

Generalized Unrelated Machine Scheduling Problem

Shichuan Deng*

Jian Li†

Yuval Rabani‡

February 15, 2022

Abstract

We study the *generalized load-balancing* (GLB) problem, where we are given n jobs, each of which needs to be assigned to one of m unrelated machines with processing times $\{p_{ij}\}$. Under a job assignment σ , the *load* of each machine i is $\psi_i(\mathbf{p}_i[\sigma])$ where $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is a symmetric monotone norm and $\mathbf{p}_i[\sigma]$ is the n -dimensional vector $\{p_{ij} \cdot \mathbf{1}[\sigma(j) = i]\}_{j \in [n]}$. Our goal is to minimize the *generalized makespan* $\phi(\text{load}(\sigma))$, where $\phi : \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$ is another symmetric monotone norm and $\text{load}(\sigma)$ is the m -dimensional machine load vector. This problem significantly generalizes many classic optimization problems, e.g., makespan minimization, set cover, minimum-norm load-balancing, etc.

We obtain a polynomial time randomized algorithm that achieves an approximation factor of $O(\log n)$, matching the lower bound of set cover up to constant factor. We achieve this by rounding a novel *configuration* LP relaxation with exponential number of variables. To approximately solve the configuration LP, we design an approximate separation oracle for its dual program. In particular, the separation oracle can be reduced to the *norm minimization with a linear constraint* (NormLin) problem and we devise a polynomial time approximation scheme (PTAS) for it, which may be of independent interest.

1 Introduction

In the *generalized load-balancing* (GLB) problem, we are given a set \mathcal{M} of m unrelated machines, a set \mathcal{J} of n jobs. Let $p_{ij} > 0$ be the processing time of job $j \in \mathcal{J}$ on machine $i \in \mathcal{M}$, and we denote $\mathbf{p}_i = \{p_{ij}\}_{j \in \mathcal{J}}$. Suppose $\sigma : \mathcal{J} \rightarrow \mathcal{M}$ is an assignment of all jobs to machines. Under assignment σ , we use $\mathbf{p}_i[\sigma]$ to denote the n -dimensional vector $\{p_{ij} \cdot \mathbf{1}[\sigma(j) = i]\}_{j \in \mathcal{J}}$ (i.e., we zero out all entries of \mathbf{p}_i that are not assigned to i). The *load* of machine i is define to be $\text{load}_i(\sigma) = \psi_i(\mathbf{p}_i[\sigma])$, where $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is a symmetric monotone norm¹ for each $i \in \mathcal{M}$, called the *inner norms*. We define the *generalized makespan* of assignment σ as $\Phi(\sigma) = \phi(\text{load}(\sigma))$ where $\text{load}(\sigma) = \{\text{load}_i(\sigma)\}_{i \in \mathcal{M}} \in \mathbb{R}_{\geq 0}^m$ (called the load vector) and $\phi : \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$ is another symmetric monotone norm, called the *outer norm*. Our goal is to find an assignment σ such that the generalized makespan $\Phi(\sigma)$ is minimized.

Many special cases of GLB have been studied in the setting of unrelated machine scheduling. The classic makespan minimization problem is one such example, where $\phi = \mathcal{L}_\infty$ and $\psi_i = \mathcal{L}_1$ for each $i \in \mathcal{M}$. For any $\alpha < 3/2$, there is no α -approximation algorithm for makespan minimization unless $\text{P} = \text{NP}$ [21], while improving the current best approximation ratio of 2 [21, 26] remains a longstanding open problem. Several variants of makespan minimization with the objective being the general \mathcal{L}_p norm of the load vector (i.e., $\phi = \mathcal{L}_p$ and $\psi_i = \mathcal{L}_1$) have also been studied extensively. Constant approximations are known [1, 5, 19, 24], with better-than-two factors depending on p . Recently, Chakrabarty and Swamy [10] proposed the minimum-norm load-balancing (MinNormLB) problem, which significantly generalizes makespan minimization and its \mathcal{L}_p norm variants. In MinNormLB the outer norm can be a general symmetric monotone norm, but each inner norm is still $\psi_i = \mathcal{L}_1$. They provided an LP-based constant factor approximation algorithm. The

*IIS, Tsinghua University, China. Email: dsc15@mails.tsinghua.edu.cn

†IIS, Tsinghua University, China. Email: lijian83@mail.tsinghua.edu.cn

‡The Hebrew University of Jerusalem, Israel. Email: yrabani@cs.huji.ac.il

¹Recall $\psi : \mathbb{R}^X \rightarrow \mathbb{R}_{\geq 0}$ is a norm if: (i) $\psi(\mathbf{u}) = 0$ if and only if $\mathbf{u} = 0$, (ii) $\psi(\mathbf{u} + \mathbf{v}) \leq \psi(\mathbf{u}) + \psi(\mathbf{v})$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^X$, (iii) $\psi(\theta \mathbf{u}) = |\theta| \psi(\mathbf{u})$ for all $\mathbf{u} \in \mathbb{R}^X, \theta \in \mathbb{R}$. A norm ψ is monotone if $\psi(\mathbf{u}) \leq \psi(\mathbf{v})$ for all $0 \leq \mathbf{u} \leq \mathbf{v}$, and symmetric if $\psi(\mathbf{u}) = \psi(\mathbf{u}')$ for any permutation \mathbf{u}' of \mathbf{u} .

approximation factor was subsequently improved to $(4 + \epsilon)$ in [11], and the current best result is a $(2 + \epsilon)$ -approximation [17], almost matching the best-known guarantee for makespan minimization.

Although the outer norm has been studied extensively, the inner norm has received relatively less attention except the \mathcal{L}_1 case. However, other inner norms can be naturally motivated. One example is the \mathcal{L}_∞ norm, which can model the RAM size lower bound for each machine, if we view the job sizes $\{p_{ij}\}$ as RAM size requirements and process the assigned jobs sequentially. Other \mathcal{L}_p norms may be used for minimizing average latency [5], energy efficient scheduling [24], etc.

Before stating our results for GLB, we discuss a line of closely related problems. Here the load of machine i is defined by a submodular set function mapping the assigned jobs $\sigma^{-1}(i)$ to a scalar. Svitkina and Fleischer [28] study this problem where the outer norm is $\phi = \mathcal{L}_\infty$ and show that there is no polynomial time algorithm that can achieve an approximation ratio of $o(\sqrt{n/\log n})$. They also provide a factor $O(\sqrt{n \log n})$ approximation algorithm. When $\phi = \mathcal{L}_1$, the problem is known as the minimum submodular-cost allocation problem [12], and admits a tight $O(\log n)$ -approximation, with set cover being its special case [29]. This result implies a nontrivial approximation for the following setting of GLB. Let Top_k be the top- k norm defined as follows: for any vector $\mathbf{u} \geq 0$, $\text{Top}_k(\mathbf{u})$ outputs the sum of the k largest entries in \mathbf{u} . For each $J \subseteq \mathcal{J}$, define $\mathbf{p}_i[J] = \{p_{ij} \cdot \mathbf{1}[j \in J]\}_{j \in \mathcal{J}}$. It is easy to verify that $f(J) = \text{Top}_{k_i}(\mathbf{p}_i[J])$ is a submodular set function for each $k_i \in [n]$ and $\mathbf{p}_i \geq 0$. Thus, GLB with $\phi = \mathcal{L}_1$ and $\psi_i = \text{Top}_{k_i}$ can be reduced to the minimum submodular-cost allocation problem and we readily obtain an $O(\log n)$ -approximation using the result in [29].

However, the reduction above does not work for general symmetric monotone inner norm ψ_i , since a general symmetric monotone norm does not necessarily induce a submodular set function. For example, let $\psi_i(\mathbf{u}) = \max\{2 \text{Top}_1(\mathbf{u}), \text{Top}_3(\mathbf{u})\}$, and $\mathbf{p}_i = \mathbf{1}_n$ (all processing time $p_{ij} = 1$). It is easy to see that ψ_i is indeed a symmetric monotone norm. However, assigning job-sets $\{j_1\}$, $\{j_1, j\}$, $\{j_1, j_2\}$, $\{j_1, j_2, j\}$ incurs loads 2, 2, 2, 3, respectively. The property of diminishing marginal returns (see, e.g., [14]) is violated and the function $f(J) = \psi_i(\mathbf{p}_i[J])$ is not submodular.

1.1 Our Contributions

Motivated by the recent progress on unrelated machine scheduling problems with more general objectives, we study GLB systematically with different combinations of inner norms and outer norms. A summary of known and new results can be found in Figure 1.

The first natural question is whether it is possible to obtain constant factor approximation algorithms for GLB with general outer and inner norms. However, it is not difficult to see an $\Omega(\log n)$ lower bound for GLB, even when $\psi_i = \mathcal{L}_\infty$ and ϕ is simply the \mathcal{L}_1 norm, via reduction from the unweighted set cover problem. Formally, in an unweighted set cover instance, we are given a family of m subsets $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ of $[n]$, and the goal to find $I \subseteq [m]$ with minimum cardinality such that $\bigcup_{i \in I} S_i = [n]$. We identify m machines with $[m]$ and n jobs with $[n]$. For each machine $i \in [m]$ and job $j \in [n]$, define the job size $p_{ij} = 1$ if $j \in S_i$, and ∞ if $j \notin S_i$. The inner norms are $\psi_i = \mathcal{L}_\infty$ for each $i \in \mathcal{M}$, and the outer norm ϕ is simply the \mathcal{L}_1 norm. It is easy to verify that, any assignment σ with a *finite* generalized makespan $\Phi(\sigma)$ corresponds to a solution $I_\sigma = \{i \in [m] : \sigma^{-1}(i) \neq \emptyset\}$ to the set cover instance with the same objective, and vice versa. Using the NP-hardness of approximating set cover [13], we obtain the following theorem.

Theorem 1. *For every fixed constant $\epsilon > 0$, it is NP-hard to approximate GLB within a factor of $(1 - \epsilon) \ln n$, even when $\phi = \mathcal{L}_1$ and $\psi_i = \mathcal{L}_\infty$ for each $i \in \mathcal{M}$.*

On the positive side, we show, somewhat surprisingly, that GLB with general inner and outer norms, admits an $O(\log n)$ factor approximation algorithm, almost matching the approximability of set cover, a very special case of GLB. This is the main technical result of this paper.

Theorem 2. *There exists a polynomial time randomized algorithm for GLB that, with high probability, outputs an $O(\log n)$ -approximate solution.*

We also study another special case. In particular, we obtain an LP-based $(3 + \epsilon)$ -approximation for the special case where the outer norm ϕ is \mathcal{L}_∞ , and the inner norm is $\psi_i = \text{Top}_{k_i}$, $k_i \in [n]$ for each machine $i \in \mathcal{M}$. We provide details of this algorithm (Theorem 13) in the appendix.

Machine load (inner norm ψ_i)	Generalized makespan (outer norm ϕ)		
	\mathcal{L}_1 (SUM)	\mathcal{L}_∞ (MAX)	SYMMONNORM
\mathcal{L}_1 (SUM)	<i>trivial</i>	LB: 3/2 [21]; UB: 2 [21, 26]	LB: 3/2 [21]; UB: $2 + \epsilon$ [17]
\mathcal{L}_∞ (MAX)	LB: $(1 - \epsilon) \ln n$ (Theorem 1); UB: $O(\log n)$ [29]	<i>trivial</i>	LB: $(1 - \epsilon) \ln n$ (Theorem 1); UB: $O(\log n)$ (Theorem 2)
Top $_{k_i}$ norm		LB: 3/2 [21]; UB: $3 + \epsilon$ (Theorem 13)	
SYMMONNORM	LB: $(1 - \epsilon) \ln n$ (Theorem 1);	LB: 3/2 [21];	
	UB: $O(\log n)$ (Theorem 2)		

Figure 1: A summary of GLB cases with different objectives, where n is the number of given jobs. LB (lower bound) represents known inapproximability results, and UB (upper bound) represents current approximation guarantees. SYMMONNORM refers to symmetric monotone norms.

