\ln(\lceil 1/\delta \rceil) + 1 + \delta$, but this still requires ρ to be large.)

However, a tweak of the latter approach does work, by noting that if x_{ij} first appears in constraint (9) for index $\ell \in \mathsf{POS}$, then to satisfy (10), we really need that $\rho \geq \left(1-\frac{1}{1+\delta}\right)^{-1} \cdot \sum_{\ell' \in \mathsf{POS}:\ell' \geq \ell} \frac{1}{\ell'}$; the RHS is at most $\frac{(1+\delta)^2}{\delta^2 \ell}$, which is at most δ for *large enough* ℓ. *This is precisely where the enumeration of C turns out to be key, since it allows us to eliminate constraints* (9) *for* $\ell \leq \ell_0$. Recall that C specifies the ℓ_0 most-costly jobs under σ^* and their σ^*-assignments, where ℓ_0 is the smallest index in POS that is at least $\frac{2}{\delta^3}$. Due to this enumeration, $\overline{x}$ is integral whenever $p_{ij} > \zeta_\ell$, and so if we drop integral variables and corresponding constraints from (Aux), we are left with constraints (9) for indices $\ell > \ell_0$ in POS. Now we can take $\mu_\ell = \delta\ell$ for all $\ell \in \mathsf{POS}$, $\ell > \ell_0$, and these values satisfy (10), since each $\ell > \ell_0$ is large.

We now describe precisely how we utilize Theorem 4.6 to prove our lemma. We apply Theorem 4.6 to (Aux) after getting rid of all integral variables, and modifying or dropping constraints to take into account the integral variables. Recall that $U = [m] \times J$. Let $E = \{(i,j) \in U : \overline{x}_{ij} \in (0,1)\}$. We retain only the variables in E, and only these will be rounded, so this takes care of part (a) of the lemma. We drop any constraint that contains only integral variables. Thus, the POS-constraints (9) for all $\ell \in \mathsf{POS}$, $\ell \leq \ell_0$ are dropped, and so part (d) holds. Let J' be the set of jobs corresponding to the remaining constraints (7); note that all jobs in J' are completely assigned by (the variables in) E, and the remaining jobs are completely assigned by the integral variables. Let $I \subseteq [m]$ index the machine-constraints (8) that remain, and $I^{\mathsf{POS}} \subseteq \{\ell \in \mathsf{POS} : \ell > \ell_0\}$ index the POS-constraints (9) that remain. This yields the following system of constraints.

$$
\text{(P')} \quad \begin{cases}
\displaystyle\sum_{i:(i,j)\in E} x_{ij} = 1 & \forall j \in J' \\
\displaystyle\sum_{j:(i,j)\in E} p_{ij}x_{ij} \leq \sum_j p_{ij}\overline{x}_{ij} - \sum_{j:\overline{x}_{ij}=1} p_{ij} & \forall i \in I \\
\displaystyle\sum_{(i,j)\in E:p_{ij}>\zeta_\ell} x_{ij} \leq \ell - 1 - \left|\{(i,j) : p_{ij} > \zeta_\ell, \overline{x}_{ij} = 1\}\right| & \forall \ell \in I^{\mathsf{POS}} \\
x \geq 0. &
\end{cases}
$$

Clearly, $(\overline{x}_{ij})_{(i,j)\in E}$ is a feasible solution to (P'). We apply Theorem 4.6 to (P'), taking $\mathcal{M}_0$ to be the partition matroid now with ground set E encoding that every job in J' is assigned to at most one machine, and treating both the machine constraints indexed by I, and the POS-constraints indexed by I^{POS} as knapsack constraints. The weight vector w is irrelevant. We take $\mu_i = (1+\delta)$ for all $i \in I$, and $\mu_\ell = \delta\ell$ for all $\ell \in I^{\mathsf{POS}}$, and we claim that this satisfies condition (10). To see this, consider any $(i,j) \in E$. Let ℓ be the smallest index in I^{POS} whose POS-constraint contains the variable x_{ij}. Then, x_{ij} appears possibly in the machine constraint for machine i, and the POS-constraints for indices $\ell' \in \mathsf{POS}$, $\ell' \geq \ell$. We have

