Balancing Notions of Equity: Trade-offs Between Fair Portfolio Sizes and Achievable Guarantees*

Swati Gupta† Jai Moondra‡ Mohit Singh§

Abstract

Motivated by fairness concerns, we study existence and computation of *portfolios*, defined as: given an optimization problem with feasible solutions $\mathcal{D}$, a class $\mathbf{C}$ of fairness objective functions, a set $X \subseteq \mathcal{D}$ of feasible solutions is an α-approximate portfolio if for each objective $f \in \mathbf{C}$, there is an α-approximation for f in X. We study the trade-off between the size $|X|$ of the portfolio and its approximation factor α for various combinatorial problems, such as scheduling, covering, and facility location, and choices of $\mathbf{C}$ as top-k, ordered and symmetric monotonic norms. Our results include: (i) an α-approximate portfolio of size $O\left(\frac{\log d}{\log(\alpha/4)}\right)$ for ordered norms and lower bounds of size $\Omega\left(\frac{\log d}{\log \alpha + \log\log d}\right)$ for the problem of scheduling identical jobs on d unidentical machines, (ii) $O(\log n)$-approximate $O(\log n)$-sized portfolios for facility location on n points for symmetric monotonic norms, and (iii) $\log^{O(r^2)} d$-size $O(1)$-approximate portfolios for ordered norms and $O(\log d)$-approximate for symmetric monotonic norms for covering polyhedra with a constant r number of constraints. The latter result uses our novel `OrderAndCount` framework that obtains an exponential improvement in portfolio sizes compared to current state-of-the-art, which may be of independent interest.

Acknowledgements. This work was supported by NSF Grants CCF-2106444, CCF-1910423, 2112533, NSF CAREER Grant 2239824, and the Georgia Tech ARC-ACO Fellowship.

1 Introduction

With rapid adoption and proliferation of data-driven decisions, widespread inequalities exist in our society in various forms, often perpetuated by optimized decisions to problems in practice. For example, the existence of food deserts is well-documented across the world [32, 2, 12, 15]. The US Department of Agriculture [32] defines a food desert as a low-income census tract where families below poverty line do not have a large[1] grocery chain within 1 mile of their location in urban areas or 10 miles in rural areas. [18] similarly show that medical deserts – regions with significant fraction of population below poverty line, but far off from the nearest medical facility – disproportionately affect racial minorities in the US. The decisions to open such facilities are driven by demand, and therefore, optimized decisions tend to overlook sparsely populated regions with vulnerable populations. As another example, over the last decade, many retailers have adopted scheduling optimization systems [3]. These systems draw on various data to predict customer demand and make decisions about the most efficient workforce schedule. Some systems, e.g. Percolata, estimate sales productivity scores for each worker and create schedules based on these scores. Concerns about fairness of workload again arise, as such optimizations result in highly variable, unpredictable, and discordant schedules for workers. Further, there is evidence of workload inequity in many work environments, including academia [28], last-mile delivery drivers [26], and hospital workers [30].

In such applications, the decision is often to maximize the efficiency in the system, however, this results in unequal costs borne by various groups of people. A large number of fairness notions have been proposed in the literature that attempt at "balancing" such costs across groups or individuals, such as minimizing some norm of the distances traveled by groups of people [6, 9, 18, 29], finding simultaneous solutions [24, 16, 17], balancing statistical outcomes in machine learning [10, 14, 19], and balancing allocations in social welfare problems [11]. However, even these notions of fairness can be fundamentally incompatible in the sense that a single solution may not be fair with respect to two or more notions of fairness [22, 18]. One workaround is to understand the

*The full version of the paper can be accessed at `https://arxiv.org/abs/2311.03230`

†Massachusetts Institute of Technology.
‡Georgia Institute of Technology.
§Georgia Institute of Technology.

[1] "Large" is defined as a store with at least $2 million annual profit and containing all traditional food departments.

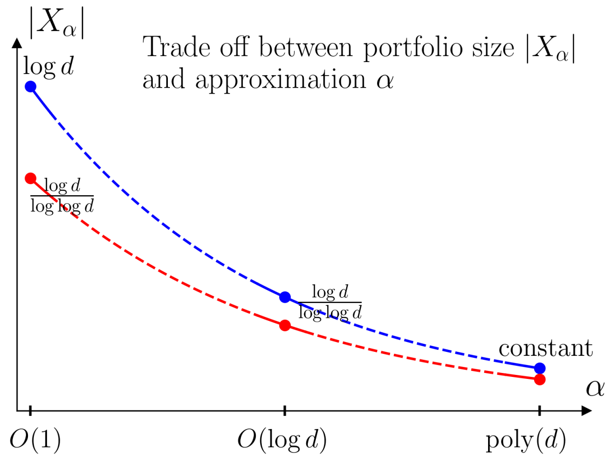

Figure 1: A qualitative plot to illustrate the trade-off between approximation α and the smallest portfolio size $|X_\alpha|$ for the Machine-Loads-Identical-Jobs problem for ordered norms. The worst-case lower bound $|X_\alpha| = \Omega\left(\frac{\log d}{\log \alpha + \log \log d}\right)$ is illustrated in red, and the upper bound $|X_\alpha| = O\left(\frac{\log d}{\log(\alpha/4)}\right)$ is illustrated in blue. The two bounds converge for $\alpha = \Omega(\log d)$.

possibilities offered by a (small) set of solutions, called *portfolios*, so that there is some representative solution achieving approximate fairness for any single notion of fairness [18]. Motivated by the practice of selecting organ transplantation policies, [18] define the *portfolio* problem as follows: *given an optimization problem with a set or domain of feasible vectors $\mathcal{D}$, a class* $\mathbf{C}$ *of objective functions that represent various equity notions, an approximation factor α, and size s, find a portfolio $X \subseteq \mathcal{D}$ of s solutions, so that for any objective $f \in \mathbf{C}$ there exists a solution $x \in X$ that α-approximates* $\min_{x \in \mathcal{D}} f(x)$. X is called an α-approximate portfolio. The case $s = 1$ generalizes simultaneous approximations.

For various combinatorial problems and different classes of objectives, it is not clear what the minimum size of an α-approximate portfolio needed to achieve a given approximation factor is. Larger portfolios are needed for better approximations, and the goal is to keep size s of the portfolio small. Further, as the set $\mathbf{C}$ of equity objectives grows larger, small portfolios may not even exist. For the class of equity objectives, we study

1. Top-k norms, $k \in [d]$, where the top-k norm of a vector $x \in \mathbb{R}^d$ is the sum of the k highest coordinates of x by absolute value. Top-k norms generalize the L_1, L_∞ norms.

2. Ordered norms, where given a non-zero *weight vector* $w \in \mathbb{R}^d_{\geq 0}$ with decreasing weights $w_1 \geq \ldots \geq w_d \geq 0$, the ordered norm of $x \in \mathbb{R}^d_{\geq 0}$ is the weighted sum of coordinates of x with the kth highest coordinate of x weighted by the kth highest weight w_k. Ordered norms generalize top-k norms and have a natural fairness interpretation when x is a vector of individual costs.