Overview of our algorithm for Theorem 2. We devise a randomized rounding algorithm based on a novel configuration LP. A configuration is an arbitrary set of jobs. We create variables $x_{i,J} \in [0, 1]$ for each $i \in \mathcal{M}$ and $J \subseteq \mathcal{J}$, indicating whether J is the configuration of i . Our goal is to obtain a fractional solution $x \in [0, 1]^{\mathcal{M} \times 2^{\mathcal{J}}}$, such that its generalized makespan is bounded by some unknown optimal solution. To this end, using a standard technique (see, e.g., [10, 17]), we guess a logarithmic number of constraints, also referred to as “budget constraints”, and guarantee that at least one such guess produces a feasible LP that leads to a good approximate solution. We also add various ingredients to facilitate the later rounding procedure, by restricting the support size of vertex solutions, and including parameters to tighten and relax the constraints.

Though the number of variables is exponential, there are only a polynomial number of constraints, thus we consider solving its dual. Unfortunately, the dual constraints are difficult to separate exactly, so we consider approximate separation on *another* parameterized linear program Q , and employ the following round-or-cut framework, which has recently been a powerful tool in the design of approximation algorithms (see, e.g., [2, 8, 9]): Suppose given any candidate dual solution y , we either find a violated constraint of Q , or certify that another differently parameterized program Q' is feasible, then the ellipsoid algorithm, in polynomial time, either concludes that Q is infeasible, or that Q' is feasible. By establishing an equivalence between the feasibility of Q (also Q') and the original relaxation, and enumerating all possible Q , we can solve the primal relaxation up to a constant accuracy.

We use randomized rounding on the primal solution similar to weighted set cover (see, e.g., [30]), and assign all jobs with high probability. At this junction, we isolate two factors that affect the final approximation guarantee. The first arises from assigning multiple distinct configurations to the same machine, incurring a proportional approximation factor. The second comes from the violation of budget constraints in the primal relaxation by the integral solution, and the approximation ratio is proportional to the maximum factor of violation. These two factors add up to $O(\log n)$ in the rounding analysis, yielding our main result.

Finally, we highlight a useful subroutine for the so-called *norm minimization with a linear constraint* (NormLin) problem, which we employ in the approximate separation oracle. Roughly speaking, NormLin generalizes the min-cost version of the knapsack problem with a value lower bound, using a symmetric monotone norm objective (see Section 4 for the precise definition). We devise a polynomial time approximation scheme (PTAS) for NormLin. To the best of our knowledge, this is the first known approximation algorithm for this problem, which may be useful in other context. Svitkina and Fleischer [28] consider a similar problem, where the objective is a submodular set function. They show a lower bound of $\Omega(\sqrt{n/\log n})$ (for any bi-criteria approximation), even for monotone submodular functions and 0-1 values.

1.2 Related Work

In a closely related *simultaneous optimization* problem, one seeks a job assignment incurring a load vector that simultaneously approximates all optimums under different *outer* norms. Though no simultaneous α -approximate solutions exist for unrelated machines for any constant α , even for simple $\psi_i = \mathcal{L}_1$ inner norms [6], Alon *et al.* [1] give an algorithm in the case of restricted assignment (i.e., each job has a fixed size but can only be assigned to a subset of machines) and unit-size jobs, which is simultaneously optimal for all \mathcal{L}_p norms. Azar *et al.* [6] extend this result to a simultaneous 2-approximation for all \mathcal{L}_p norms under restricted assignment. This is generalized to a simultaneous 2-approximation (again in the restricted assignment setting) for all symmetric monotone norms by Goel and Meyerson [16].

Another relevant result is the generalized machine activation problem introduced by Li and Khuller [22], where a machine activation cost is incurred for each machine i , by applying a non-decreasing, piece-wise linear function $\omega_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ on the sum of job sizes assigned to i . They achieve a bi-criteria approximation guarantee on fractional solutions, and obtain various almost-tight application results.

The configuration LP is used in many allocation/assignment optimization problems, for example, the Santa Claus problem and fair allocation of indivisible goods [3, 4, 7, 15], and restricted scheduling on unrelated machines [18, 27].

1.3 Organization

We start by stating some notations and preliminaries in Section 2. In Section 3, we present our main LP-rounding algorithm, and the proof of Theorem 2. We provide the details of the approximate separation oracle in Section 4. Finally, we provide a constant factor approximation algorithm for a special case of GLB in the appendix.

2 Preliminaries

Throughout this paper, for vector $\mathbf{u} \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$, define \mathbf{u}^\downarrow as the non-increasingly sorted version of \mathbf{u} , and $\mathbf{u}[S] = \{\mathbf{u}_j \cdot \mathbf{1}[j \in S]\}_{j \in \mathcal{X}}$ for each $S \subseteq \mathcal{X}$. Let $\text{Top}_k : \mathbb{R}^{\mathcal{X}} \rightarrow \mathbb{R}_{\geq 0}$ be the top- k norm that returns the sum of the k largest *absolute values* of entries in any vector, $k \leq |\mathcal{X}|$. Denote $[n]$ as the set of positive integers no larger than $n \in \mathbb{Z}$, and $a^+ = \max\{a, 0\}$, $a \in \mathbb{R}$.

Claim 3. ([17]). *For each n -dimensional vector $\mathbf{u} \geq 0$ and $k \in [n]$, one has*

$$\text{Top}_k(\mathbf{u}) = \min_{t \geq 0} \left\{ kt + \sum_{j \in [n]} (\mathbf{u}_j - t)^+ \right\} = k\mathbf{u}_k^\downarrow + \sum_{j \in [n]} (\mathbf{u}_j - \mathbf{u}_k^\downarrow)^+,$$

i.e., the minimum is attained at the k -th largest entry of \mathbf{u} .

The following lemma is due to Goel and Meyerson [16].

Lemma 4. ([16]). *If $\mathbf{u}, \mathbf{v} \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$ and $\alpha \geq 0$ satisfy $\text{Top}_k(\mathbf{u}) \leq \alpha \cdot \text{Top}_k(\mathbf{v})$ for each $k \leq |\mathcal{X}|$, one has $\psi(\mathbf{u}) \leq \alpha \cdot \psi(\mathbf{v})$ for any symmetric monotone norm $\psi : \mathbb{R}^{\mathcal{X}} \rightarrow \mathbb{R}_{\geq 0}$.*

We need the following Chernoff bounds.

Lemma 5. (Chernoff bounds (see, e.g., [25])). *Let X_1, \dots, X_n be independent Bernoulli variables with $\mathbb{E}[X_i] = p_i$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X] = \sum_{i=1}^n p_i$. For $\nu \geq 6\mu$, one has $\Pr[X \geq \nu] \leq 2^{-\nu}$.*

3 The Generalized Load Balancing Problem

In this section, we study GLB and prove Theorem 2. For clarity of presentation, we have not optimized the constant factors.

3.1 The Configuration LP

Our algorithm is based on rounding polynomial many configuration LPs. In each configuration LP, instead of using the natural x_{ij} variables to indicate the job assignment, e.g., [21, 26], we create variables $x_{i,J} \in [0, 1]$ for each $i \in \mathcal{M}$ and $J \subseteq \mathcal{J}$, indicating whether J is the set of jobs assigned to i . Similar configuration variables have been used in other allocation/assignment optimization problems [3, 4, 7, 15, 18, 27].

Fix an unknown optimal assignment $\sigma^* : \mathcal{J} \rightarrow \mathcal{M}$. Let $\mathbf{o} = \text{load}(\sigma^*) \in \mathbb{R}_{\geq 0}^m$ be the optimal load vector and $\text{opt} = \phi(\mathbf{o})$ be the optimal objective thereof. We need to guess certain characteristics of the optimal solution (we will show there are polynomial many possible guesses), and for each guess we write a configuration LP P-LB($\mathbf{R}, \lambda, \tau$). Here $\mathbf{R}, \lambda, \tau$ are the parameters we need to adjust, and we explain them later.

We first explain constraints (P-LB.1), which can be used to bound the ϕ -norm. Using Lemma 4, to obtain an $O(\log n)$ -approximate solution, it suffices to obtain a load vector $\text{load}(\sigma) \in \mathbb{R}_{\geq 0}^m$ such that $\text{Top}_k(\text{load}(\sigma))$ is bounded by $O(\log n) \cdot \text{Top}_k(\mathbf{o})$ for each $k \in [m]$. Rather than bounding the top- k norms for each k , we use the trick developed in [10, 17], focusing on a subset of geometrically-placed indexes of $[m]$, e.g., all integer powers of 2 smaller than m . Let $\text{POS} \subseteq [m]$ be this index subset (which we define formally later). Our LP seeks a fractional solution that (roughly) has Top_k norm at most $\text{Top}_k(\mathbf{o})$ for each $k \in \text{POS}$.

$$\min \quad 0 \quad (\text{P-LB}(\mathbf{R}, \lambda, \tau))$$

$$\text{s.t.} \quad k\rho_k + \sum_{i \in \mathcal{M}, J \subseteq \mathcal{J}} (h(\psi_i(J)/\tau) - \rho_k)^+ x_{i,J} \leq \text{Top}_k(\vec{\varrho}) \quad \forall k \in \text{POS} \quad (\text{P-LB.1})$$

$$\sum_{J \subseteq \mathcal{J}} x_{i,J} \leq 1 \quad \forall i \in \mathcal{M} \quad (\text{P-LB.2})$$

$$\sum_{i \in \mathcal{M}, J \ni j} x_{i,J} \geq \lambda \quad \forall j \in \mathcal{J} \quad (\text{P-LB.3})$$

$$\sum_{(i,J): \psi_i(J) > \tau} x_{i,J} \leq 0 \quad (\text{P-LB.4})$$

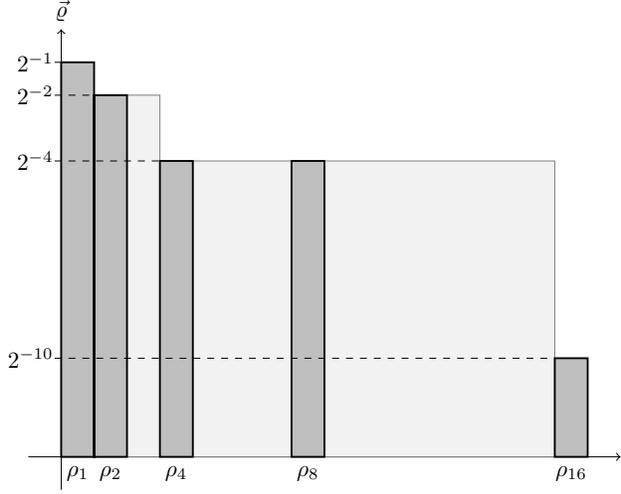
$$\sum_{i \in \mathcal{M}, J \subseteq \mathcal{J}} x_{i,J} \leq n \quad (\text{P-LB.5})$$

$$x \geq 0.$$

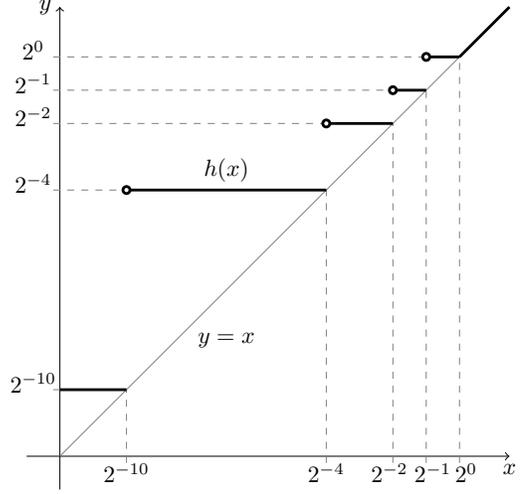
To write linear constraints on such Top_k norms, we utilize Claim 3 and work with close estimates of \mathbf{o}_k^\downarrow , $k \in \text{POS}$, denoted by vector $\mathbf{R} = \{\rho_k\}_{k \in \text{POS}} \in \mathbb{R}_{\geq 0}^{\text{POS}}$ (formally defined later). With a slight abuse of notation, denote $\psi_i(J) = \psi_i(\mathbf{p}_i[J])$ as the ψ_i -norm of assigning job-set J to machine i . We use $k\rho_k + \sum_{i \in \mathcal{M}, J \subseteq \mathcal{J}} (\psi_i(J) - \rho_k)^+ x_{i,J}$ to represent the Top_k norm of LP solution x . We also use a certain “expansion” $\vec{\varrho} \in \mathbb{R}_{\geq 0}^m$ of $\{\rho_k\}_{k \in \text{POS}}$ (proposed in [17]) as an upper bound vector, as shown in constraint (P-LB.1).