$$\sum_{\ell'\in I^{\mathsf{POS}}:\ell'\geq\ell}\frac{1}{\mu_{\ell'}} \leq \sum_{\ell'\in\mathsf{POS}:\ell'\geq\ell}\frac{1}{\delta\ell'} \leq \frac{1}{\delta\ell}\cdot\sum_{k\geq 0}(1+\delta)^{-k} = \frac{1}{\delta\ell}\cdot\frac{1+\delta}{\delta} \leq \frac{1}{\delta^2\ell_0} \leq \frac{\delta}{2}.$$

Since $\frac{1}{1+\delta} + \frac{\delta}{2} \leq 1$, it follows that (10) is satisfied for (i,j).

Finally, we argue that applying Theorem 4.6 with the above parameters yields the lemma. Let $\widetilde{x}$ be the solution obtained by concatenating the integral portion of $\overline{x}$, and the integer solution returned by Theorem 4.6. Part (b) holds because we return a basis of $\mathcal{M}_0$, and the jobs not in J' are completely assigned by the integral portion of $\overline{x}$. Parts (c) and (e) follow directly from our choice of the μ values. ◀

▶ Remark 4.8. We note that the subtly stronger guarantee obtained for the knapsack constraint for index s, in Theorem 4.6, versus Corollary 11 in [19] – i.e., an additive error of $\mu_s \cdot \max_{e\in R} A_{se}$ versus an additive error of $\mu_s \cdot \max_{e\in N} A_{se}$ – is crucial for us. The weaker guarantee would yield, after dropping integral (and hence 0-valued) variables, that the load on each machine i is at most $\sum_j p_{ij}\overline{x}_{ij} + (1+\delta)\max_{j:\overline{x}_{ij}>0} p_{ij}$. But it is hard to obtain a bound on the f-norm of the vector $\{\max_{j:\overline{x}_{ij}>0} p_{ij}\}_{i\in[m]}$, since multiple coordinates here could correspond to the same job, and therefore constraints (9) do not help.

5 A PTAS for MinNormLB on identical machines

We now consider the oft-studied and important special case where all machines are identical – that is, we have $p_{ij} = p_j$ for all jobs j and machines i – and devise a PTAS in this setting.

▶ **Theorem 5.1.** *There is a PTAS for* MinNormLB *on identical machines.*

The above result substantially generalizes the well-known PTAS by Hochbaum and Shmoys for minimizing makespan (i.e., the special case of ℓ_∞ norm) on identical machines [11]. We utilize some of their insights, but need to combine them with various novel ideas to handle the substantial richness of arbitrary monotone, symmetric norms. In the full version, we show that our PTAS extends to a further generalization of MinNormLB, wherein each machine incurs a *cost* given by a "bounded-growth" convex, non-decreasing function μ of its load, and we seek to minimize the f-norm of the *machine-cost vector.* We also observe that Graham's list-scheduling rule – considering jobs in any sequence, schedule the next job on the currently least-loaded machine – yields a simple 2-approximation algorithm.

▶ **Theorem 5.2.** *Graham's list-scheduling rule yields a 2-approximation for* MinNormLB *on identical machines.*