3. Symmetric monotonic norms, which are norms that are invariant to the permutation of coordinates and nondecreasing in each coordinate. L_p norms, top-k norms, and ordered norms are all symmetric monotonic norms.[2]

In this work, we partially answer the question:

"What is the trade-off between achievable portfolio size and corresponding approximation factors for various combinatorial optimization problems? Is there a general recipe for constructing small portfolios for ordered and symmetric monotonic norms?"

In particular, we focus on three general combinatorial problems: scheduling, covering, and facility location, motivated by workplace scheduling and access to critical facilities. While much effort has gone into determining the best-possible simultaneous approximations (portfolio of size 1), little is known about the construction of portfolios of size greater than 1. For top-k norms, [16] essentially obtain a $(1+\epsilon)$-approximate portfolio of size $O\left(\frac{\log d}{\epsilon}\right)$; a similar bound holds for L_p norms [17, 18]. However, for ordered norms, only a general construction of poly$(d^{1/\epsilon})$-sized $(1+\epsilon)$-approximate portfolios was known before this work, due to [6], while no bound was

[2]Ordered norms are fundamental to symmetric monotonic norms in two aspects: each symmetric monotonic norm (1) is $O(\log d)$-approximated by some ordered norm [29], and (2) is the supremum of some set of ordered norms.

Table 1: Approximations for size > 1 portfolios for ordered norms and symmetric monotonic norms, for arbitrary $\epsilon \in (0, 1]$. Previously, only a poly$(d^{1/\epsilon})$-sized portfolio was known [6] for $(1+\epsilon)$-approximation for ordered norms, for dimension d problems.

Problem or set of feasible vectors $\mathcal{D}$	Worst-case approximation factor for simultaneous approximation	Guarantees for portfolio of size > 1		
		Size	Approximation for ordered norms	Approximation for symmetric monotonic norms
Machine-Loads-Identical-Jobs d machines	$\Omega(\sqrt{d})$	$O\left(\frac{\log d}{\epsilon}\right)$	$4+\epsilon$	$O(\log d)$
Covering-Polyhedra with r constraints: $\{x \in \mathbb{R}^d_{\geq 0} : Ax \geq b\}$, $A \in \mathbb{R}^{r \times d}_{\geq 0}, b \in \mathbb{R}^r_{\geq 0}$	$\Omega(\sqrt{d})$	$\left(\frac{\log(d/\epsilon)}{\epsilon}\right)^{O(r^2)}$	$1+\epsilon$	$O(\log d)$
Uncapacitated-Facility-Location on n points	$\Omega(\sqrt{n})$	$O(\log n)$	$O(\log n)$	

known for symmetric monotonic norms. We observe that their result generalizes to symmetric monotonic norms in Appendix B.1. It was also known that when the size of the portfolio is 1 (also referred to as *simultaneous* approximations) and it is α-approximate for top-k norms, then it is in fact α-approximate for all symmetric monotonic norms [16]. *This property is no longer true for portfolios of size greater than 1* (Theorem 2.3). In particular, we show that the approximation guarantee of a portfolio for top-k norms and ordered orders can differ by a factor polynomial in d. Consequently, we cannot restrict to constructing portfolios only for top-k norms and need new techniques for the much larger set of ordered norms and symmetric monotonic norms. We show that there exist polytopes $\mathcal{D} \subseteq \mathbb{R}^d$ for which the portfolio size must be $d^{\Omega(1/\log\log d)}$ (i.e., nearly polynomial in d) for ordered and symmetric monotonic norms even for approximation as large as $O(\log d)$ (see Appendix B). In our main contributions, listed next, we develop a general algorithmic framework called `OrderAndCount` to obtain portfolios for ordered norms for covering problems, and obtain size polylog(d) or smaller portfolios for various combinatorial problems.

1. Characterizing Trade-off for Machine-Loads-Identical-Jobs. As our first result, we consider the Machine-Loads-Identical-Jobs problem where n identical jobs must be scheduled on d unidentical machines to minimize some norm of the vector of machine loads. This is a simple model for workload distribution among d workers with different processing speeds, and various norms correspond to various fairness criteria for fair distribution of jobs.

Result 1. (Theorems 2.1, 2.2, Section 2) *For the* Machine-Loads-Identical-Jobs (MLIJ) *problem with d machines, given any approximation factor $\alpha > 4$, we can find an α-approximate portfolio X_α for ordered norms satisfying $|X_\alpha| = O\left(\frac{\log d}{\log(\alpha/4)}\right)$. Further, we construct instances with the lower bound of $|X_\alpha| = \Omega\left(\frac{\log d}{\log \alpha + \log\log d}\right)$.*

In other words, the size-approximation trade-off is that the product $\log(\alpha) \cdot |X_\alpha|$ remains nearly a constant as function of α. This result completely characterizes the trade-off between portfolio sizes and achievable approximation factors (up to $\log\log$ factor) for the Machine-Loads-Identical-Jobs problem (See Figure 1). To obtain this result, we use our `OrderAndCount` approach, which exploits the fact that each ordered norm, while convex in general, is a *linear function* when restricted to a region where all vectors satisfy the same order of coordinate values. That is, if vector $x \in \mathbb{R}^d$ satisfies $x_{\pi(1)} \geq x_{\pi(2)} \geq \ldots \geq x_{\pi(d)} \geq 0$ for some order π on $[d]$, the ordered norm $\|x\|_{(w)}$ is the linear function $\sum_k w_k x_{\pi(k)}$. This gives the following algorithm to obtain portfolios for ordered norms: for each order π, we can restrict to the set $\mathcal{D}_\pi$ of vectors in $\mathcal{D}$ that satisfy order π, and collect the set of extreme points of $\mathcal{D}_\pi$. This in general results in exponentially many solutions (there are exponentially many orders π and potentially exponentially many extreme points of each $\mathcal{D}_\pi$). We show that for MLIJ, (i) it suffices to restrict to a specific order π^* (that depends on the problem instance), and that (ii) there are at most d extreme points of $\mathcal{D}_{\pi^*}$. These d extreme points can further be α-approximated by a subset of $O(\log_{\alpha/4} d)$ *integral* points using our rounding algorithm.

2. Exponential Improvement in Portfolios for Covering. Next, we consider the Covering-Polyhedra problem, which simply includes r covering constraints of the form: $a^\top x \geq b$ (for $a \in \mathbb{R}^d_{\geq 0}$, and

$b \in \mathbb{R}_{\geq 0}$) along with nonnegativity $x \geq 0$. This generalizes the MLIJ problem above, and models many natural scenarios for workload distribution.

For example, consider the following scenario: a centralized server needs to distribute several jobs among d machines that each contain r different parallel processing units (e.g., a CPU, GPU, and NPU) that each handle a different type of job. Different fairness criteria then balance loads over the machines differently. This problem is particularly interesting in the context of volunteer-dependent non-profit organizations, such as HIV social care centers, blood donation drives, food recovery organizations [27], etc. Numerous studies have been conducted on the reasons for the attrition of volunteers, and overburdening by the amount of demands placed on them is one of the key ones [25, 23]. This work can help balance workloads in volunteer organizations, to help mitigate its impact on attrition.

Back to the machine load scheduling terminology, if b_j units for the jth job type need to be scheduled, and the machine $i \in [d]$ has processing speed $A_{j,i}$ for the jth type of job, then the total loads $x_i, i \in [d]$ on the machines must satisfy $\sum_{i\in[d]} A_{j,i}x_i \geq b_j$. For a given norm $\|\cdot\|$ or fairness criterion, this translates to minimizing $\|x\|$ over the covering polyhedron $\{x \in \mathbb{R}^d : Ax \geq b, x \geq 0\}$.