We need another important ingredient, that is to tighten the above-mentioned Top_k norm constraints by rounding up each $\psi_i(J)$ to the nearest value in \mathbf{R} . Let $h : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ be this round-up function determined by \mathbf{R} (See Fig. 2). Though this may create an unbounded gap between $h(\psi_i(J))$ and $\psi_i(J)$, the new LP is still feasible under suitable parameters. This rounding-up process is technically useful in our final analysis of randomized rounding.

The configuration LP is parameterized by \mathbf{R} (therefore $\vec{\varrho}$ and h are also determined), a constant $0 \leq \lambda \leq 1$ and a parameter $\tau \geq 1$. As explained above, the constraints (P-LB.1) are for bounding Top_k norms. Note that each $\psi_i(J)$ is scaled by $\tau^{-1} \leq 1$, due to our approximate procedure of solving the LP (see Section 3.2). (P-LB.2) says each machine can be selected to an extent of at most 1. (P-LB.3) says each job needs to be assigned to an extent of at least $\lambda \leq 1$, relaxed again because of our approximate procedure. In (P-LB.4), each assignment (i, J) that incurs a cost larger than τ is set to 0 (similar to the parametric pruning trick used in classic unrelated machine scheduling problems [21, 26]). In (P-LB.5), the total extent of assignment over all possible (i, J) is at most n , because there are only n jobs. This constraint, together with (P-LB.2), further limits the support size of vertex solutions, which will be crucial for optimizing our approximation ratio. We also have the obvious constraints $x \geq 0$.



(a) $\vec{\rho} \in \mathbb{R}_{\geq 0}^{16}$, where the entries are non-increasing and represented using rectangles with unit width.



(b) The round-up function h . The axes are both in logarithmic scale.

Figure 2: An example when $m = 16$ and $\text{POS} = \{1, 2, 4, 8, 16\}$. Here our guess is $\mathbf{R} = (2^{-1}, 2^{-2}, 2^{-4}, 2^{-4}, 2^{-10})$, which induces its expansion $\vec{\rho}$ and the corresponding round-up function h .

The guessing step. We need some additional notations. Let $\text{POS} = \{\min\{2^s, m\} : s \in \mathbb{Z}_{\geq 0}\}$. We have $1, m \in \text{POS}$, $\text{POS} \subseteq [m]$ and $|\text{POS}| \leq \log_2 m + 2$. For each $k \in [m]$, define $\text{next}(k)$ as the smallest number in POS that is larger than k , and $\text{next}(m) = m + 1$; $\text{prev}(k)$ as the largest number in POS that is smaller than k , and $\text{prev}(1) = 0$.

Recall σ^* is an optimal assignment. We guess the machine i^* with the largest $\text{load}_{i^*}(\sigma^*)$, and the job $j^* = \arg \max_{\sigma^*(j)=i^*} p_{i^*j}$ (i.e., j^* is the job assigned to machine i^* with largest processing time). Assume w.l.o.g. that $\psi_{i^*}(\{j^*\}) = 1/n$. Since the norms are monotone, one has $\text{load}_{i^*}(\sigma^*) \leq \psi_{i^*}(p_{i^*j^*} \cdot \mathbf{1}_n) \leq n \cdot \psi_{i^*}(p_{i^*j^*} \cdot e_{j^*}) = n \cdot \psi_{i^*}(\{j^*\}) = 1$ for each $i \in \mathcal{M}$, where e_{j^*} is the zero vector except that the j^* -th coordinate is 1. Thus, we can assume $\mathbf{o} \in [0, 1]^m$ in the following discussion. In the remainder of this section, we assume i^* and j^* are correctly guessed (it is easy to see there are only polynomial number of possible guesses).

We now guess a *non-increasing* vector $\mathbf{R} = \{\rho_k\}_{k \in \text{POS}} \in \mathbb{R}_{\geq 0}^{\text{POS}}$, where

- (i) each ρ_k is an integer power of 2 in $[1/(2mn), 1]$,
- (ii) there are at most $\log_2 n + 4$ distinct entries in \mathbf{R} .

The length is $|\text{POS}| = O(\log m)$ and the number of possible values is the number of integer powers of 2 in $[1/(2mn), 1]$, which is $O(\log(mn))$. The number of such possible non-increasing vectors is at most the number of natural number (i.e., $\mathbb{Z}_{\geq 0}$) solutions to the simple equation $n_1 + n_2 + \dots + n_{O(\log(mn))} = O(\log m)$, thus $\binom{O(\log(mn))}{O(\log m)} \leq 2^{O(\log(mn))} = (mn)^{O(1)}$.

We do not make any additional assumptions on \mathbf{R} for now. For each \mathbf{R} , we define a *non-increasing* vector $\vec{\rho} \in \mathbb{R}_{\geq 0}^m$ as its expansion (similar to [17]), and a *non-decreasing* function $h : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ satisfying $h(x) \geq x$ for $x \geq 0$ (see Fig. 2 for an example), where

$$\vec{\rho}_k = \begin{cases} \rho_k & k \in \text{POS} \\ \rho_{\text{prev}(k)} & k \notin \text{POS}, \end{cases} \quad h(x) = \begin{cases} \min\{t \in \mathbf{R} : t \geq x\} & x \leq \max\{t \in \mathbf{R}\} \\ 1 & x \in (\max\{t \in \mathbf{R}\}, 1] \\ x & x > 1. \end{cases}$$

The dual program. The number of variables in $\text{P-LB}(\mathbf{R}, \lambda, \tau)$ is exponential but the number of constraints is polynomial, thus we consider its dual as follows, and try to solve it using the ellipsoid method.

The dual variables are $\{r_k\}_{k \in \text{POS}}, \{y_i\}_{i \in \mathcal{M}}, \{z_j\}_{j \in \mathcal{J}}, s, t$.

$$\begin{aligned}
\max \quad & - \sum_{k \in \text{POS}} (\text{Top}_k(\vec{\varrho}) - k\rho_k) r_k - \sum_{i \in \mathcal{M}} y_i + \lambda \sum_{j \in \mathcal{J}} z_j - nt & (\text{D-LB}(\mathbf{R}, \lambda, \tau)) \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} z_j - y_i - \sum_{k \in \text{POS}} (h(\psi_i(J)/\tau) - \rho_k)^+ r_k \leq s \cdot \mathbf{1}[\psi_i(J) > \tau] + t \quad \forall i \in \mathcal{M}, J \subseteq \mathcal{J} \\
& r, s, t, y, z \geq 0.
\end{aligned}$$

It is unclear how to separate the dual constraints exactly. We transform $\text{D-LB}(\mathbf{R}, \lambda, \tau)$ into another feasibility problem that is easier to deal with. We observe the following: if $\text{P-LB}(\mathbf{R}, \lambda, \tau)$ is feasible, the optimum of $\text{D-LB}(\mathbf{R}, \lambda, \tau)$ is also zero. Because the dual program has a trivial zero solution and the constraints are scale-invariant (i.e., if (r, s, t, y, z) is feasible, (cr, cs, ct, cy, cz) is feasible for any $c \geq 0$), it either has optimum zero or is unbounded. Thus $\text{P-LB}(\mathbf{R}, \lambda, \tau)$ is feasible if and only if $\text{D-LB}(\mathbf{R}, \lambda, \tau)$ is bounded, which is equivalent to the following polytope being empty.

$$\begin{aligned}
& \left\{ (r, s, t, y, z) \geq 0 \mid \right. & (\text{Q}(\mathbf{R}, \lambda, \tau)) \\
& - \sum_{k \in \text{POS}} (\text{Top}_k(\vec{\varrho}) - k\rho_k) r_k - \sum_{i \in \mathcal{M}} y_i + \lambda \sum_{j \in \mathcal{J}} z_j - nt \geq 1; \\
& \left. - y_i \leq s \cdot \mathbf{1}[\psi_i(J) > \tau] + t + \sum_{k \in \text{POS}} (h(\psi_i(J)/\tau) - \rho_k)^+ r_k - \sum_{j \in \mathcal{J}} z_j, \forall i \in \mathcal{M}, J \subseteq \mathcal{J} \right\}.
\end{aligned}$$

Observation 6. $\text{P-LB}(\mathbf{R}, \lambda, \tau)$ is feasible if and only if $\text{Q}(\mathbf{R}, \lambda, \tau)$ is empty.

3.2 Rounding and Analysis

Next, we employ a round-or-cut argument, similar to [2]. Intuitively speaking, we run the ellipsoid algorithm on a particular $\text{Q}(\mathbf{R}, \lambda, \tau)$. In each iteration, we either find a separating hyperplane, or directly certify that another $\text{Q}(\mathbf{R}, \lambda', \tau')$ is non-empty, which implies that $\text{P-LB}(\mathbf{R}, \lambda', \tau')$ is infeasible by Observation 6. We then pick \mathbf{R} properly so that the latter cannot happen, therefore certify in polynomial time that $\text{Q}(\mathbf{R}, \lambda, \tau)$ is indeed empty. This in turn helps us to efficiently compute a feasible solution to $\text{P-LB}(\mathbf{R}, \lambda, \tau)$.

For clarity of presentation, we focus on two sets of parameters in the remainder of this section, $(\lambda, \tau) \in \{(1/2, 3/2), (1, 1)\}$. We emphasize that this only affects the constant factors in the final analysis. We need the following core lemma on an approximate separation oracle for $\text{Q}(\mathbf{R}, 1/2, 3/2)$, and defer its proof to Section 4. The rest of this section is dedicated to the proof of our main theorem.

Lemma 7. Fix \mathbf{R} . There exists a polynomial time algorithm that, given $(r, s, t, y, z) \geq 0$, either outputs a violated constraint in $\text{Q}(\mathbf{R}, 1/2, 3/2)$, or certifies that $\text{Q}(\mathbf{R}, 1, 1)$ is non-empty.

Proof of Theorem 2. We enumerate all possible estimates \mathbf{R} . Consider the choice of $\mathbf{R} = \{\rho_k\}_{k \in \text{POS}}$ such that the following holds,

- (i) $\rho_k \in [\mathbf{o}_k^\downarrow, 2\mathbf{o}_k^\downarrow]$ for $k \in \text{POS}$ s.t. $\mathbf{o}_k^\downarrow \geq \mathbf{o}_1^\downarrow/(2m)$,
- (ii) $\rho_k = 2^{\lceil \log_2(\mathbf{o}_1^\downarrow/(2m)) \rceil} \in [\mathbf{o}_1^\downarrow/(2m), \mathbf{o}_1^\downarrow/m]$ for $k \in \text{POS}$ s.t. $\mathbf{o}_k^\downarrow < \mathbf{o}_1^\downarrow/(2m)$.