We defer the proof of Theorem 5.2 to the full version, and discuss the PTAS for MinNormLB here. It is useful to first recall how the PTAS in [11] for makespan works. Their algorithm considers a "guess", say $t_1 \geq \max_j p_j$, of the optimal makespan, *opt*. We classify jobs as large or small, based on whether $p_j \geq \varepsilon t_1$ (large jobs) or $p_j < \varepsilon t_1$ (small jobs). A key insight is that since the large jobs have $p_j \in [\varepsilon t_1, t_1]$, rounding their p_js to the nearest power of $(1+\varepsilon)$ yields $O\big(\log_{1+\varepsilon}\frac{1}{\varepsilon}\big)$ distinct rounded job sizes, and a constant (depending on ε) number of distinct possibilities, called *configurations*, for assigning large jobs to a machine so that its rounded load is at most $(1+\varepsilon)t_1$. One can write an *integer program*, with a (ε-dependent) constant number of variables, to determine which configurations are used on the machines, which can be solved in time $\text{poly}(m,n)$ for any fixed $\varepsilon > 0$ using the algorithm of Lenstra [18]. If $opt \le t_1$, then this yields an assignment of large jobs having makespan at most $(1+\varepsilon)t_1$. Finally, one can argue that either the small jobs can be packed (arbitrarily) on the machines while maintaining a makespan of at most $(1+\varepsilon)t_1$, or we have $opt > t_1$.

Now consider MinNormLB. We may assume that $n > m$ as otherwise it is easy to see that an optimal solution is to assign each job to a separate machine. Clearly, we may also assume that $p_j > 0$ for all jobs j. Recall that $\vec{o}$ is an optimal load vector, and $\mathsf{OPT} = f(\vec{o})$. As before, we assume that f is normalized. Since all machines are identical, it will be convenient to re-index machines so that $\vec{o} = \vec{o}^{\downarrow}$. Since we now need to control all Top_ℓ norms, following a common theme, it is natural that we now work with guesses t_ℓ of o_ℓ (which is $\vec{o}^{\downarrow}_\ell$) for all $\ell \in \mathsf{POS}$. The chief issue that arises is: *how do we now define large and small jobs given the multiple t_ℓs?* Suppose we choose a threshold of εt_ℓ, for some $\ell \in \mathsf{POS}$, for this purpose. For indices $\ell' < \ell$, the problem then is that we need to consider all jobs with $p_j \in [\varepsilon t_\ell, t_{\ell'}]$; this would yield a *non-constant number* of job types, and hence a non-constant number of configurations for machines $i < \ell$. Machines $i > \ell$ also present a problem: the packing of small jobs may cause us to exceed the target load of o_i (or rather t_i^{exp}) by εt_ℓ, which can create too large an error compared to the target load.

We circumvent these issues as follows. We select a *specific* optimal solution. Given distinct vectors $u, v \in \mathbb{R}^m$, we say that u is *lexicographically smaller* than v, denoted $u \prec_{\mathsf{lex}} v$ if there exists some index $i \in [m]$ such that $u_{i'} = v_{i'}$ for all $i' = 1, \dots, i-1$, and $u_i < v_i$. Let $\vec{o}^{\downarrow}$ be the *lexicographically smallest* sorted load-vector among all optimal solutions, i.e., $\vec{o}^{\downarrow} \prec_{\mathsf{lex}} (\overrightarrow{\mathsf{load}_\sigma})^{\downarrow}$ for every optimal solution σ. As before, we may assume that $\vec{o} = \vec{o}^{\downarrow}$. Let σ^* be an optimal solution yielding the load-vector $\vec{o}$. Let i^* be the smallest index such that machine i^* has at least two jobs assigned to it under σ^*; this is well defined since $n > m$.

▷ Claim 5.3. Let $\vec{o}$, σ^* and i^* be as defined above. Then we have $o_i \ge o_{i^*}/2$ for all $i \ge i^*$.

In addition to our guess $\vec{t}$ of $(o_\ell)_{\ell \in \mathsf{POS}}$, we now also guess i^* and the load o_{i^*}, and use εo_{i^*} as the threshold demarcating large and small jobs. The small jobs are now indeed small compared to the target load of o_i for $i \ge i^*$. Also, since machines $i < i^*$ only have one job assigned to them, we can infer that these machines are assigned the $i^* - 1$ largest jobs.

We describe our algorithm below, and conclude with its analysis.