The challenge in extending `OrderAndCount` to such problems is (i) bounding the number of possible orders that the optimal solution x^* might satisfy, and then (ii) selecting a subset of corresponding extreme points for each order that must be included in the portfolio. For the first challenge we develop a novel *primal-dual counting technique* which allows us to count the number of possible orders in an appropriate dual space that is structurally much simpler (Section 3). For the second challenge, we show that a sparsification procedure allows us to reduce the number of extreme points for each order. Together, using `OrderAndCount`, we give poly-logarithmic sized portfolios for Covering-Polyhedra for constant r:

Result 2. (Theorem 3.1, Section 3) *For* Covering-Polyhedra *in d dimensions and r constraints, for any $\epsilon \in (0, 1]$, we obtain a $(1+\epsilon)$-approximate portfolio $X_{1+\epsilon}$ of size $|X_{1+\epsilon}| = O\big(\log(d/\epsilon)/\epsilon\big)^{3r^2-2r}$ for ordered norms, and $O(\log d)$-approximate portfolio of size $O(\log^{3r^2-2r} d)$ for symmetric monotonic norms. Our algorithm runs in time polynomial in d and $(\log(d)/\epsilon)^{r^2}$.*

The trade-off between ϵ and $X_{1+\epsilon}$ is that $|X_{1+\epsilon}|^{1/\Omega(r^2)} \cdot \epsilon$ remains nearly a constant. For all $r = o\left(\frac{\sqrt{\log d}}{\log\log d}\right)$, the above-mentioned result is the first *exponential improvement* over the current best bound of poly$(d^{1/\epsilon})$ [6], to the best of our knowledge.

3. Facility Location Problems. Finally, we consider the generalizations of classical k-clustering[3] and (uniform) uncapacitated facility location problems. In both problems, given a metric space (X, dist) on $|X| = n$ points, we seek to *open* a facility set $F \subseteq X$ at some points in the space. Each feasible solution induces a vector $x_F \in \mathbb{R}^n$ of distances of points in X to their closest open facility. The k-clustering problem seeks the facility set F with at most k open facilities to minimize some norm of this vector, e.g., k-median seeks to minimize the L_1 norm while k-center seeks to minimize the L_∞ norm. In the Uncapacitated-Facility-Location problem, there is no bound on the number of facilities and we instead seek to minimize the sum of the number of open facilities[4] $|F|$ and some given norm of the vector x_F of closest distances to open facilities.

Result 3. (Theorems 4.1, 4.3, Section 4) *We give a portfolio of size 1 for k-*Clustering *that opens at most $O\left(k \cdot \frac{\log n}{\epsilon}\right)$ facilities and is $(3+\epsilon)$-approximate for any symmetric monotonic norm objective, improving upon the previous-best approximation factor $6+\epsilon$ [16, 24]. Using this result, we obtain a size $O(\log n)$-portfolio for* Uncapacitated-Facility-Location *that is $O(\log n)$-approximate for symmetric monotonic norms.*

Such solutions for k-Clustering that violate the constraint of opening at most k facilities are known as bicriteria solutions; [17] showed that it is necessary to open $O(k \log n)$ facilities to guarantee constant-factor approximation for all symmetric monotonic norms. Our algorithm seeks to iteratively cover points in X with balls of increasing radii until all points are covered. While an $O(1)$-approximate portfolio of size $O(\log n)$ was

[3]Unfortunately, both k-clustering and top-k norms use the index 'k' as the parameter in their commonly used names. We hope our usage of the term 'k' will be clear from context.

[4]In the most general version of the problem, each facility has an opening cost; we assume unit opening costs for simplicity, but our techniques and guarantees can be modified suitably for the more general setting.

known for Uncapacitated-Facility-Location for L_p norms [18], our result is the first for the much larger class of symmetric monotonic norms, to the best of our knowledge.

The rest of the paper is organized as follows: we give an overview of techniques next before listing related work in Section 1.2. We give preliminaries in Section 1.3. Machine-Loads-Identical-Jobs is discussed in Section 2, Covering-Polyhedra are discussed in Section 3, and k-Clustering and Uncapacitated-Facility-Location are discussed in Section 4. We discuss open problems and conclude in Section 5.

1.1 Overview of Techniques. In Section 1.1.1, we describe the main ideas behind the `OrderAndCount` algorithm for Covering-Polyhedra and MLIJ. In Section 1.1.2, we discuss the additional rounding algorithm for MLIJ. In Section 1.1.3, we discuss k-clustering and uncapacitated facility location.

1.1.1 Covering-Polyhedra. Recall that we are given a covering polyhedron $\mathcal{P} = \{x \in \mathbb{R}^d : Ax \geq b, x \geq 0\}$ with r constraints, i.e., $A \in \mathbb{R}^{r\times d}_{\geq 0}$ and $b \in \mathbb{R}^r_{\geq 0}$, and a parameter $\epsilon \in (0, 1]$. We construct a $(1 + \epsilon)$-approximate portfolio X for ordered norms, i.e. a subset $X \subseteq \mathcal{P}$ such that for all ordered norms $\|\cdot\|$ on $\mathbb{R}^d$, $\min_{x\in X} \|x\| \leq (1 + \epsilon) \min_{x\in\mathcal{P}} \|x\|$. [29] show that any symmetric monotonic norm in dimension d can be $O(\log d)$-approximated by an ordered norm, and therefore our construction also implies a $O(\log d)$-approximate for symmetric monotonic norms. Denote $N = O\big(\frac{\log(d/\epsilon)}{\epsilon}\big)$; the size of our portfolio X will be $N^{O(r^2)}$ where r is the number of rows in constraint matrix A. As remarked, this size is poly-logarithmic in d for constant r.

For integer $d \geq 1$, let $\mathrm{Perm}(d)$ denote the set of all orders or permutations on $[d]$. We say that a vector $x \in \mathbb{R}^d_{\geq 0}$ satisfies order $\pi \in \mathrm{Perm}(d)$ if $x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0$ and denote the sorted vector $x^\downarrow = x_\pi = (x_{\pi(1)}, \ldots, x_{\pi(d)})$. An ordered norm [6] $\|\cdot\|_{(w)}$ is specified by a non-zero *weight vector* $w \in \mathbb{R}^d_{\geq 0}$ such that $w_1 \geq \ldots \geq w_d \geq 0$ and defined as $\|x\|_{(w)} := w^\top |x|^\downarrow = \sum_{k\in[d]} w_k x_{\pi(k)}$.

Since there are an infinite number of weight vectors (and hence ordered norms), it is unclear if there is even a finite set of 'extreme points' $X \subseteq \mathcal{P}$ with $\min_{x\in X} \|x\|_{(w)} = \min_{x\in\mathcal{P}} \|x\|_{(w)}$ for all ordered norms $\|\cdot\|_{(w)}$. In our terminology, such a set X is an optimal (or 1-approximate) portfolio for ordered norms. We first obtain a *finite* optimal portfolio X, and then outline the procedure to reduce the size of X to $N^{O(r^2)}$, losing approximation factor $(1 + \epsilon)$ in the process.