Notice that such a vector is uniquely determined by \mathbf{o} . We denote it by $\mathbf{R}^* = \{\rho_k^*\}_{k \in \text{POS}}$. We show that our exhaustive search is guaranteed to run procedures on \mathbf{R}^* . First, according to our initial guesses, one has $\mathbf{o}_1^\downarrow \geq \psi_{i^*}(\{j^*\}) = 1/n$, $\mathbf{o}_1^\downarrow \leq 1$, therefore (recall that each number in \mathbf{R}^* is an integer power of 2) $\rho_1^* \leq 1$, $\rho_k^* \geq \mathbf{o}_1^\downarrow/(2m) \geq 1/(2mn)$ for each $k \in \text{POS}$, all falling into our guessing range $[1/(2mn), 1]$. Then, there are at most n non-zero entries in \mathbf{o} , since the number of jobs is n . Hence, for $k \in \text{POS}$ and $k > n$, one has $\mathbf{o}_k^\downarrow = 0$, and the number of distinct entries indexed by $\text{POS} \cap [n]$ is at most $\log_2 n + 3$. This implies that the number of distinct entries in \mathbf{R}^* is at most $\log_2 n + 4$. Suppose $\mathbf{R} = \mathbf{R}^*$ in what follows.

For a starting point (r, s, t, y, z) , we repeatedly call the separation oracle in Lemma 7 and use the ellipsoid algorithm to modify the solution. If the oracle always returns a violated constraint as a separating hyperplane,

we eventually obtain that, for a polynomial-sized subset $\mathcal{H} \subseteq \mathcal{M} \times 2^{\mathcal{J}}$, the following polytope is empty,

$$\begin{aligned} & \left\{ (r, s, t, y, z) \geq 0 \mid \right. && \text{(Q}_{\mathcal{H}}(\mathbf{R}, 1/2, 3/2)) \\ & - \sum_{k \in \text{POS}} (\text{Top}_k(\vec{\varrho}) - k\rho_k) r_k - \sum_{i \in \mathcal{M}} y_i + \frac{1}{2} \sum_{j \in \mathcal{J}} z_j - nt \geq 1; \\ & \left. - y_i \leq s \cdot \mathbf{1}[\psi_i(J) > 3/2] + t + \sum_{k \in \text{POS}} (h(2\psi_i(J)/3) - \rho_k)^+ r_k - \sum_{j \in J} z_j, \forall (i, J) \in \mathcal{H} \right\}. \end{aligned}$$

Following a similar argument as Observation 6, by removing constraints in D-LB($\mathbf{R}, 1/2, 3/2$) that are indexed by $\mathcal{M} \times 2^{\mathcal{J}} \setminus \mathcal{H}$ and obtaining a polynomial-sized LP D-LB $_{\mathcal{H}}(\mathbf{R}, 1/2, 3/2)$, the latter is bounded. Thus, its dual P-LB $_{\mathcal{H}}(\mathbf{R}, 1/2, 3/2)$ is feasible with a polynomial number of variables indexed by \mathcal{H} . We directly solve P-LB $_{\mathcal{H}}(\mathbf{R}, 1/2, 3/2)$ and obtain a *vertex solution* \hat{x} . It is easy to see that \hat{x} is feasible to P-LB($\mathbf{R}, 1/2, 3/2$), because P-LB $_{\mathcal{H}}(\mathbf{R}, 1/2, 3/2)$ can be obtained from P-LB($\mathbf{R}, 1/2, 3/2$) by eliminating all variables that are NOT indexed by \mathcal{H} .

We look at the tight constraints at \hat{x} among (P-LB.2) and (P-LB.5). If $m \leq n$, the number is at most $m + 1$; if $m > n$, the number of tight ones in (P-LB.2) is at most n , otherwise (P-LB.5) is violated. Hence there are at most $\min\{m, n\} + 1$ tight constraints among them at \hat{x} . Since there are at most $\log_2 n + 4$ distinct entries in \mathbf{R} hence (P-LB.1), the total number of non-trivial tight constraints at \hat{x} is at most $3n + 5$, and the support size of \hat{x} is at most $3n + 5$ (see, e.g., [20]). Denote the support of \hat{x} by $\hat{\mathcal{H}} \subseteq \mathcal{H}$.

Another possibility is that, in some iteration of the ellipsoid algorithm, the oracle certifies that Q($\mathbf{R}, 1, 1$) is non-empty, and P-LB($\mathbf{R}, 1, 1$) is infeasible due to Observation 6. We show that this is impossible given the choice of $\mathbf{R} = \mathbf{R}^*$. Indeed, we first notice the following.

- (i) for $k \in \text{POS}$, since $\rho_k \in \mathbf{R}$ and $\rho_k \geq \mathbf{o}_k^\downarrow$, we have $\vec{\varrho}_k = \rho_k \geq h(\mathbf{o}_k^\downarrow)$,
- (ii) $\vec{\varrho}_k = \rho_{\text{prev}(k)} \geq h(\mathbf{o}_{\text{prev}(k)}^\downarrow) \geq h(\mathbf{o}_k^\downarrow)$ when $k \notin \text{POS}$,

thus $\vec{\varrho}$ and $h(\mathbf{o}^\downarrow)$ (apply h element-wise) are both non-increasing and $\vec{\varrho} \geq h(\mathbf{o}^\downarrow)$. From the optimal assignment $\sigma^* : \mathcal{J} \rightarrow \mathcal{M}$, we let $x_{i,J}^* = 1$ if $J = \{j : \sigma^*(j) = i\}$ and 0 otherwise. Fix $k \in \text{POS}$ in (P-LB.1) of P-LB($\mathbf{R}, 1, 1$). The LHS at x^* is $k\rho_k + \sum_{s \in [m]} (h(\mathbf{o}_s^\downarrow) - \rho_k)^+ \leq k\rho_k + \sum_{s \in [m]} (\vec{\varrho}_s - \rho_k)^+ = \text{Top}_k(\vec{\varrho})$, via Claim 3, the definition of $\vec{\varrho}$, and that $\vec{\varrho} \geq h(\mathbf{o}^\downarrow)$ is non-increasing. Other constraints in P-LB($\mathbf{R}, 1, 1$) are easily satisfied by x^* . Therefore P-LB($\mathbf{R}, 1, 1$) must be feasible, a contradiction.

In conclusion, when $\mathbf{R} = \mathbf{R}^*$, we obtain a feasible solution \hat{x} to P-LB($\mathbf{R}, 1/2, 3/2$) with support size $|\hat{\mathcal{H}}| \leq 3n + 5$. We proceed to use randomized rounding to obtain a feasible assignment of jobs. Let $\mathcal{I} \leftarrow \emptyset$ and $T = \lceil 6 \ln n \rceil$. For each $t = 1, \dots, T$ and each $(i, J) \in \hat{\mathcal{H}}$, set $\mathcal{I} \leftarrow \mathcal{I} \cup \{(i, J)\}$ independently with probability $\hat{x}_{i,J}$.

The Machine Duplicates. For each $i \in \mathcal{M}$, we have $\sum_{J: (i, J) \in \hat{\mathcal{H}}} \hat{x}_{i,J} \leq 1$ using the constraint (P-LB.2). Therefore for the number of appearances of i in \mathcal{I} , one has

$$\mathbb{E}[|\{J : (i, J) \in \mathcal{I}\}|] = \sum_{J: (i, J) \in \hat{\mathcal{H}}} 1 - (1 - \hat{x}_{i,J})^T \leq T \sum_{J: (i, J) \in \hat{\mathcal{H}}} \hat{x}_{i,J} \leq T,$$

using Bernoulli's inequality. Hence, using Chernoff bound in Lemma 5, on random variables $\mathbf{1}[(i, J) \in \mathcal{I}]$, $(i, J) \in \hat{\mathcal{H}}$, one has

$$\Pr[|\{J : (i, J) \in \mathcal{I}\}| > 6T] = \Pr \left[\sum_{J: (i, J) \in \hat{\mathcal{H}}} \mathbf{1}[(i, J) \in \mathcal{I}] > 6T \right] \leq 2^{-6T} \leq 1/(n^{24}). \quad (1)$$

Note that since $|\hat{\mathcal{H}}| \leq 3n + 5$, at most $3n + 5$ machines can possibly have a non-zero number of appearances in \mathcal{I} . For any other machine i , we always have $|\{J : (i, J) \in \mathcal{I}\}| = 0$.

The Coverage of Jobs. For each $j \in \mathcal{J}$, the probability of it *not* being contained in any selected J in \mathcal{I} is

$$\begin{aligned} \prod_{(i,J) \in \hat{\mathcal{H}}: J \ni j} (1 - \hat{x}_{i,J})^T &\leq \prod_{(i,J) \in \hat{\mathcal{H}}: J \ni j} \exp(-T \hat{x}_{i,J}) \\ &= \exp\left(-T \sum_{(i,J) \in \hat{\mathcal{H}}: J \ni j} \hat{x}_{i,J}\right) \leq \exp(-T/2) \leq 1/n^3, \end{aligned} \quad (2)$$

where we use (P-LB.3) and obtain $\sum_{(i,J) \in \hat{\mathcal{H}}: J \ni j} \hat{x}_{i,J} \geq 1/2$ in the penultimate inequality.

The Norm Constraints. According to (P-LB.4), $\hat{x}_{i,J} > 0$ implies $\psi_i(J) \leq 3/2$ thus $h(2\psi_i(J)/3) \leq h(1) = 1$. Define subsets $\hat{\mathcal{H}}_t = \{(i, J) \in \hat{\mathcal{H}} : h(2\psi_i(J)/3) = 2^t\}$ where $t = 0$ or 2^t is an entry in \mathbf{R} . Let the subsets be indexed by $t \in \text{PWR} \subseteq \mathbb{Z}_{\leq 0}$, then $|\text{PWR}| \leq \log_2 n + 5$ since there are at most $\log_2 n + 4$ distinct entries in \mathbf{R} , and $\bigcup_{t \in \text{PWR}} \hat{\mathcal{H}}_t = \hat{\mathcal{H}}$ because $h(2\psi_i(J)/3) \leq 1$ for each $(i, J) \in \hat{\mathcal{H}}$. For each $t \in \text{PWR}$, we define a random variable $Y_t = |\hat{\mathcal{H}}_t \cap \mathcal{I}|$ and obtain

$$\mathbb{E}[Y_t] = \sum_{(i,J) \in \hat{\mathcal{H}}_t} 1 - (1 - \hat{x}_{i,J})^T \leq T \sum_{(i,J) \in \hat{\mathcal{H}}_t} \hat{x}_{i,J},$$

via Bernoulli's inequality. Using Markov's inequality, with probability at least $1/2$, one has $Y_t \leq 2\mathbb{E}[Y_t]$. Since these (at most) $\log_2 n + 5$ random variables $\{Y_t : t \in \text{PWR}\}$ are independent, there is a probability at least $1/(32n)$ that $Y_t \leq 2\mathbb{E}[Y_t]$ for all $t \in \text{PWR}$. Suppose this happens. Then for each $k \in \text{POS}$, because \hat{x} satisfies (P-LB.1), we have

$$\begin{aligned} \sum_{(i,J) \in \mathcal{I}} (h(2\psi_i(J)/3) - \rho_k)^+ &= \sum_{t \in \text{PWR}} \sum_{(i,J) \in \mathcal{I} \cap \hat{\mathcal{H}}_t} (2^t - \rho_k)^+ = \sum_{t \in \text{PWR}} Y_t \cdot (2^t - \rho_k)^+ \\ &\leq \sum_{t \in \text{PWR}} 2\mathbb{E}[Y_t] (2^t - \rho_k)^+ \leq 2T \sum_{t \in \text{PWR}} \sum_{(i,J) \in \hat{\mathcal{H}}_t} \hat{x}_{i,J} (2^t - \rho_k)^+ \leq 2T (\text{Top}_k(\vec{\varrho}) - k\rho_k). \end{aligned} \quad (3)$$

Putting it all together. Using (1)(2)(3) and the union bound, for *large enough* n , with probability at least $1/(32n) - (3n + 5)/(n^{24}) - 1/n^2 \geq 1/(64n)$, the following facts hold:

- (a) for each $i \in \mathcal{M}$, $|\{J : (i, J) \in \mathcal{I}\}| \leq 6T \leq 38 \ln n$,
- (b) for each $j \in \mathcal{J}$, $\exists (i, J) \in \mathcal{I}$ s.t. $j \in J$,
- (c) for each $k \in \text{POS}$, $\sum_{(i,J) \in \mathcal{I}} (h(2\psi_i(J)/3) - \rho_k)^+ \leq 14 \ln n (\text{Top}_k(\vec{\varrho}) - k\rho_k)$.