▶ **Algorithm 1 (PTAS for MinNormLB on identical machines).**
Input: a non-increasing vector $\vec{t} \in \mathbb{R}_{\ge 0}^{\mathsf{POS}}$, an index $i^* \in [m]$, $\theta \in \mathbb{R}_{\ge 0}$ intended to be a good overestimate of o_{i^*}, and a parameter $0 < \varepsilon \le 1$. We set $\delta = \varepsilon$ in the definition of $\mathsf{POS} = \mathsf{POS}_{m,\delta}$.
Output: an assignment $\widetilde{\sigma}$ such that $(\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}})^{\downarrow}_\ell \le (1+\varepsilon)t_\ell + \varepsilon\theta$ for all $\ell \in \mathsf{POS}$, or that $(\vec{t}, i^*, \theta)$ is an invalid guess.

P1. Perform the following checks, and if any of these fail, then declare that $(\vec{t}, i^*, \theta)$ is invalid and return. Define $t_{m+1} := 0$ for notational convenience.
  (a) If $i^* \in \mathsf{POS}$, then check that $\theta = t_{i^*}$, otherwise check that $t_{\mathsf{next}(i^*)} \le \theta \le t_{\mathsf{prev}(i^*)}$.
  (b) For all $\ell \in \mathsf{POS}$, $\ell \le i^*$, check that there are at most $\ell - 1$ jobs with $p_j > t_\ell$.
  (c) Check that there are at most $i^* - 1$ jobs with $p_j > \theta$.

P2. Assign the $i^* - 1$ largest jobs to machines $1, \dots, i^* - 1$, assigning the largest job to machine 1, the second-largest job to machine 2, and so on. Let J' be the remaining set of jobs. Note that check (c) in step P1 implies that $p_j \le \theta$ for all $j \in J'$.

P3. Let $J'_L := \{j \in J' : p_j \geq \varepsilon\theta\}$, and $J'_S := \{j \in J' : p_j < \varepsilon\theta\}$. We refer to the jobs in J'_L and J'_S as large and small jobs respectively.

P4. Round the processing time of each large job to the nearest power of $(1+\varepsilon)$, i.e., for each $j \in J'_L$, round its processing time to $p'_j := (1+\varepsilon)^{\lceil \log_{1+\varepsilon} p_j \rceil}$. Let $R = \{p'_j : j \in J'_L\}$ be the distinct rounded processing times, and $N = |R|$; note that $N \leq 1 + \log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right)$. For a budget $B \geq 0$, define $\mathcal{C}(B) := \left\{\nu \in \mathbb{Z}^R_{\geq 0} \setminus \{\vec{0}\} : \sum_{p \in R} \nu_p \cdot p \leq B\right\}$ as the set of non-trivial *configurations* that yield a total rounded load of at most B. (Observe that $\mathcal{C}(0) = \emptyset$.)

Recall that $\overrightarrow{t^{\mathsf{exp}}} \in \mathbb{R}^m_{\geq 0}$ is the expansion of $\vec{t}$, defined by $t^{\mathsf{exp}}_i = t_i$ for $i \in \mathsf{POS}$, and $t^{\mathsf{exp}}_i = t_{\mathsf{prev}(i)}$ for $i \in [m] \setminus \mathsf{POS}$. For notational convenience, define $t^{\mathsf{exp}}_{m+1} := 0$. For $i \in \{i^*, \dots, m+1\}$, define its load budget to be $B_i = (1+\varepsilon)\min\{\theta, t^{\mathsf{exp}}_i\}$. (Note that $B_{m+1} = 0$.) For any $0 \leq B \leq B_{i^*}$, any $\nu \in \mathcal{C}(B)$ and $p \in R$, we have $\nu_p \leq B/p \leq \frac{1+\varepsilon}{\varepsilon}$; so $|\mathcal{C}(B)| \leq \left(\frac{1}{\varepsilon}+2\right)^N$, and we can enumerate the configurations in $\mathcal{C}(B)$ in polynomial time. Assign the large jobs by using the algorithm of Lenstra [18] to solve the integer program (Config-IP), wherein the integer variable x_ν specifies the number of machines in $\{i^*, \dots, m\}$ for which configuration $\nu \in \mathcal{C}(B_{i^*})$ is used.