Note that [6] showed that there are at most $\mathrm{poly}(d^{1/\epsilon})$ ordered norms on $\mathbb{R}^d$ up to a $(1 + \epsilon)$-approximation. Therefore, a $(1 + \epsilon)$-approximate portfolio of size $\mathrm{poly}(d^{1/\epsilon})$ for ordered norms can be obtained by taking the minimum norm points with respect to these ordered norms. Our goal is to reduce this size even further when r is small.

We can assume without loss of generality by re-scaling A that $b = \mathbf{1}_r$. We can also assume without loss of generality that all rows of A are independent (i.e. A is full row-rank). For a fixed ordered norm $\|\cdot\|_{(w)}$, we can write a convex program to minimize $\|x\|_{(w)}$ over $\mathcal{P}$:

$$\text{(Primal)} \qquad \min \|x\|_{(w)} \quad \text{s.t.} \quad Ax \geq \mathbf{1}_r, x \geq 0.$$

Denote the optimal solution to this convex program as $x(w)$. Suppose we are told that $x(w)$ satisfies a specific order $\pi \in \mathrm{Perm}(d)$, then we can add constraints $x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0$ to Primal:

$$\min \|x\|_{(w)} \quad \text{s.t.} \quad Ax \geq \mathbf{1}_r, x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0.$$

$x(w)$ is still optimal for this new convex program. However, under the new order constraints, the function $\|x\|_{(w)} = w^\top x^\downarrow = \sum_{k\in[d]} w_k x_{\pi(k)}$ is *linear*. Therefore, without loss of generality, we can assume that $x(w)$ is a vertex of this new feasible region $\mathcal{P}_\pi := \{x \in \mathbb{R}^d : Ax \geq \mathbf{1}_r, x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0\}$. Collecting all vertices across all $\pi \in \mathrm{Perm}(d)$, this gives us the optimal portfolio $X := \bigcup_{\pi\in\mathrm{Perm}(d)}(\text{vertices of } \mathcal{P}_\pi)$ for ordered norms. Clearly, X is finite. Further, by our discussion, each ordered norm $\|\cdot\|_{(w)}$ achieves its minimum over $\mathcal{P}$ at some point in X.

The size of portfolio X could be much larger than our target size due to two issues:

1. Each $\mathcal{P}_\pi$ can have too many vertices. For each vertex of $\mathcal{P}_\pi$, d out of $r + d$ constraints $Ax \geq \mathbf{1}_r, x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0$ must be tight. Therefore, there could be $\binom{d+r}{d} \sim d^r$ vertices of $\mathcal{P}_\pi$.

2. There are $d!$ orders $\pi \in \mathrm{Perm}(d)$. Since we are taking a union over all such orders, union bound gives the

following bound on the portfolio size $|X|$:

$$|X| \le \binom{\text{number of vertices}}{\text{in each } \mathcal{P}_\pi} \times \binom{\text{number of orders}}{\pi} \sim d^r \times d!. \tag{1.1}$$

To reduce the size of the portfolio to poly-logarithmic in d, we use the following two ideas:

1. *Sparsification.* For each row of matrix A, we round down each entry in the row to the nearest multiple of $(1+\epsilon)$ and set all entries smaller than $\frac{\epsilon}{3d^2}$ times the maximum entry in the row to zero. This obtains a new instance where the optimal solution changes by a factor at most $1+\epsilon$ for any norm. In this new instance, there are at most $N = O\left(\frac{\log(d/\epsilon)}{\epsilon}\right)$ distinct entries in each row and thus, at most N^r distinct columns. That is, the set $[d]$ of columns/coordinates can be partitioned into $S_1, \dots, S_{N^r}$ such that any two columns in each S_l are equal.

   We show that for any minimum norm point $x(w)$ for any ordered norm $\|\cdot\|_{(w)}$, this result allows us to assume that the coordinates of $x(w)$ in any S_l are equal. This allows us to reduce the first factor in eqn. (1.1) to N^{r^2}.

2. *Primal-dual counting.* We still need to reduce the second factor in eqn. (1.1). Our goal is to show that there is a set of orders[5] $\Pi \subseteq \mathrm{Perm}(d)$ such that (a) the optimal $x(w)$ for each weight vector w satisfies some order $\pi \in \Pi$, and (b) $|\Pi| \le N^{2r(r-1)}$. If we plug this into eqn. (1.1) combined with the sparsification idea, we get an overall bound of $N^{r^2+2r(r-1)} = N^{3r^2-2r}$.

   To show this, we formulate a 'dual' mathematical program to Primal. To formulate this dual, we appeal to the characterization of dual norms of ordered norms, and state the Cauchy-Schwarz inequality for them:

LEMMA 1.1. (DUAL ORDERED NORMS) *Given a weight vector $w \in \mathbb{R}^d$, the dual norm $\|\cdot\|^*_{(w)}$ to ordered norm $\|\cdot\|_{(w)}$ is given by*

$$\|y\|^*_{(w)} = \max_{k\in[d]} \frac{\sum_{i\in[k]} |y|^\downarrow_i}{\sum_{i\in[k]} w_i}.$$

LEMMA 1.2. (ORDERED CAUCHY-SCHWARZ) *For all $x, y \in \mathbb{R}^d_{\ge 0}$,*

$$\|x\|_{(w)}\|y\|^*_{(w)} \ge x^\top y.$$

Further, equality holds if and only if

(a) there is some order $\pi \in \mathrm{Perm}(d)$ such that x, y both satisfy π.

(b) for each $k \in [d]$ either $x^\downarrow_k = x^\downarrow_{k+1}$ or $\frac{\sum_{i\in[k]} y^\downarrow_i}{\sum_{i\in[k]} w_i} = \|y\|^*_{(w)}$.

We let the $(r-1)$-dimensional simplex be denoted by $\boldsymbol{\Delta}_r = \{\lambda \in \mathbb{R}^r_{\ge 0} : \sum_{i\in[r]} \lambda_i = 1\}$. The dual program to Primal is formulated as follows:

$$\min_{\lambda\in\boldsymbol{\Delta}_r} \|\lambda^\top A\|^*_{(w)}. \tag{Dual}$$

For any x that is feasible for Primal and λ that is feasible for Dual, Lemma 1.2 implies that $\|x\|_{(w)}\|A^\top\lambda\|^*_{(w)} \ge x^\top A^\top \lambda \ge \mathbf{1}_r^\top \lambda = \|\lambda\|_1 = 1$. Moreover, we show that equality holds if and only if x and λ are optimal to Primal and Dual respectively, and x and $A^\top\lambda$ satisfy the same order π. This allows us to count the total number of candidate orders in the dual using geometric arguments. We show that the simplex $\boldsymbol{\Delta}_r$ can be partitioned into different regions $\bigcup_{\pi\in\mathrm{Perm}(d)} R_\pi$, where for all λ in region R_π, $A^\top\lambda$ satisfies order π. These regions are formed by $\binom{N^r}{2}$ hyperplanes of the form $\{\lambda : (A^\top\lambda)_j = (A^\top\lambda)_{j'}\}$ partitioning $\boldsymbol{\Delta}_r$. Using an inductive argument, we show that $\binom{N^r}{2}$ hyperplanes can partition $\boldsymbol{\Delta}_r$ into at most $N^{2r(r-1)}$ regions, finishing the counting of Π, and hence the proof.