We repeat the randomized rounding $64n$ times, and boost the success probability to at least $1 - (1 - 1/(64n))^{64n} \geq 1 - e^{-1} \geq 0.6$. Suppose (a)(b)(c) hold in the rest of this section. Next, we merge all configurations that are identified with the same machine in \mathcal{I} , that is, $J_i \leftarrow \bigcup_{J: (i,J) \in \mathcal{I}} J$ and $\bigcup_{i \in \mathcal{M}} J_i = \mathcal{J}$ according to (b). We only merge $O(\log n)$ configurations for each machine using (a). We also have the following subadditivity

$$\psi_i(J_1) + \psi_i(J_2) \geq \psi_i(\mathbf{p}_i[J_1] + \mathbf{p}_i[J_2]) \geq \psi_i(\mathbf{p}_i[J_1 \cup J_2]) = \psi_i(J_1 \cup J_2),$$

from the triangle inequality on the norm ψ_i , and $s^+ + t^+ \geq (s + t)^+$ for any $s, t \in \mathbb{R}$. Since $h(x) \geq x$ for all $x \geq 0$, one has for each $k \in \text{POS}$,

$$\begin{aligned} \sum_{i \in \mathcal{M}} (2\psi_i(J_i)/3 - 38 \ln n \cdot \rho_k)^+ &\leq_{(a)} \sum_{i \in \mathcal{M}} \left(\sum_{J: (i,J) \in \mathcal{I}} 2\psi_i(J)/3 - |\{J : (i, J) \in \mathcal{I}\}| \cdot \rho_k \right)^+ \\ &\leq \sum_{i \in \mathcal{M}} \sum_{J: (i,J) \in \mathcal{I}} (h(2\psi_i(J)/3) - \rho_k)^+ \leq_{(c)} 14 \ln n (\text{Top}_k(\vec{\varrho}) - k\rho_k). \end{aligned}$$

Define a vector $\mathbf{v} \in \mathbb{R}_{\geq 0}^m$ where $\mathbf{v}_i = \psi_i(J_i)$ for each $i \in \mathcal{M}$. For each $k \in \text{POS}$, via Claim 3, we have

$$\begin{aligned} \text{Top}_k(\mathbf{v}) &\leq k \cdot (57 \ln n \cdot \rho_k) + \sum_{i \in \mathcal{M}} (\psi_i(J_i) - 57 \ln n \cdot \rho_k)^+ \\ &\leq 57 \ln n \cdot k \rho_k + 21 \ln n (\text{Top}_k(\vec{\varrho}) - k \rho_k) \leq 57 \ln n \cdot \text{Top}_k(\vec{\varrho}). \end{aligned}$$

Then for $k \notin \text{POS}$, one has $k < \text{next}(k) < 2k$ and thus

$$\text{Top}_k(\mathbf{v}) \leq \text{Top}_{\text{next}(k)}(\mathbf{v}) \leq 57 \ln n \cdot \text{Top}_{\text{next}(k)}(\vec{\varrho}) \leq 114 \ln n \cdot \text{Top}_k(\vec{\varrho}),$$

and the two inequalities above show that $\phi(\mathbf{v}) \leq O(\log n) \cdot \phi(\vec{\varrho})$ using Lemma 4. By our initial assumption that $\mathbf{R} = \mathbf{R}^*$, we have $\vec{\varrho}_k \leq 2\mathbf{o}_k^\downarrow + \mathbf{o}_1^\downarrow/m$ for each $k \in \text{POS}$, and we now compare $\phi(\vec{\varrho})$ and $\text{opt} = \phi(\mathbf{o})$. First for $k \in \text{POS}$, because $\vec{\varrho}$ is non-increasing, we have

$$\begin{aligned} \text{Top}_k(\vec{\varrho}) &= \sum_{s < k, s \in \text{POS}} \vec{\varrho}_s + \sum_{s < k, s \notin \text{POS}} \vec{\varrho}_s + \vec{\varrho}_k \\ &\leq 2 \sum_{s < k, s \in \text{POS}} \mathbf{o}_s^\downarrow + 2 \sum_{s < k, s \notin \text{POS}} \mathbf{o}_{\text{prev}(s)}^\downarrow + 2\mathbf{o}_k^\downarrow + k \cdot \frac{\mathbf{o}_1^\downarrow}{m} \\ &\leq \mathbf{o}_1^\downarrow + 2\mathbf{o}_k^\downarrow + 2 \sum_{s < k, s \in \text{POS}} (\text{next}(s) - s) \mathbf{o}_s^\downarrow \\ &\leq \mathbf{o}_1^\downarrow + 2\mathbf{o}_k^\downarrow + 2\mathbf{o}_1^\downarrow + 2 \sum_{1 < s < k, s \in \text{POS}} 2(s - \text{prev}(s)) \mathbf{o}_s^\downarrow \\ &\leq 4 \sum_{s \leq k, s \in \text{POS}} (s - \text{prev}(s)) \mathbf{o}_s^\downarrow \leq 4 \sum_{s' \leq k} \mathbf{o}_{s'}^\downarrow = 4 \text{Top}_k(\mathbf{o}), \end{aligned}$$

where we use that \mathbf{o}^\downarrow is non-increasing, and $\text{next}(k) - k \leq 2(k - \text{prev}(k))$ for each $k \in \text{POS}$, by definition of POS. Then for $k \notin \text{POS}$, likewise we obtain

$$\text{Top}_k(\vec{\varrho}) \leq \text{Top}_{\text{next}(k)}(\vec{\varrho}) \leq 4 \text{Top}_{\text{next}(k)}(\mathbf{o}) \leq 8 \text{Top}_k(\mathbf{o}),$$

whence it follows that $\phi(\vec{\varrho}) \leq 8\phi(\mathbf{o})$ by Lemma 4 again, and $\phi(\mathbf{v}) \leq O(\log n) \cdot \phi(\mathbf{o})$. Since the norms are monotone and $\bigcup_{i \in \mathcal{M}} J_i = \mathcal{J}$, any assignment induced by these subsets reveals the same approximation ratio. That is, one can easily obtain a job assignment $\sigma : \mathcal{J} \rightarrow \mathcal{M}$ such that $\sigma^{-1}(i) \subseteq J_i$ for each $i \in \mathcal{M}$, thus $\mathbf{p}_i[\sigma] \leq \mathbf{p}_i[J_i]$ and

$$\Phi(\sigma) = \phi(\{\psi_i(\mathbf{p}_i[\sigma])\}_{i \in \mathcal{M}}) \leq \phi(\{\psi_i(\mathbf{p}_i[J_i])\}_{i \in \mathcal{M}}) = \phi(\mathbf{v}) \leq O(\log n) \cdot \text{opt}. \quad \square$$

4 The Approximate Separation Oracle

In this section, we provide details of the claimed separation oracle in Lemma 7. As a useful subroutine, we consider the following norm minimization with a linear constraint problem and obtain a PTAS, which may be of independent interest.

Definition 8. (NormLin). Given a symmetric monotone norm $\psi : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$, two non-negative n -dimensional vectors $\mathbf{p} = \{p_j\}_{j \in [n]}$, $\mathbf{z} = \{z_j\}_{j \in [n]}$, and a non-negative real number Z , the goal is to find $J \subseteq [n]$ such that $\sum_{j \in J} z_j \geq Z$, so as to minimize $\psi(\mathbf{p}[J])$.

Theorem 9. *There exists a polynomial time approximation scheme for NormLin.*

We first show the separation oracle in Section 4.1, then prove Theorem 9 in Section 4.2.

4.1 Proof of Lemma 7

Fix the solution (r, s, t, y, z) . If it violates the first constraint in $\mathbf{Q}(\mathbf{R}, 1/2, 3/2)$, we directly choose it as a separating hyperplane. If not, we try to approximately verify the second set of constraints, which is the crux of this lemma.

We notice that, for each such constraint, the LHS is determined by i , and the RHS is determined by $\psi_i(J)$ and $\sum_{j \in J} z_j$. Therefore, we consider a surrogate optimization problem as follows: for each $Z \geq 0$, we want to check whether there exist $i \in \mathcal{M}$, $J \subseteq \mathcal{J}$ such that $\sum_{j \in J} z_j \geq Z$, $\psi_i(J)$ is minimized, and the corresponding constraint in $\mathbf{Q}(\mathbf{R}, 1/2, 3/2)$ is violated. Unfortunately, there are an exponential number of possible values for Z , so we can only consider a small subset of approximate values.

Fix $i \in \mathcal{M}$. We enumerate the largest variable z_j in the chosen set $j \in J$, and consider minimizing $\psi_i(J)$ restricted to jobs $\mathcal{J}_{\leq j} := \{j' \in \mathcal{J} : z_{j'} \leq z_j\}$. After we fix z_j , the possible values of $\sum_{j \in J} z_j$ fall in the interval $[z_j, nz_j]$, so we enumerate $Z \in \{z_j, 2z_j, 4z_j, \dots, 2^s z_j\}$, where $s \in \mathbb{Z}$ is the smallest integer such that $2^s \geq n$. The problem now becomes to minimize $\psi_i(J)$ subject to $\sum_{j \in J} z_j \geq Z$ and $J \subseteq \mathcal{J}_{\leq j}$.