$$\text{(Config-IP)} \qquad \begin{aligned} &\sum_{\nu \in \mathcal{C}(B_{i^*})} \nu_p \cdot x_\nu = \left|\{j \in J'_L : p'_j = p\}\right| && \forall p \in R && (11) \\ &\sum_{\nu \in \mathcal{C}(B_{i^*}) \setminus \mathcal{C}(B_\ell)} x_\nu \leq \ell - i^* && \forall \ell \in \mathsf{POS} \cup \{m+1\} : \ell > i^* && (12) \\ &x_\nu \in \mathbb{Z}_{\geq 0} && \forall \nu \in \mathcal{C}(B_{i^*}) && (13) \end{aligned}$$

If (Config-IP) is infeasible, declare that $(\vec{t}, i^*, \theta)$ is invalid and return. Otherwise, suppose that $\widetilde{x}$ is a feasible solution to (Config-IP). Obtain an assignment of the large jobs to machines $i^*, i^*+1, \dots, m$ from $\widetilde{x}$ as follows. Let $\widetilde{\mathcal{C}}$ be the *multiset* of configurations where we have $\widetilde{x}_\nu$ copies of each configuration $\nu \in \operatorname{supp}(\widetilde{x})$. We map each $\nu \in \widetilde{\mathcal{C}}$ to a disjoint set A_ν of large jobs, comprising ν_p large jobs with $p'_j = p$ for every $p \in R$, so that $(A_\nu)_{\nu \in \widetilde{\mathcal{C}}}$ forms a partition of J'_L. Constraint (11) shows that this is always possible.

We go over the configurations $\nu \in \widetilde{\mathcal{C}}$ in non-increasing order of their load, $\sum_{p \in S} \nu_p \cdot p$, and assign one configuration each to machines $i^*, i^*+1, \dots, i^* + |\widetilde{\mathcal{C}}| - 1$ in this order, where by assigning a configuration ν to a machine i, we mean that we assign the jobs in A_ν to machine i. Note that $|\widetilde{\mathcal{C}}| = \sum_{\nu \in \mathcal{C}(B_{i^*})} \widetilde{x}_\nu \leq m + 1 - i^*$ due to constraint (12) for index $m+1$.

P5. We assign the small jobs to the machines in $\{i^*, \dots, m\}$ arbitrarily while ensuring that the total actual load (i.e., under the p_j processing times) on each machine i is at most $B_i + \varepsilon\theta$. If we are unable to assign all small jobs this way, then we declare that $(\vec{t}, i^*, \theta)$ is invalid, and return.

P6. Return the assignment $\widetilde{\sigma}$ computed in steps P2, P4, and P5.

Analysis. Recall that σ^* is the optimal solution yielding the sorted load vector $\vec{o}^{\downarrow} = \vec{o}$. We first assume that $(\vec{t}, i^*, \theta)$ is such that $\vec{t} \geq (o_\ell)_{\ell \in \mathsf{POS}}$, we have the right i^*, $\theta \geq o_{i^*}$, and $(\vec{t}, i^*, \theta)$ passes the check in step P1(a). Under these assumptions, we show that the algorithm returns an assignment $\widetilde{\sigma}$, and $\overrightarrow{\mathsf{load}}^{\downarrow}_{\widetilde{\sigma}}$ is "close" to $\overrightarrow{t^{\mathsf{exp}}}$. This will then imply Theorem 5.1, since we can find in polytime $(\overrightarrow{t^*}, i^*, \theta^*)$ satisfying the above properties, and such that $t^*_\ell \leq (1+\varepsilon)o_\ell$ for all $\ell \in [m]$, and $\theta^* \leq (1+\varepsilon)o_{i^*}$.