[5]The formal proofs work not with orders $\mathrm{Perm}(d)$ but what we call 'reduced orders' (to be defined later). Our goal in this section is to present the intuitive ideas and outline of the proof; we defer technicalities to Section 3.

1.1.2 Machine-Loads-Identical-Jobs. Recall that here we are given n identical jobs and d unidentical machines, say with processing times $p_1, \dots, p_d > 0$ per job. We need to schedule each job (integrally) on a machine so as to minimize some norm of the machine loads; set $\mathcal{D}$ is the set of all possible machine load vectors. If $n_i \in \mathbb{Z}_{\geq 0}$ jobs are scheduled on machine $i \in [d]$, then we must have (1) $\sum_{i \in [d]} n_i = n$ and (2) machine loads $x_i = n_i p_i$ for each $i \in [d]$. That is, machine load vector x satisfies $\sum_{i \in [d]} \frac{x_i}{p_i} = n$, $x \geq 0$. If fractional loads were allowed, $\mathcal{D}$ would precisely be a covering polyhedron with $r = 1$ constraint, and a portfolio would follow as a corollary of our result for covering polyhedra.

However, since each job must be assigned integrally, minimization over the covering polyhedron $\min_{x: \sum_i x_i/p_i = n} \|x\|$ is only a relaxation of the original integral problem. Further, this relaxation has $\Omega(d)$ integrality gap: consider $n = 1$, $p_i = 1$ for each i. Then for L_∞ norm, the minimum-norm point on the hyperplane is $x = (1/d, \dots, 1/d)$ while the integral optimum is $x^* = (1, 0, \dots, 0)$, so that $\|x^*\|_\infty / \|x\|_\infty = d$. Nevertheless, this relaxation is still useful, and we will give a rounding algorithm based on this relaxation that will *bypass* this integrality gap issue.

We outline the main idea to help with this. First, relabel the machines so that the processing times on different machines satisfy $0 < p_1 \leq p_2 \leq \dots \leq p_d$. We consider the special cases where each p_i is a power of 2; we call such instances 'doubling instances'; it is easily seen that an arbitrary instance is 2-approximated by some doubling instance. We show that doubling instances have the following nice property: for any symmetric monotonic norm $\|\cdot\|$ on $\mathbb{R}^d$, the optimal schedule with loads x^{OPT} satisfies $x_1^{\mathrm{OPT}} \geq \dots \geq x_d^{\mathrm{OPT}} \geq 0$. This gives an idea for an integral relaxation: for a given norm $\|\cdot\|$, we guess the index $l \in [d]$ such that $x_l^{\mathrm{OPT}} > 0$ but $x_{l+1}^{\mathrm{OPT}} = \dots = x_d^{\mathrm{OPT}} = 0$. Given this guess, a better convex relaxation for the problem is: $\min \|x\|$ such that $\sum_{i \in [l]} \frac{x_i}{p_i} = n$ and $x_1 \geq \dots \geq x_l \geq x_{l+1} = \dots = x_d = 0$. We show that for any ordered norm $\|\cdot\|$, one of these relaxations has a constant integrality gap and we can round a fractional solution to an integral one.

1.1.3 k-Clustering and Uncapacitated-Facility-Location. For the k-clustering problem in a metric space on n points with distances dist, where at most k facilities must be opened in a metric space to minimize some norm of the vector of distance of points to their nearest open facilities. [24] first showed that their exists a solution that opens at most $O(k \log n + \frac{1}{\epsilon})$ facilites with approximation factor at most $9 + \epsilon$ for each symmetric monotonic norm. Further, they showed that such a *bicriteria* solution that violates the number of open facilities by factor $\Omega(\log n)$ is in fact necessary. [16] essentially give an algorithm with at most $O\left(\frac{k \log n}{\epsilon}\right)$ facilities with an improved approximation factor $6 + \epsilon$.

In our algorithm that further improves the approximation to $3 + \epsilon$, we use the following result of [8]: given a radius $R > 0$, they find k facilities that cover at least as many points within radius $3R$ that any other set of k facilities can cover within radius R. Given this subroutine, our algorithm is as follows: starting with a suitably small radius R_0, we use the subroutine to find k facilities that cover as least as many points within radius $3R_0$ that any other set of facilities cover in radius R_0. Then we keep increasing this radius by factor $1 + \epsilon$ and repeat this step until the radius is large enough so that all n points are covered. R_0 is chosen so that the number of iterations in the algorithm is at most $O((\log n)/\epsilon)$, leading to the bound on the number of open facilities. Further, the procedure is explicitly designed to have pointwise $3 + \epsilon$-approximation guarantees on the distances with respect to any solution that opens k facilities. This gives the simultaneous approximation guarantee for k-clustering. Note that if we don't need the result to be polynomial-time, then we don't need to lose the factor of 3 and can further improve the approximation to $1 + \epsilon$.

We remark that [8]'s algorithm also works when facilities are only allowed to open in a subset of the points, and consequently our algorithm works in that setting as well. As pointed out to us by a reviewer, if this restriction is dropped (i.e. if facilities can open anywhere in X), then a linear programming-based rounding algorithm using [31]'s technique gives an even better $(2 + \epsilon)$-approximation in polynomial-time.

For the facility location problem that does not put a bound k on the number $|F|$ of open facilities but instead seeks to minimize the sum of $|F|$ and the norm of the distance vector, we use the result for k-clustering as a subroutine: first, we can search for the value $|F|$ among the $O(\log n)$ values $\{2^0, 2^1, \dots, 2^{\log_2 n}\}$. Then, we use the corresponding $|F|$-clustering solution that opens at most $O(|F| \log n)$ facilities (for any constant $\epsilon \in (0, 1]$), thus giving an $O(\log n)$-approximation for the facility location objective.

1.2 Related Work. Portfolios were explicitly first studied by [18] who studied them for facility location problems. Similar notions were implicit in other previous works: [16] essentially constructed $O(\log d)$-size $O(1)$-approximate portfolios for top-k norms in dimension d, [17] used the structure of L_p norms to get a similar bound, and [6] essentially constructed poly(d)-size $O(1)$-approximate portfolios for ordered norms. All three techniques rely on counting the number of unique norms (up to $O(1)$-approximation). In contrast, all our techniques rely on counting vectors in the set $\mathcal{D}$ of feasible vectors. This shift is useful, for example, in obtaining polynomial-sizes portfolios for symmetric monotonic norms (see Appendix B.1).

Portfolios of size-1 or simultaneous approximations have been very well-studied, with the earliest results going as far back as [7]. [24, 16, 17] all studied general techniques that often involve (implicitly) obtaining portfolios and combining them into one solution. [16] proved that a simultaneous α-approximation for top-k norms is a simultaneous α-approximation for symmetric monotonic norms. [17] observed that the basic structure of [4]'s algorithm for the Traveling Salesman Problem (TSP) can be applied to many other problems, obtaining logarithmic or constant-factor apporoximate simultaneous approximations. Our technique for k-clustering is somewhat similar to this algorithm, with the main idea being to reduce the original problem to a *partial problem* where only a subset of clients need to be satisfied.