This is an instance of NormLin. By Theorem 9, for each sub-problem above, if it is feasible w.r.t. Z , in polynomial time we obtain J such that $\sum_{j \in J} z_j \geq Z$ and $\psi_i(J)$ is at most $3/2$ times the optimum. If, among the output solutions, there exist i, J that violate the said constraint in $\mathbf{Q}(\mathbf{R}, 1/2, 3/2)$, we output the corresponding separating hyperplane; otherwise, though the optimization above is approximate, we still certify the following,

$$-y_i \leq s \cdot \mathbf{1}[\psi_i(J) > 1] + t + \sum_{k \in \text{POS}} (h(\psi_i(J)) - \rho_k)^+ r_k - \frac{1}{2} \sum_{j \in J} z_j, \forall i \in \mathcal{M}, J \subseteq \mathcal{J}. \quad (*)$$

To see this, for the sake of contradiction, suppose there exist i, J that violate $(*)$. When we enumerate the correct $j = \arg \max_{j' \in \mathcal{J}} \{z_{j'}\}$ and Z such that $\sum_{j \in J} z_j \in [Z, 2Z)$, since J is a feasible solution in this case, according to our supposition, we obtain J' such that $\sum_{j \in J'} z_j \geq Z$ and $\psi_i(J') \leq 3\psi_i(J)/2$. But using the assumption of violating $(*)$, one has

$$\begin{aligned} & s \cdot \mathbf{1}[\psi_i(J') > 3/2] + t + \sum_{k \in \text{POS}} \left(h\left(\frac{\psi_i(J')}{3/2}\right) - \rho_k \right)^+ r_k - \sum_{j \in J'} z_j \\ & \leq s \cdot \mathbf{1}[\psi_i(J) > 1] + t + \sum_{k \in \text{POS}} (h(\psi_i(J)) - \rho_k)^+ r_k - Z \\ & \leq s \cdot \mathbf{1}[\psi_i(J) > 1] + t + \sum_{k \in \text{POS}} (h(\psi_i(J)) - \rho_k)^+ r_k - \frac{1}{2} \sum_{j \in J} z_j < -y_i, \end{aligned}$$

violating this constraint in $\mathbf{Q}(\mathbf{R}, 1/2, 3/2)$, and we should output this hyperplane defined by (i, J') in the first place, which is a contradiction.

Let $(r', s', t', y', z') = (r, s, t, y, z/2)$. Since by now the first constraint in $\mathbf{Q}(\mathbf{R}, 1/2, 3/2)$ must be satisfied by (r, s, t, y, z) , using $(*)$, it is easy to verify that (r', s', t', y', z') satisfies all the constraints in $\mathbf{Q}(\mathbf{R}, 1, 1)$, hence it is non-empty. \square

4.2 Norm Minimization with a Linear Constraint

W.l.o.g., assume $\mathbf{p} > 0$, otherwise the problem can be easily reduced to such an instance. We fix a small constant $\epsilon \in (0, 1/2]$ and round each p_j up to its nearest integer power of $1 + \epsilon$ and solve the new instance. This causes us to lose a factor of at most $1 + \epsilon$ in the objective since ψ is a monotone norm. Fix an unknown optimal solution J^* to the *modified instance* with $\mathbf{o} = \mathbf{p}[J^*]$ and $\psi(\mathbf{o}) \leq (1 + \epsilon)\text{opt}$, where $\text{opt} \geq 0$ is the optimum in the *original instance*.

To obtain a solution J with objective $\psi(\mathbf{p}[J])$ bounded by $(1 + O(\epsilon))\psi(\mathbf{o})$, we use the same technique as in Section 3 and consider the Top_k norms of $\mathbf{p}[J]$. As before, we start with some guessing procedures similar to Section 3.1, but with several subtle differences for obtaining a PTAS. Note that we abuse some of the notations in Section 3.1 for convenience. They play very similar roles, but may be defined slightly differently from those in Section 3.

Guessing the optimum. We guess the largest $p_{j_1}, j_1 \in J^*$, the largest $z_{j_2}, j_2 \in J^*$, and suppose w.l.o.g. that $\mathbf{o}_1^\downarrow = p_{j_1} = 1, z_{j_2} = 1$. We assume j_1, j_2 are correctly guessed in the sequel (it is easy to see there are only polynomial number of possible choices).

Define indexes $\text{POS} \subseteq [n]$ iteratively as follows [17]: Set $\text{POS} \leftarrow \{1\}$. Whenever $n \notin \text{POS}$, choose the current largest $t \in \text{POS}$ and add $\lceil(1 + \epsilon)t\rceil$ to POS but no larger than n , i.e.,

$$\text{POS} \leftarrow \text{POS} \cup \{\min\{n, \lceil(1 + \epsilon) \max\{t \in \text{POS}\}\rceil\},\}$$

and it is easy to verify $|\text{POS}| = O(\log_{1+\epsilon} n)$. For each $k \in [n]$, define $\text{next}(k)$ as the smallest number in POS that is larger than k , and $\text{next}(n) = n + 1$; $\text{prev}(k)$ as the largest number in POS that is smaller than k , and $\text{prev}(1) = 0$. Using POS , we guess the following.

- (1) A non-increasing vector $\mathbf{R} = \{\rho_k\}_{k \in \text{POS}} \in \mathbb{R}_{\geq 0}^{\text{POS}}$ such that each entry is a *non-positive* integer power of $(1 + \epsilon)$, and

- (i) $\rho_k = \mathbf{o}_k^\downarrow$ for $k \in \text{POS}$ s.t. $\mathbf{o}_k^\downarrow \geq \epsilon/n$,
- (ii) $\rho_k = (1 + \epsilon)^{\lceil \log_{1+\epsilon}(\epsilon/n) \rceil}$ for $k \in \text{POS}$ s.t. $\mathbf{o}_k^\downarrow < \epsilon/n$.

Since $|\text{POS}| = O(\log_{1+\epsilon} n)$ and the number of possible values is $O(\log_{1+\epsilon}(n/\epsilon))$, the number of such non-increasing vectors is $\binom{O(\log_{1+\epsilon}(n/\epsilon))}{O(\log_{1+\epsilon} n)} \leq 2^{O(\log_{1+\epsilon}(n/\epsilon))} = (n/\epsilon)^{O(1/\epsilon)}$, using a basic counting method as before. We use exhaustive search and assume \mathbf{R} is correct w.r.t. \mathbf{o}^\downarrow in what follows. Define a non-decreasing function $h : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ where

$$h(x) = \begin{cases} \min\{t \in \mathbf{R} : t \geq x\} & x \leq \max\{t \in \mathbf{R}\} \\ x & x > \max\{t \in \mathbf{R}\}. \end{cases}$$

We notice that $h(x) \geq x$ holds for any $x \geq 0$, and $h(x) \leq 1$ for each $x \leq 1$ since $\rho_1 = \mathbf{o}_1^\downarrow = 1$ is the largest entry in \mathbf{R} according to our guesses.

- (2) We start with $\text{PWR} \leftarrow \emptyset$. Given \mathbf{R} and h , for each $s \in \mathbb{Z}$ such that $(1 + \epsilon)^s \in \mathbf{R}$, $\text{PWR} \leftarrow \text{PWR} \cup \{s\}$. We have $\text{PWR} \subseteq \mathbb{Z}_{\leq 0}$ since $\mathbf{o}_1^\downarrow = 1$ and thus $\mathbf{R} \subseteq [0, 1]^{\text{POS}}$.

Define a partition $\mathcal{C} = \{C_t : t = -1, 0, \dots, \lceil 1/\epsilon \rceil\}$ of PWR as follows: For each $s \in \text{PWR}$, guess whether the *number* of indexes in J^* (i.e., the optimum) such that $h(p_j) = (1 + \epsilon)^s$ is $> \lceil 1/\epsilon \rceil$. If so, $C_{-1} \leftarrow C_{-1} \cup \{s\}$; otherwise, guess $0 \leq t \leq \lceil 1/\epsilon \rceil$ as this number and $C_t \leftarrow C_t \cup \{s\}$. Since $|\text{PWR}| \leq |\text{POS}| = O(\log_{1+\epsilon} n)$ and there are $\lceil 1/\epsilon \rceil + 2$ classes, the number of possible partitions is at most $(1/\epsilon)^{O(\log_{1+\epsilon} n)} = n^{O(\epsilon^{-1} \log(1/\epsilon))}$. We use exhaustive search and assume \mathcal{C} is correct w.r.t. J^* in what follows.

We use \mathbf{R} to construct an entry-wise upper bound for \mathbf{o} , using which we write linear constraints and bound the Top_k norms of an LP solution, $k \in \text{POS}$; h is a round-up function that naturally classifies $[n]$ into $O(\log_{1+\epsilon} n)$ classes according to $h(p_j), j \in [n]$. Crucially, we use \mathcal{C} to guess the number of *selected indexes* of each class in the optimum J^* , and only attempt to exactly match the numbers for those classes with a *selected cardinality* $\leq \lceil 1/\epsilon \rceil$. That is, if a class contains $> \lceil 1/\epsilon \rceil$ indexes in J^* , it is added to C_{-1} . This “meta-classification” of classes is useful in our analysis of the approximation factor for *deterministic* rounding.

Similar as before, we define a non-increasing expansion vector $\vec{\rho} \in \mathbb{R}_{\geq 0}^n$, where $\vec{\rho}_k = \rho_k$ if $k \in \text{POS}$, and $\vec{\rho}_k = \rho_{\text{prev}(k)}$ if $k \notin \text{POS}$. Since $\rho_k \geq \mathbf{o}_k^\downarrow$ for each $k \in \text{POS}$ and $\rho_k \in \mathbf{R}$, it follows that $\vec{\rho}_k = \rho_k \geq h(\mathbf{o}_k^\downarrow)$ when $k \in \text{POS}$, $\vec{\rho}_k = \rho_{\text{prev}(k)} \geq h(\mathbf{o}_{\text{prev}(k)}^\downarrow) \geq h(\mathbf{o}_k^\downarrow)$ for each $k \notin \text{POS}$ and thus $\vec{\rho} \geq h(\mathbf{o}^\downarrow)$. We need the following result. The lemma is implied by Lemma 2.7 in [17], and we provide the proof in Appendix A for completeness.

Lemma 10. (Lemma 2.7, [17]). $\psi(\vec{\rho}) \leq (1 + 11\epsilon)\psi(\mathbf{o})$.

LP relaxation. The following relaxation uses the variable $x_j \geq 0$ to represent the extent we include $j \in [n]$ in the solution. Similar to our main relaxation P-LB($\mathbf{R}, \lambda, \tau$), it attempts to bound the norm objective via (NLin. 1), using Top_k norms and Lemma 4. We also have constraints (NLin. 3), (NLin. 4) that restrict the number of indexes selected in the classes $\{j \in [n] : h(p_j) = (1 + \epsilon)^s\}$, $s \in \text{PWR}$, as mentioned above.

$$\min \quad 0 \quad (\text{NLin})$$

$$\text{s.t.} \quad \sum_{j \in [n]} (h(p_j) - \rho_k)^+ x_j \leq \text{Top}_k(\vec{\varrho}) - k\rho_k \quad \forall k \in \text{POS} \quad (\text{NLin. 1})$$

$$\sum_{j \in [n]} z_j x_j \geq Z \quad (\text{NLin. 2})$$

$$\sum_{j: h(p_j) = (1+\epsilon)^s} x_j = t \quad \forall s \in C_t, 0 \leq t \leq \lceil 1/\epsilon \rceil \quad (\text{NLin. 3})$$

$$\sum_{j: h(p_j) = (1+\epsilon)^s} x_j \geq 1 + \lceil 1/\epsilon \rceil \quad \forall s \in C_{-1} \quad (\text{NLin. 4})$$

$$x_j = 0 \quad \forall j \text{ s.t. } p_j > 1 \text{ or } z_j > 1 \quad (\text{NLin. 5})$$

$$x \in [0, 1]^{[n]}.$$

Claim 11. NLin is feasible.

Proof. Set $x_j^* = 1$ if $j \in J^*$ and zero otherwise. It is easy to see that all but the first set of constraints are satisfied by x^* . For (NLin. 1), we fix $k \in \text{POS}$. The LHS is

$$\sum_{s \in [n]} (h(\mathbf{o}_s^\downarrow) - \rho_k)^+ \leq \sum_{s \in [n]} (\vec{\varrho}_s - \rho_k)^+ = \sum_{s \leq k} (\vec{\varrho}_s - \rho_k)^+ = \text{Top}_k(\vec{\varrho}) - k\rho_k,$$

where we use $h(\mathbf{o}^\downarrow) \leq \vec{\varrho}$, $\vec{\varrho}_k = \rho_k$ and the fact that $\vec{\varrho}$ is non-increasing. \square

Proof of Theorem 9. Using Claim 11, we obtain a feasible solution \bar{x} . Our rounding method is very simple. Let $J \leftarrow \emptyset$. For each $s \in \text{PWR}$, define the partial support $J_s = \{j \in [n] : h(p_j) = (1 + \epsilon)^s, \bar{x}_j > 0\}$. $\bigcup_{s \in \text{PWR}} J_s$ is the entire support of \bar{x} by (NLin. 5), $\rho_1 = \mathbf{o}_1^\downarrow = 1$ and definition of h . We have the following cases.