We begin by observing in Claim 5.4 that (under the above assumptions), $(\vec{t}, i^*, \theta)$ passes checks (b) and (c) in step P1, and that step P2 is valid. Then, in Lemmas 5.5 and 5.6, we argue that steps P4 and P5 are successful.

▷ Claim 5.4. (i) $(\vec{t}, i^*, \theta)$ passes checks (b) and (c) in step P1. (ii) The optimal solution σ^* assigns the $i^* - 1$ largest jobs to machines $1, \dots, i^* - 1$, assigning one job per machine.

▶ **Lemma 5.5.** *Step P4 successfully returns an assignment of large jobs such that every machine $i \in \{i^*, \dots, m\}$ has (rounded and actual) load at most B_i.*

Proof. Consider the following solution to (Config-IP). Each machine $i \in \{i^*, \dots, m\}$ that has jobs assigned to it under σ^* naturally maps to a corresponding configuration ν, where ν_p is the number of large jobs assigned to i with $p'_j = p$. We have

$$\sum_{p\in R} \nu_p \cdot p = \sum_{j \in J'_L : \sigma^*(j)=i} p'_j \le (1+\varepsilon) \sum_{j:\sigma^*(j)=i} p_j = (1+\varepsilon) o_i \le (1+\varepsilon)\min\{\theta, t_i^{\mathsf{exp}}\} = B_i$$

where the first inequality is because $p'_j \le (1+\varepsilon)p_j$ for all $j \in J'_L$. Thus, each machine $i \in \{i^*, \dots, m\}$ maps to a configuration in $\mathcal{C}(B_i) \subseteq \mathcal{C}(B_{i^*})$. Let $\hat{x} \in \mathbb{Z}_{\ge 0}^{\mathcal{C}(B_{i^*})}$ be the integral vector, where $\hat{x}_\nu$ is the number of machines that map to configuration ν, for each $\nu \in \mathcal{C}(B_{i^*})$. By construction, it is easy to see that constraints (11) are satisfied. It is also clear that constraint (12) holds for index $\ell = m+1$. So consider constraint (12) for some index $\ell \in \mathsf{POS}$, $\ell > i^*$. A configuration ν with $\hat{x}_\nu > 0$ whose load is larger than B_ℓ corresponds to a machine in $\{i^*, \dots, m\}$ whose actual load is larger than $B_\ell/(1+\varepsilon) \ge o_\ell$. There are at most $\ell - i^*$ such machines, so (12) holds for index ℓ.

Since (Config-IP) is feasible, it follows that we return an assignment of large jobs to machines $i^*, i^*+1, \dots, m$ where every machine has rounded (and hence, actual) load at most B_{i^*}. Consider a machine $i \in \{i^*, \dots, m\}$ that is assigned a configuration. Let $\ell = i$ if $i \in \mathsf{POS} \cup \{i^*\}$, and $\ell = \max\{i^*, \mathsf{prev}(i)\}$ otherwise; note that $B_i = B_\ell$. If $\ell = i^*$, then $B_i = B_{i^*}$, and there is nothing left to prove. So suppose $\ell \ne i^*$. Then $\ell \in \mathsf{POS}$, $\ell > i^*$. By constraint (12) there are at most $\ell - i^* \le i - i^*$ configurations in $\widetilde{\mathcal{C}}$ with load larger than B_ℓ. By the way in which we assign configurations to machines, it follows that machine i is *not* assigned one of these configurations. Therefore, the total rounded (and hence, actual) load on machine i is at most $B_\ell = B_i$. ◀

▶ **Lemma 5.6.** *Step P5 successfully assigns all small jobs. At the end of this step every machine $i \in \{i^*, \dots, m\}$ has total actual load at most $B_i + \varepsilon\theta$.*

Proof of Theorem 5.1. We can find $(\overrightarrow{t^*}, i^*, \theta^*)$ in polytime such that $o_\ell \le t^*_\ell \le (1+\varepsilon)o_\ell$ for all $\ell \in \mathsf{POS}$, we have the right i^*, $o_{i^*} \le \theta^* \le (1+\varepsilon)o_{i^*}$, and $(\overrightarrow{t^*}, i^*, \theta^*)$ clears check P1 (a). We prove this momentarily, but first show that this yields the desired performance guarantee.