[24] studied simultaneous approximations for all symmetric monotonic norms for clustering, scheduling, and flow problems. In particular, for k-CLUSTERING, they obtained a $(9 + \epsilon, O(\log n) + \epsilon^{-1})$-approximation in polynomial time. [16] improved this to $(6 + \epsilon, O((\log n)/\epsilon)$. [16] also gave a PTAS for scheduling on identical machines. [1] proved that this PTAS is in-fact a simultaneous 1.388-approximation. [13] gave a simultaneous approximation algorithm for TSP.

Optimizing for a fixed non-standard objective has been widely considered in the literature, and the list is too long to fit here. [6] studied ordered norm and symmetric monotonic norm objectives for scheduling and clustering problems and proved that any symmetric monotonic norm is the supremum of some ordered norms, thus establishing ordered norms as fundamental to the study of symmetric convex functions. [29] proved that any symmetric monotonic norm can be $O(\log d)$-approximated by an ordered norm, further strengthening this connection.

1.3 Preliminaries. We give formal definitions and useful preliminary results in this section. Omitted proofs are included in Appendix A. Throughout, we assume that $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ is a set of feasible vectors with each coordinate representing the cost to individuals/groups in a combinatorial problem (e.g., distances to open facilities in facility location problems or machine loads in scheduling problems). First, we define portfolios formally:

DEFINITION 1. (PORTFOLIOS) *Given a domain or set of feasible vectors $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$, a class of objectives $\mathbf{C} : \mathcal{D} \to \mathbb{R}_{\geq 0}$, and an approximation parameter $\alpha \geq 1$, a portfolio $X \subseteq \mathcal{D}$ is a set of vectors such that for all objectives $f \in \mathbf{C}$,*

$$\min_{x \in X} f(x) \leq \alpha \min_{x \in \mathcal{D}} f(x).$$

When the portfolio has size 1*, it is called a* simultaneous α-approximation *[16, 24].*

For vector $x \in \mathbb{R}^d$, $|x|^{\downarrow}$ is the vector with coordinates of $|x|$ sorted in decreasing order. $\sigma x = (x_1, x_1 + x_2, \ldots, x_1 + \ldots + x_d)$ denotes the prefix sums of x and $\Delta x := (x_1 - x_2, x_2 - x_3, \ldots, x_d - x_{d+1})$, with the convention $x_{d+1} = 0$. We will denote by $\mathbf{1}_k \in \mathbb{R}^d$ the vector with k ones followed by zeros.

DEFINITION 2. (NORM CLASSES) *Given a vector $x \in \mathbb{R}^d$,*

1. *for $k \in [d]$, the top-k norm of x is $\sigma(|x|^{\downarrow})_k = \mathbf{1}_k^\top |x|^{\downarrow}$, i.e. the sum of highest k coordinates of $|x|$. The class of top-k norms is denoted* **Top***;*
2. *given a nonzero* weight vector $w \in \mathbb{R}^d$ *such that $w_1 \geq \ldots \geq w_d \geq 0$, the ordered norm $\|x\|_{(w)}$ is defined as $w^\top |x|^{\downarrow}$. The class of ordered norms is denoted* **Ord***;*
3. *a symmetric monotonic norm is a norm that is monotone in each coordinate and invariant to the permutation of coordinates. The class of symmetric monotonic norms is denoted* **Sym**.

For nonnegative $x, y \in \mathbb{R}^d_{\geq 0}$, we say that y majorizes x or $x \preceq y$ if $(\sigma x^{\downarrow})_k \leq (\sigma y^{\downarrow})_k$ for all $k \in [d]$. The first lemma connects symmetric norm values with majorization.

LEMMA 1.3. ([20]) *If $x \preceq y$, then $\|x\| \leq \|y\|$ for any $\|\cdot\| \in$ **Sym**.*

This helps connect simultaneous approximation for **Top** to simultaneous approximation for **Sym**: if x^* is simultaneously α-approximate for **Top** over $\mathcal{D}$, then $(\sigma|x^*|^{\downarrow})_k \leq \alpha(\sigma|y|^{\downarrow})_k$ for all k and $y \in \mathcal{D}$, or that $x^* \preceq \alpha y$ for all $y \in \mathcal{D}$. As an immediate consequence:

OBSERVATION 1. ([16], THEOREM 2.3) *For any $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$, if x^* is a simultaneous α-approximation for **Top**, then x^* is a simultaneous α-approximation for **Sym**.*

For any $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$, a $(1+\epsilon)$-approximate portfolio for **Top** can be obtained by choosing optimal solutions corresponding to top-k norms for $k = \lfloor 1+\epsilon \rfloor, \lfloor (1+\epsilon)^2 \rfloor, \ldots$ [2]. There are $\log_{1+\epsilon}(d) = O((\log d)/\epsilon)$ such values, and so:

OBSERVATION 2. *For any $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ and $\epsilon \in (0,1]$, there is a $(1+\epsilon)$-approximate portfolio of size $O((\log d)/\epsilon)$ for top-k norms **Top**.*

We remark that unlike Observation 1 for simultaneous approximations, portfolio guarantees do not carry over from **Top** to **Sym** or **Ord**. Indeed, despite the above observation for **Top**, the best-known upper bound for $O(1)$-approximate portfolio sizes for **Ord** and **Sym** is polynomial in d. The next lemma allows symmetric monotonic norms to be $O(\log d)$-approximated by ordered norms; it will be useful to convert portfolios for **Ord** to portfolios for **Sym**:

LEMMA 1.4. ([29]) *Any symmetric monotonic norm $\|\cdot\|$ on $\mathbb{R}^d$ can be $O(\log d)$-approximated by an ordered norm on $\mathbb{R}^d$.*

COROLLARY 1.1. *Given $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$, an α-approximate portfolio X for **Ord** over $\mathcal{D}$ is an $O(\alpha \log d)$-approximate portfolio for **Sym** over $\mathcal{D}$.*

The next lemma shows that portfolios can be composed in different ways:

LEMMA 1.5. (PORTFOLIO COMPOSITION) *Given class **C** of functions over $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$*

1. *If X_1 is an α_1-approximate portfolio for **C** over $\mathcal{D}$ and X_2 is an α_2-approximate portfolio for **C** over X_1, then X_2 is an $\alpha_1\alpha_2$-approximate portfolio for **C** over $\mathcal{D}$.*
2. *If $\mathcal{D} = \bigcup_{i\in[n]} \mathcal{D}_i$ and X_i is an α-approximate portfolio for **C** over $\mathcal{D}_i$ for each $i \in [n]$, then $\bigcup_{i\in[n]} X_i$ is an α-approximate portfolio for **C** over $\mathcal{D}$.*

We restate the characterization of the class of dual norms to ordered norms and a corresponding Cauchy-Scwarz inequality; their proofs are included in Appendix A.

LEMMA 1.1. (DUAL ORDERED NORMS) *Given a weight vector $w \in \mathbb{R}^d$, the dual norm $\|\cdot\|^*_{(w)}$ to ordered norm $\|\cdot\|_{(w)}$ is given by*

$$\|y\|^*_{(w)} = \max_{k\in[d]} \frac{\sum_{i\in[k]} |y|^{\downarrow}_i}{\sum_{i\in[k]} w_i}.$$

LEMMA 1.2. (ORDERED CAUCHY-SCHWARZ) *For all $x, y \in \mathbb{R}^d_{\geq 0}$,*

$$\|x\|_{(w)}\|y\|^*_{(w)} \geq x^\top y.$$

Further, equality holds if and only if

1. *there is some order $\pi \in \mathrm{Perm}(d)$ such that x, y both satisfy π.*
2. *for each $k \in [d]$ either $x^{\downarrow}_k = x^{\downarrow}_{k+1}$ or $\frac{\sum_{i\in[k]} y^{\downarrow}_i}{\sum_{i\in[k]} w_i} = \|y\|^*_{(w)}$.*

[2] [16] use this proof strategy to obtain simultaneous approximations.