- (i) If $s \in C_t$ for some $0 \leq t \leq \lceil 1/\epsilon \rceil$, using (NLin. 3) one has $\sum_{j \in J_s} \bar{x}_j = t$ and thus $|J_s| \geq t$. We choose t indexes $J'_s \subseteq J_s$ that have the largest z_j values and set $J \leftarrow J \cup J'_s$. It is easy to verify that $\sum_{j \in J'_s} z_j \geq \sum_{j \in J_s} z_j \bar{x}_j$.
- (ii) If $s \in C_{-1}$, using (NLin. 4) one has $\sum_{j \in J_s} \bar{x}_j \geq 1 + \lceil 1/\epsilon \rceil$ and $|J_s| \geq \lceil \sum_{j \in J_s} \bar{x}_j \rceil$. We choose $\lceil \sum_{j \in J_s} \bar{x}_j \rceil$ indexes $J'_s \subseteq J_s$ that have the largest z_j values and set $J \leftarrow J \cup J'_s$. It is easy to verify that $|J'_s| / \sum_{j \in J_s} \bar{x}_j = \lceil \sum_{j \in J_s} \bar{x}_j \rceil / \sum_{j \in J_s} \bar{x}_j \leq (2 + \lceil 1/\epsilon \rceil) / (1 + \lceil 1/\epsilon \rceil) \leq 1 + \epsilon$, and $\sum_{j \in J'_s} z_j \geq \sum_{j \in J_s} z_j \bar{x}_j$.

Using (NLin. 2), it immediately follows that

$$\sum_{j \in J} z_j = \sum_{s \in \text{PWR}} \sum_{j \in J'_s} z_j \geq \sum_{s \in \text{PWR}} \sum_{j \in J_s} z_j \bar{x}_j = \sum_{j \in [n]} z_j \bar{x}_j \geq Z,$$

showing that J is a feasible solution. For the objective, we fix $k \in \text{POS}$. One has

$$\begin{aligned} \sum_{j \in J} (h(p_j) - \rho_k)^+ &= \sum_{s \in \text{PWR}} \sum_{j \in J'_s} (h(p_j) - \rho_k)^+ = \sum_{s \in \text{PWR}} |J'_s| ((1 + \epsilon)^s - \rho_k)^+ \\ &\leq \sum_{s \in \text{PWR}} (1 + \epsilon) \sum_{j \in J_s} ((1 + \epsilon)^s - \rho_k)^+ \bar{x}_j \\ &= (1 + \epsilon) \sum_{j \in [n]} (h(p_j) - \rho_k)^+ \bar{x}_j \leq (1 + \epsilon) (\text{Top}_k(\vec{\varrho}) - k\rho_k), \end{aligned}$$

where we use (NLin.1) in the last inequality. Using $h(x) \geq x$ and Claim 3, for each $k \in \text{POS}$,

$$\begin{aligned} \text{Top}_k(\mathbf{p}[J]) &\leq k\rho_k + \sum_{s \in [n]} (\mathbf{p}[J]_s - \rho_k)^+ \leq k\rho_k + \sum_{j \in J} (h(p_j) - \rho_k)^+ \\ &\leq k\rho_k + (1 + \epsilon) (\text{Top}_k(\vec{\varrho}) - k\rho_k) \leq (1 + \epsilon) \text{Top}_k(\vec{\varrho}). \end{aligned} \quad (4)$$

Then for $k \notin \text{POS}$, by considering $\text{prev}(k)$ and the iterative definition of POS, we have $(k-1)(1+\epsilon) > k$, thus $k > 1 + 1/\epsilon$, and $\text{next}(k) < 1 + (1+\epsilon)k \leq (1+2\epsilon)k$. Since $\mathbf{p}[J]^\downarrow$ and $\vec{\varrho}$ are non-increasing, and $k < \text{next}(k) < (1+2\epsilon)k$, using (4) we have

$$\text{Top}_k(\mathbf{p}[J]) \leq \text{Top}_{\text{next}(k)}(\mathbf{p}[J]) \leq (1 + \epsilon) \text{Top}_{\text{next}(k)}(\vec{\varrho}) \leq (1 + 5\epsilon) \text{Top}_k(\vec{\varrho}), \quad (5)$$

thus from (4)(5) and Lemma 4, Lemma 10, it follows that

$$\psi(\mathbf{p}[J]) \leq (1 + 5\epsilon)\psi(\vec{\varrho}) \leq (1 + 71\epsilon)\psi(\mathbf{o}) \leq (1 + 143\epsilon)\text{opt}.$$

Finally, it is easy to see that the running time of the algorithm is $(n/\epsilon)^{O(\epsilon^{-1} \log(1/\epsilon))}$, which is determined by our guessing procedure. \square

5 Conclusion and Future Directions

In this work, we systematically study the approximation algorithms for GLB with general inner and outer norms. We propose a randomized polynomial time algorithm with logarithmic approximation factor, matching the lower bound up to constant. For certain special case that generalizes classic makespan minimization, we develop a constant factor approximation algorithm.

We propose some interesting future directions. Note the $\Omega(\log n)$ lower bound in Theorem 1 does not necessarily hold for special cases with outer norm $\phi = \mathcal{L}_\infty$ and arbitrary symmetric monotone inner norms. Hence, there is a gap of $\log n$ between the known lower bound and our result (see Fig. 1), which is an interesting open question. Other natural inner and outer objectives are also worth studying, e.g., submodular set functions [28, 29], piece-wise linear functions [22], etc.

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. doi:[https://doi.org/10.1002/\(SICI\)1099-1425\(199806\)1:1<55::AID-JOS2>3.0.CO;2-J](https://doi.org/10.1002/(SICI)1099-1425(199806)1:1<55::AID-JOS2>3.0.CO;2-J).
- [2] Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k -center with covering constraints. In *Integer Programming and Combinatorial Optimization - 21st International Conference*, volume 12125 of *Lecture Notes in Computer Science*, pages 52–65, 2020. doi:[10.1007/978-3-030-45771-6_5](https://doi.org/10.1007/978-3-030-45771-6_5).
- [3] Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. *ACM Trans. Algorithms*, 8(3):24:1–24:9, 2012. doi:[10.1145/2229163.2229168](https://doi.org/10.1145/2229163.2229168).
- [4] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.*, 39(7):2970–2989, 2010. doi:[10.1137/080723491](https://doi.org/10.1137/080723491).
- [5] Yossi Azar and Amir Epstein. Convex programming for scheduling unrelated parallel machines. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 331–337, 2005. doi:[10.1145/1060590.1060639](https://doi.org/10.1145/1060590.1060639).
- [6] Yossi Azar, Leah Epstein, Yossi Richter, and Gerhard J. Woeginger. All-norm approximation algorithms. *J. Algorithms*, 52(2):120–133, 2004. doi:[10.1016/j.jalgor.2004.02.003](https://doi.org/10.1016/j.jalgor.2004.02.003).

- [7] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 31–40, 2006. doi:10.1145/1132516.1132522.
- [8] Robert D. Carr, Lisa Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 106–115, 2000. URL: <http://dl.acm.org/citation.cfm?id=338219.338241>.
- [9] Deeparnab Chakrabarty and Maryam Negahbani. Generalized center problems with outliers. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107 of *LIPICs*, pages 30:1–30:14, 2018. doi:10.4230/LIPICs.ICALP.2018.30.
- [10] Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 126–137, 2019. doi:10.1145/3313276.3316322.
- [11] Deeparnab Chakrabarty and Chaitanya Swamy. Simpler and better algorithms for minimum-norm load balancing. In *27th Annual European Symposium on Algorithms*, volume 144 of *LIPICs*, pages 27:1–27:12, 2019. doi:10.4230/LIPICs.ESA.2019.27.
- [12] Chandra Chekuri and Alina Ene. Submodular cost allocation problem and applications. In *Automata, Languages and Programming - 38th International Colloquium*, volume 6755 of *Lecture Notes in Computer Science*, pages 354–366, 2011. doi:10.1007/978-3-642-22006-7_30.
- [13] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing*, pages 624–633, 2014. doi:10.1145/2591796.2591884.
- [14] Jack R. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization - Eureka, You Shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 11–26, 2001. doi:10.1007/3-540-36478-1_2.
- [15] Uriel Feige. On allocations that maximize fairness. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 287–293, 2008. URL: <http://dl.acm.org/citation.cfm?id=1347082.1347114>.
- [16] Ashish Goel and Adam Meyerson. Simultaneous optimization via approximate majorization for concave profits or convex costs. *Algorithmica*, 44(4):301–323, 2006. doi:10.1007/s00453-005-1177-7.
- [17] Sharat Ibrahimpur and Chaitanya Swamy. Minimum-norm load balancing is (almost) as easy as minimizing makespan. In *48th International Colloquium on Automata, Languages, and Programming*, volume 198 of *LIPICs*, pages 81:1–81:20, 2021. doi:10.4230/LIPICs.ICALP.2021.81.
- [18] Klaus Jansen and Lars Rohwedder. On the configuration-LP of the restricted assignment problem. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2670–2678, 2017. doi:10.1137/1.9781611974782.176.
- [19] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. A unified approach to scheduling on unrelated parallel machines. *J. ACM*, 56(5):28:1–28:31, 2009. doi:10.1145/1552285.1552289.
- [20] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.
- [21] Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990. doi:10.1007/BF01585745.
- [22] Jian Li and Samir Khuller. Generalized machine activation problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 80–94, 2011. doi:10.1137/1.9781611973082.7.

- [23] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.
- [24] Konstantin Makarychev and Maxim Sviridenko. Solving optimization problems with diseconomies of scale via decoupling. *J. ACM*, 65(6):42:1–42:27, 2018. doi:10.1145/3266140.
- [25] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005. doi:10.1017/CB09780511813603.
- [26] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993. doi:10.1007/BF01585178.
- [27] Ola Svensson. Santa claus schedules jobs on unrelated machines. *SIAM J. Comput.*, 41(5):1318–1341, 2012. doi:10.1137/110851201.
- [28] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM J. Comput.*, 40(6):1715–1737, 2011. doi:10.1137/100783352.
- [29] Zoya Svitkina and Éva Tardos. Facility location with hierarchical facility costs. *ACM Trans. Algorithms*, 6(2):37:1–37:22, 2010. doi:10.1145/1721837.1721853.
- [30] Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001. URL: <http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-65367-7>.