Recall that $\varepsilon \le 1$ and $\delta = \varepsilon$. By Lemma 2.8 (b), taking $u = \vec{o}$ and $v = \overrightarrow{t^*}$, we obtain that $f(\overrightarrow{t^{*\mathsf{exp}}}) \le (1+\delta)(1+\varepsilon)f(\vec{o})$. Our analysis shows that the algorithm run with input $(\overrightarrow{t^*}, i^*, \theta^*)$ returns an assignment $\widetilde{\sigma}$. Since we pass checks (b) and (c) in step P1, each machine $i \in \{1, \dots, i^*-1\}$ has load (due to the one job assigned to it) $o_i \le t^{*\mathsf{exp}}_i$. Lemmas 5.5 and 5.6 show that the total load on each machine $i \in \{i^*, \dots, m\}$ is at most

$$B_i + \varepsilon\theta^* \le (1+\varepsilon)t^{*\mathsf{exp}}_i + \varepsilon(1+\varepsilon)o_{i^*} \le (1+\varepsilon)t^{*\mathsf{exp}}_i + 4\varepsilon o_i \le (1+5\varepsilon)t^{*\mathsf{exp}}_i.$$

The second inequality follows from Claim 5.3 and since $\varepsilon \le 1$, and the last one since $\overrightarrow{t^*} \ge (o_\ell)_{\ell\in\mathsf{POS}}$ (and therefore $\overrightarrow{t^{*\mathsf{exp}}} \ge o$). It follows that $\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}} \le (1+5\varepsilon)t^{*\mathsf{exp}}$, and so $f(\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}}) \le (1+5\varepsilon)f(\overrightarrow{t^{*\mathsf{exp}}})$. Combining this with the bound on $f(\overrightarrow{t^{*\mathsf{exp}}})$, and simplifying, we obtain that $f(\overrightarrow{\mathsf{load}_{\widetilde{\sigma}}}) \le (1+23\varepsilon)f(\vec{o})$.

We complete the proof by showing how to find $(\overrightarrow{t^*}, i^*, \theta^*)$ in polynomial time satisfying the properties stated at the beginning of the proof. There are only m choices for i^*. Given the correct i^*, we know that the load on each machine $i < i^*$ is the processing time of the i-th largest job. So we know $o_1, \dots, o_{i^*-1}$ exactly, and can set $t^*_\ell = o_\ell$ for all $\ell \in \mathsf{POS}$, $\ell < i^*$. Let $\mathsf{POS}' = \{i^*\} \cup \{\ell \in \mathsf{POS} : \ell > i^*\}$. Let J' be the set of jobs excluding the

$i^* - 1$ largest jobs. Let $\mathsf{UB}' := \sum_{j \in J'} p_j$. The jobs in J' are assigned by σ^* to machines in $\{i^*, \ldots, m\}$, and we have $o_{i^*} \geq o_i \geq o_{i^*}/2$ for all $i \geq i^*$ (Claim 5.3). It follows that $\frac{\mathsf{UB}'}{m-i^*+1} \leq o_{i^*} \leq T := \min\bigl\{\frac{2\mathsf{UB}'}{m-i^*+1}, o_{i^*-1}\bigr\}$. So by Lemma 2.9 (b), taking $\mathcal{L} = \mathsf{POS}'$, $\mathsf{ub} = T$, and $\mathsf{lb} = \frac{\mathsf{UB}'}{2(m-i^*+1)}$, we can identify a set of size $O\bigl(m^{O(\frac{1}{\varepsilon})}\bigr)$ containing a non-increasing vector $v \in \mathbb{R}_{\geq 0}^{\mathsf{POS}'}$ such that $o_\ell \leq v_\ell \leq \min\{(1+\varepsilon)o_\ell, T\}$ for all $\ell \in \mathsf{POS}'$. So the vector $\overrightarrow{t^*} = \bigl((o_\ell)_{\ell \in \mathsf{POS}: \ell < i^*}, (v_\ell)_{\ell \in \mathsf{POS}: \ell \geq i^*}\bigr)$ and $\theta^* = v_{i^*}$ have the desired properties. ◀