2 `OrderAndCount` for Machine-Loads-Identical-Jobs

In this section, we present the first application of `OrderAndCount` framework to the Machine-Loads-Identical-Jobs (MLIJ) problem. Recall that we are asked to assign n copies of a job among d processors or machines with different processing times $p_i, i \in [d]$. This is the simplest model for workload distribution where some tasks must be distributed among individuals in a workplace: processors correspond to individuals, processing times represent their efficiencies, and balancing loads on machines corresponds to managing the workloads of the individuals. Given a norm $\|\cdot\|$ on $\mathbb{R}^d$, the goal is to schedule the jobs to minimize the norm of the machine load vector. We seek a portfolio of solutions (i.e. schedules) for ordered norms **Ord** and symmetric monotonic norms **Sym**.

First, we observe a simple example where no solution is simultaneous $o(\sqrt{d})$-approximation: suppose there are $n = d$ jobs and $p_1 = 1$ while $p_2 = \ldots = p_d = \sqrt{d}$. The optimal solution for L_∞ (i.e. maximum load) minimization assigns one job per machine to get maximum load $\sqrt{d}$. The optimal solution for L_1 (i.e. total load) minimization assigns all jobs to the most efficient machine, i.e., machine 1, for total load of d. Therefore, any assignment with $< d/2$ jobs on machine 1 is an $\Omega(\sqrt{d})$-approximation for L_1 norm, and any assignment with $\geq d/2$ jobs on machine 1 is an $\Omega(\sqrt{d})$-approximation for L_∞ norm. This motivates us to increase the portfolio size. We prove the following results characterizing the approximation-portfolio size trade-off in this section; we note that the guarantee for **Sym** follows from the guarantee for **Ord** using Lemma 1.4.

Theorem 2.1. *For any instance of* MLIJ *on d machines and any $\alpha > 4$, we can find in polynomial time an α-approximate portfolio of size $O\left(\frac{\log d}{\log(\alpha/4)}\right)$ for ordered norms* **Ord**. *This portfolio is also $O(\alpha \log d)$-approximate for symmetric monotonic norms* **Sym**.

Theorem 2.2. *For any constant $\alpha > 1$, there exists an instance of* MLIJ *on d machines where any α-approximate portfolio X_α for ordered norms* **Ord** *has size $|X_\alpha| = \Omega\left(\frac{\log d}{\log \alpha + \log\log d}\right)$. The same bound holds for symmetric monotonic norms* **Sym**.

We will also prove (Theorem 2.3) that there are instances of MLIJ with optimal portfolio of size 2 for **Top** but with no $O(1)$-approximate portfolio of size $o\left(\frac{\log d}{\log\log d}\right)$ for **Ord**.

We start with some notation. Since all jobs are identical, we can identify a schedule by the number of jobs on each machine. If $n_i \in \mathbb{Z}_{\geq 0}$ jobs are scheduled on machine i, then $\sum_{i \in [d]} n_i = n$, and the load vector is $x = x(n) = (n_1 p_1, \ldots, n_d p_d)$. Therefore, the set of feasible vectors is $\mathcal{D} = \{x \in \mathbb{R}^d : x \geq 0;\ \sum_i n_i = n;\ \ x_i = n_i p_i\ \forall\, i \in [d]\}$. We can relabel the machine indices and assume without loss of generality that $0 < p_1 \leq \ldots \leq p_d$.

2.1 Portfolio Upper Bound.

At a high level, we show that special instances of MLIJ that we call *doubling instances* satisfy two key properties: (i) any instance of MLIJ is 2-approximated by some doubling instance (Lemma 2.1), and (ii) the optimal solution x^{OPT} to a doubling instance satisfies $x_1^{\text{OPT}} \geq x_2^{\text{OPT}} \geq \ldots \geq x_d^{\text{OPT}}$ (Lemma 2.2). These inequalities allows us to relax the integrality constraints and look at the polyhedron $\mathcal{P} = \{x : \sum_i \frac{x_i}{p_i} = n; x_1 \geq \ldots \geq x_d \geq 0\}$, where the coordinate-wise inequality constraints can be put in for doubling instances. This sets up `OrderAndCount`: there is only one possible order for vectors $x \in \mathcal{P}$, which is $x_1 \geq \ldots \geq x_d \geq 0$. Each ordered norm $\|x\|_{(w)} = w^\top x$ is a linear function over $\mathcal{P}$, and so the set of vertices of $\mathcal{P}$ form an optimal portfolio for ordered norms over $\mathcal{P}$ for the doubling instance and a 2-approximate portfolio for the original instance. We show that we can restrict to $O(\log_{\alpha/4} d)$ of these vertices, losing factor $\alpha/4$. Finally, we lose another factor 2 in carefully rounding back to the original, to get an overall approximation factor α for ordered norms.

Lemma 2.1. *Given an instance of* MLIJ *with d machines and n copies of a job, we can get an instance of the problem with d machines and n jobs such that: for any load vector x' for this modified instance, the corresponding load vector x for the original instance satisfies*

$$\frac{1}{\sqrt{2}} x \leq x' \leq \sqrt{2} x.$$

Proof. To construct the new instance, round each p_i to its closest power of 2, say p'_i. Then $\frac{1}{\sqrt{2}} p'_i \leq p_i \leq \sqrt{2} p'_i$. When n_i jobs are scheduled on processor i, corresponding load vectors $x = (n_1 p_1, \ldots, n_d p_d)$ and $x' = (n_1 p'_1, \ldots, n_d p'_d)$ are within factor $\sqrt{2}$ of each other. ∎

Formally, a doubling instance is the one constructed in the proof: it has each p_i equal to some power of 2. We show next that for doubling instances, optimal load vector x^{OPT} for any norm always satisfies the order $x_1^{\mathrm{OPT}} \geq \ldots \geq x_d^{\mathrm{OPT}}$. Note that this is not true if the instance is not doubling; see Figure 2.

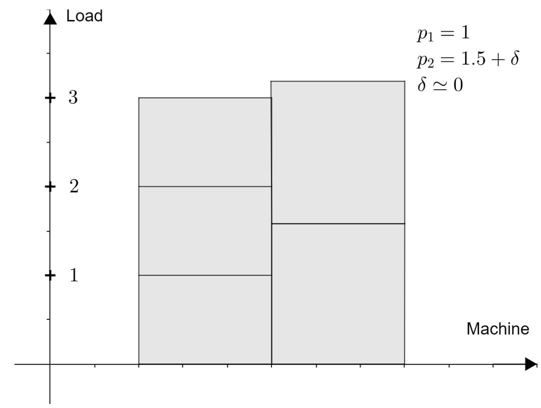

Figure 2: An example for makespan minimization with 2 machines and 5 jobs where $x_1^{\mathrm{OPT}} < x_2^{\mathrm{OPT}}$ for optimal load vector x^{OPT}.