A Proof of Lemma 10

By our guesses, we have $\rho_k = \mathbf{o}_k^\downarrow$ if $\mathbf{o}_k^\downarrow \geq \epsilon/n$ and $\rho_k = (1 + \epsilon)^{\lceil \log_{1+\epsilon}(\epsilon/n) \rceil}$ otherwise. For each $k \in \text{POS}$, one then has $\rho_k \leq \mathbf{o}_k^\downarrow + 2\epsilon/n$ and

$$\begin{aligned}
\text{Top}_k(\vec{\rho}) &\leq \sum_{s \leq k, s \in \text{POS}} \mathbf{o}_s^\downarrow + \sum_{s < k, s \notin \text{POS}} \mathbf{o}_{\text{prev}(s)}^\downarrow + k \cdot \frac{2\epsilon}{n} \\
&\leq 2\epsilon + \mathbf{o}_k^\downarrow + \sum_{s < k, s \in \text{POS}} (\text{next}(s) - s) \mathbf{o}_s^\downarrow \\
&= 2\epsilon + k \mathbf{o}_k^\downarrow + \sum_{s < k, s \in \text{POS}} (\text{next}(s) - 1) (\mathbf{o}_s^\downarrow - \mathbf{o}_{\text{next}(s)}^\downarrow) \\
&\leq 2\epsilon + (1 + \epsilon)k \mathbf{o}_k^\downarrow + \sum_{s < k, s \in \text{POS}} (1 + \epsilon)s \cdot (\mathbf{o}_s^\downarrow - \mathbf{o}_{\text{next}(s)}^\downarrow) \\
&= 2\epsilon + (1 + \epsilon) \sum_{s \leq k, s \in \text{POS}} (s - \text{prev}(s)) \mathbf{o}_s^\downarrow \\
&\leq 2\epsilon + (1 + \epsilon) \sum_{s' \leq k} \mathbf{o}_{s'}^\downarrow \leq (1 + 3\epsilon) \text{Top}_k(\mathbf{o}), \tag{6}
\end{aligned}$$

where we use $\text{next}(s) - 1 \leq (1 + \epsilon)s$ for each $s \in \text{POS}$, and the fact that \mathbf{o}^\downarrow is non-increasing. The last inequality is due to $\text{Top}_k(\mathbf{o}) \geq \mathbf{o}_1^\downarrow = 1$.

For each $k \notin \text{POS}$, by the iterative definition of POS, it follows that $(1 + \epsilon)(k - 1) \geq (1 + \epsilon)\text{prev}(k) > k$, thus $k > 1 + 1/\epsilon$ and $\text{next}(k) < (1 + \epsilon)k + 1 \leq (1 + 2\epsilon)k$. Using (6) we have just proved and the fact that $\vec{\rho}, \mathbf{o}^\downarrow$ are both non-increasing, we have

$$\text{Top}_k(\vec{\rho}) \leq \text{Top}_{\text{next}(k)}(\vec{\rho}) \leq (1 + 3\epsilon) \text{Top}_{\text{next}(k)}(\mathbf{o}) \leq (1 + 11\epsilon) \text{Top}_k(\mathbf{o}),$$

where we use $\text{next}(k) \leq (1 + 2\epsilon)k$ in the last inequality. Lemma 4 gives the desired result. \square

B A Simpler Special Case

In this section, we consider a special case of GLB, where each inner norm $\psi_i = \text{Top}_{k_i}$, $k_i \in [n]$ (recall for $\mathbf{u} \geq 0$, $\text{Top}_{k_i}(\mathbf{u})$ returns the sum of largest k_i entries in \mathbf{u}), and the outer norm $\phi = \mathcal{L}_\infty$. We call it the GLB-MaxTopK problem.

For any fixed $\epsilon > 0$, we devise a deterministic $(3 + \epsilon)$ -approximation for GLB-MaxTopK. Analogous to previous algorithms, we set out to guessing the values at different indexes of the optimal assignment, but with slight modifications.

Fix a small $\epsilon > 0$. Define $\text{POS} \subseteq [n]$ iteratively as in the PTAS in Theorem 9 for NormLin, as well as next, prev. Instead of optimizing Top_{k_i} norm for machine i , we consider $\text{Top}_{k'_i}$ norm, where $k'_i = k_i$ if $k_i \in \text{POS}$ and $k'_i = \text{prev}(k_i)$ otherwise. Notice by the iterative definition of POS, if $k_i \notin \text{POS}$, one has $k'_i < k_i < (1 + \epsilon)\text{prev}(k_i) = (1 + \epsilon)k'_i$.

Likewise, let $\sigma^* : \mathcal{J} \rightarrow \mathcal{M}$ be an unknown optimal assignment with optimum $\text{opt} \geq 0$, and $\mathbf{o} \in \mathbb{R}_{\geq 0}^n$ be defined in a subtly different way: for each $k \in [n]$, let \mathbf{o}_k be the maximum k -th largest job size among machines that has $k'_i \geq k$, that is, $\mathbf{o}_k = \max_{i \in \mathcal{M}: k'_i \geq k} \mathbf{p}_i[\sigma^*]_k^\downarrow$ (if there are no such machines or jobs, it is 0). It is easy to see that, since the assigned job-size vectors are non-increasingly sorted and the sets $\{i : k'_i \geq k\}$ are (inclusion-wise) non-increasing in k , \mathbf{o} is a non-increasing vector. We guess $P = \{\rho_k\}_{k \in \text{POS}}$ as a non-increasing vector of integer powers of $1 + \epsilon$ such that

- (i) $\rho_k \in [\mathbf{o}_k, (1 + \epsilon)\mathbf{o}_k]$ if $\mathbf{o}_k \geq \epsilon \mathbf{o}_1/n$,
- (ii) $\rho_k = (1 + \epsilon)^{\lceil \log_{1+\epsilon}(\epsilon \mathbf{o}_1/n) \rceil}$ otherwise.

As before, after we guess the *exact* value of \mathbf{o}_1 and fix it, there are at most $(n/\epsilon)^{O(1/\epsilon)}$ such vectors. Suppose $\mathbf{o}_1 > 0$ and all other guesses are correct in the sequel. It follows that $\rho_1 \leq (1 + \epsilon)\mathbf{o}_1 \leq (1 + \epsilon)\text{opt}$ and for each $i \in \mathcal{M}$,

- (i) if $\mathbf{o}_{k'_i} \geq \epsilon \mathbf{o}_1/n$, we have $k'_i \rho_{k'_i} \leq (1 + \epsilon)k'_i \mathbf{o}_{k'_i}$. Because in the optimal solution, there exists a machine $i' \in \mathcal{M}$ such that $k'_{i'} \geq k'_i$ and its $k'_{i'}$ -th largest assigned job size is $\mathbf{o}_{k'_{i'}}$, i' has a $\text{Top}_{k'_{i'}}$ norm at least $k'_{i'} \mathbf{o}_{k'_{i'}}$. Combined with $k'_{i'} \geq k'_i \geq k'_i$, one has $k'_i \rho_{k'_i} \leq (1 + \epsilon) \text{Top}_{k'_i}(\mathbf{p}_{i'}[\sigma^*]) \leq (1 + \epsilon) \text{Top}_{k'_{i'}}(\mathbf{p}_{i'}[\sigma^*]) \leq (1 + \epsilon)\text{opt}$;
- (ii) otherwise, we have $\rho_{k'_i} = (1 + \epsilon)^{\lceil \log_{1+\epsilon}(\epsilon \mathbf{o}_1/n) \rceil} \leq 2\epsilon \mathbf{o}_1/n$, thus $k'_i \rho_{k'_i} \leq 2\epsilon \mathbf{o}_1 \leq 2\epsilon \cdot \text{opt}$.

Consider the following relaxation where $x_{ij} \geq 0$ represents the extent we assign job j to machine i .

$$\min \quad r \quad (\text{M-Top})$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} (p_{ij} - \rho_{k'_i})^+ x_{ij} \leq r \quad \forall i \in \mathcal{M} \quad (\text{M-Top.1})$$

$$\sum_{i \in \mathcal{M}} x_{ij} = 1 \quad \forall j \in \mathcal{J} \quad (\text{M-Top.2})$$

$$x_{ij} = 0 \quad p_{ij} > \rho_1 \\ x \geq 0. \quad (\text{M-Top.3})$$

Lemma 12. M-Top has optimum at most opt .

Proof. Define an integral solution $x^* \in \{0, 1\}^{\mathcal{M} \times \mathcal{J}}$ according to the optimal assignment σ^* and let $r^* = \text{opt}$. It suffices to show that (x^*, r^*) satisfies (M-Top.1). For each $i \in \mathcal{M}$, because $\rho_{k'_i} \geq \mathbf{o}_{k'_i}$, we have

$$\sum_{j \in \mathcal{J}} (p_{ij} - \rho_{k'_i})^+ x_{ij}^* = \sum_{j \in \sigma^{*-1}(i)} (p_{ij} - \rho_{k'_i})^+ \leq \sum_{j \in \sigma^{*-1}(i)} (p_{ij} - \mathbf{o}_{k'_i})^+.$$

Since the k'_i -th largest job assigned to i has size at most $\mathbf{o}_{k'_i}$ by definition, the above sum has at most k'_i non-zero entries, thus at most $\text{Top}_{k'_i}(\mathbf{p}_i[\sigma^*])$. The lemma now follows since $k'_i \leq k_i$ and $\text{Top}_{k'_i}(\mathbf{p}_i[\sigma^*]) \leq \text{Top}_{k_i}(\mathbf{p}_i[\sigma^*]) \leq \text{opt}$. \square

We solve M-Top and obtain a solution (\bar{x}, \bar{r}) , and $\bar{r} \leq \text{opt}$ by Lemma 12. Using the classic rounding algorithm by Shmoys and Tardos [26] (based on the original job sizes p_{ij}), we obtain an integral assignment $\hat{x} \in \{0, 1\}^{\mathcal{M} \times \mathcal{J}}$ such that $\sum_i \hat{x}_{ij} = 1$ for each job j . More precisely, make $n_i := \left\lceil \sum_j \bar{x}_{ij} \right\rceil$ copies of each machine i ; in non-increasing order of p_{ij} , fractionally assign the jobs $\{j : \bar{x}_{ij} > 0\}$ to the same extent, and sequentially on the copies of i , such that the first $n_i - 1$ copies are all assigned exactly unit mass of jobs. We then use standard methods [23] and round the resulting fractional matching to an arbitrary integral matching \hat{x} , and each job is matched due to (M-Top .2).

Since we assign the jobs in non-increasing order of p_{ij} to the copies, it follows that for the t -th copy i_t of machine i , $t \geq 2$, the assigned job size under \hat{x} is at most the *average* job size on i_{t-1} under \bar{x} . Each assigned job size is at most ρ_1 due to (M-Top .3). Hence for each machine i , the $\text{Top}_{k'_i}$ norm must be attained over its k'_i foremost copies $\{i_1, \dots, i_{k'_i}\}$, which is at most $\rho_1 + \sum_{t \leq k'_i} \sum_j p_{i_t j} \bar{x}_{i_t j}$. Using simple subadditivity $(s+t)^+ \leq s^+ + t^+$ and that each machine copy is assigned to an extent of at most 1, this is in turn bounded by

$$\begin{aligned} \rho_1 + k'_i \rho_{k'_i} + \sum_{t \leq k'_i} \sum_j (p_{i_t j} - \rho_{k'_i})^+ \bar{x}_{i_t j} &\leq \rho_1 + k'_i \rho_{k'_i} + \sum_{j \in \mathcal{J}} (p_{ij} - \rho_{k'_i})^+ \bar{x}_{ij} \\ &\leq (2 + 2\epsilon)\text{opt} + \bar{r} \leq (3 + 2\epsilon)\text{opt}. \end{aligned}$$

Finally using $k_i \leq (1 + \epsilon)k'_i$, the Top_{k_i} norm is bounded by $(1 + \epsilon)(3 + 2\epsilon)\text{opt} \leq (3 + 7\epsilon)\text{opt}$, whence we obtain the following theorem.

Theorem 13. *For each $\epsilon > 0$, there exists a deterministic $(3 + \epsilon)$ -approximation algorithm for GLB-MaxTopK with running time $(n/\epsilon)^{O(1/\epsilon)} \cdot \text{poly}(m, n)$.*