References

1 Noga Alon, Yossi Azar, Gerhard J. Woeginger, and Tal Yadid. Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, 1(1):55–66, 1998.

2 Yossi Azar and Amir Epstein. Convex programming for scheduling unrelated parallel machines. In *Proceedings of the 37th STOC*, pages 331–337, 2005.

3 Deeparnab Chakrabarty. Personal Communication, 2021.

4 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st STOC*, pages 126–137, 2019.

5 Deeparnab Chakrabarty and Chaitanya Swamy. Simpler and better algorithms for minimum-norm load balancing. In *Proceedings of the 27th ESA*, pages 27:1–27:12, 2019.

6 Anindya De, Sanjeev Khanna, Huan Li, and Hesam Nikpey. An efficient PTAS for stochastic load balancing with poisson jobs. In *Proceedings of the 47th ICALP*, pages 37:1–37:18, 2020.

7 Ashish Goel and Piotr Indyk. Stochastic load balancing and related problems. In *Proceedings of the 40th FOCS*, pages 579–586, 1999.

8 Anupam Gupta, Amit Kumar, Viswanath Nagarajan, and Xiangkun Shen. Stochastic load balancing on unrelated machines. In *Proceedings of the 29th SODA*, pages 1274–1285, 2018.

9 G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities.* Cambridge Univ Press, 1934.

10 D. S. Hochbaum and D. B. Shmoys. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SICOMP*, 17(3):539–551, 1988.

11 Dorit Hochbaum and David B. Shmoys. Using dual approximation algorithms for scheduling problems: practical and theoretical results. *J. ACM*, 34(1):144–162, 1987.

12 Sharat Ibrahimpur and Chaitanya Swamy. Approximation algorithms for stochastic minimum-norm combinatorial optimization. In *Proceedings of the 61st FOCS*, pages 966–977, 2020.

13 K. Jansen. An EPTAS for scheduling jobs on uniform processors: Using an MILP relaxation with a constant number of integral variables. In *Proc. 36th ICALP*, pages 562–573, 2009.

14 Klaus Jansen, Kim-Manuel Klein, and José Verschae. Closing the Gap for Makespan Scheduling via Sparsification Techniques. In *Proceedings of the 43rd ICALP*, pages 72:1–72:13, 2016.

15 Jon M. Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM J. Comput.*, 30(1):191–217, 2000.

16 V. S. Kumar, Madhav V Marathe, S. Parthasarathy, and A. Srinivasan. A unified approach to scheduling on unrelated parallel machines. *Journal of the ACM*, 56(5):28, 2009.

17 Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990.

18 Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

19 André Linhares, Neil Olver, Chaitanya Swamy, and Rico Zenklusen. Approximate multi-matroid intersection via iterative refinement. *Math. Program.*, 183(1):397–418, 2020.

20 Konstantin Makarychev and Maxim Sviridenko. Solving optimization problems with diseconomies of scale via decoupling. *J. ACM*, 65(6):42:1–42:27, 2018.

21 Albert W Marshall, Ingram Olkin, and Barry Arnold. *Inequalities: Theory of Majorization and Its Applications.* Springer series in statistics. Springer, 2011.

22 Marco Molinaro. Stochastic ℓ_p load balancing and moment problems via the L-function method. In *Proceedings of the 30th SODA*, pages 343–354, 2019.

23 David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.