LEMMA 2.2. *Suppose x^{OPT} is the optimal load vector for some symmetric monotonic norm $\|\cdot\|$ for a doubling instance. We can assume without loss of generality that $x_1^{\mathrm{OPT}} \geq x_2^{\mathrm{OPT}} \geq \ldots \geq x_d^{\mathrm{OPT}}$.*

Proof. Suppose $x_i^{\mathrm{OPT}} < x_{i+1}^{\mathrm{OPT}}$ for some i. Transfer one job from machine $i+1$ to machine i, to get the new load vector x defined as:

$$x_l = \begin{cases} x_l^{\mathrm{OPT}} & \text{if } l \neq i, i+1, \\ x_i^{\mathrm{OPT}} + p_i & \text{if } l = i, \\ x_{i+1}^{\mathrm{OPT}} - p_{i+1} & \text{if } l = i+1. \end{cases}$$

Since p_i divides p_{i+1} and $x_{i+1}^{\mathrm{OPT}} > x_i^{\mathrm{OPT}}$, we get that $x_{i+1}^{\mathrm{OPT}} - x_i^{\mathrm{OPT}} \geq p_i$. Therefore,

$$\max(x_i, x_{i+1}) = \max\left(x_i^{\mathrm{OPT}} + p_i, x_{i+1}^{\mathrm{OPT}} - p_{i+1}\right) \leq x_{i+1}^{\mathrm{OPT}} = \max\left(x_i^{\mathrm{OPT}}, x_{i+1}^{\mathrm{OPT}}\right).$$

Further, $x_i + x_{i+1} < x_i^{\mathrm{OPT}} + x_{i+1}^{\mathrm{OPT}}$. That is, $(x_i, x_{i+1}) \prec (x_i^{\mathrm{OPT}}, x_{i+1}^{\mathrm{OPT}})$. Since all other coordinates of x and x^{OPT} are equal, a simple inductive argument shows that $x \prec x^{\mathrm{OPT}}$. Lemma 1.3 implies that $\|x\| \leq \|x^{\mathrm{OPT}}\|$, finishing the proof. □

COROLLARY 2.1. *For* **Ord**, *an α-approximate portfolio for an instance of* MLIJ *can be obtained from a $\frac{\alpha}{2}$-approximate portfolio for the corresponding doubling instance.*

For the rest of this section, we restrict ourselves to doubling instances; we will give an $\alpha/2$-approximate portfolio of size $\leq 1 + \log_{\alpha/4} d$ for ordered norms over doubling instances. For any weight vector w, Lemma 2.2 allows us to relax the integer program (IP1) to a linear program: while not every load vector forms a feasible solution to IP1, Lemma 2.2 shows that there is an optimal solution that is feasible for this IP.

(IP1) $\quad \min w^\top x \quad$ s.t.

$$\sum_i \frac{x_i}{p_i} = n, \tag{2.2}$$

$$x_i \geq x_{i+1} \quad \forall\, i \in [d-1], \tag{2.3}$$

$$\frac{x_i}{p_i} \in \mathbb{Z}_{\geq 0} \quad \forall\, i \in [d], \tag{2.4}$$

(LP1) $\quad \min w^\top x \quad$ s.t.

$$\sum_i \frac{x_i}{p_i} = n, \tag{2.5}$$

$$x_i \geq x_{i+1} \quad \forall\, i \in [d-1], \tag{2.6}$$

$$x \geq 0. \tag{2.7}$$

Our next lemma characterizes the d vertices of the constraint polytope $\mathcal{P} := \{x : \sum_i \frac{x_i}{p_i} = n; x_1 \geq \ldots \geq x_d \geq 0\}$ of LP1. We omit the straightforward proof.

LEMMA 2.3. *For any weight vector w, the optimal solution x^* to LP1 satisfies for some $l \in [d]$ that:*

$$x_1^* = \ldots = x_l^* = \frac{n}{\sum_{i \in [l]} \frac{1}{p_i}}, \quad x_{l+1}^* = \ldots = x_d^* = 0.$$

For $l \in [d]$, denote the lth vertex as $x(l) := \frac{n}{\sum_{i\in[l]} \frac{1}{p_i}} \mathbf{1}_l$, with l non-zero entries. Call $x(l)$ *good* if

$$\frac{n}{\sum_{i \in [l]} \frac{1}{p_i}} \geq p_l, \tag{2.8}$$

i.e., the value of each non-zero coordinate is at least the processing time corresponding to the last non-zero coordinate. Clearly, $x(1)$ is good since $n \geq 1$, and if $x(l)$ is good then $x(l-1)$ is also good. Let L be the largest index such that $x(L)$ is good. The next lemma says that if $x(l)$ is good, then it can be rounded to an integral load vector:

LEMMA 2.4. *If $x(l)$ is good, then it can be rounded to $\hat{x}(l)$ that is feasible for IP1 and $\frac{1}{2}x(l) \leq \hat{x}(l) \leq 2x(l)$.*

Proof. Denote $n_i = \frac{x(l)_i}{p_i}$ for all $i \in [d]$, then $n_{l+1} = \ldots = n_d = 0$ and $\sum_{i \in [d]} n_i = n$. Then one can assign either $\hat{n}_i = \lfloor n_i \rfloor$ or $\hat{n}_i = \lceil n_i \rceil$ jobs to machine $i \in [d]$, while ensuring that $\sum_{i \in [d]} \hat{n}_i = n$. The load on machine $i \in [d]$ in this new schedule is $\hat{x}(l)$, with $\hat{x}(l)_i = p_i \hat{n}_i$.

By definition of good vertices, $x(l)_i \geq p_l \geq p_i$ for each $i \in [l]$. Therefore, we get $n_i \geq 1$, thus implying $\frac{1}{2}n_i \leq \lfloor n_i \rfloor \leq n_i$ and $n_i \leq \lceil n_i \rceil \leq 2n_i$ for all $i \in [l]$. This implies $\frac{1}{2}n_i \leq \hat{n}_i \leq 2n_i$ for all $i \in [d]$. Since $n_i = \frac{x(l)_i}{p_i}$ and $\hat{n}_i = \frac{\hat{x}(l)_i}{p_i}$, we get the result. ∎

Our next lemma shows that rounding good vertices gives a 2-approximate portfolio for ordered norms:

LEMMA 2.5. $\{\hat{x}(1), \ldots, \hat{x}(L)\}$ *is a 2-approximate portfolio for* **Ord** *over the doubling instance.*

Proof. Fix a weight vector w. Let x^{OPT} be the (integral) optimal load vector for $\|\cdot\|_{(w)}$, and let l be the largest index such that $x^{\text{OPT}}_l > 0$. We will first show that there exists an index $l' \leq l$ such that (i) $x(l')$ is good, and (ii) $\|x(l')\|_{(w)} \leq \|x^{\text{OPT}}\|_{(w)}$. Together with Lemma 2.4, this implies that $\|\hat{x}(l')\|_{(w)} \leq 2\|x^{\text{OPT}}\|_{(w)}$, implying the lemma.

We note first that $x(l)$ is good: since x^{OPT} is integral and $x^{\text{OPT}}_l \neq 0$, we have $x^{\text{OPT}}_l \geq p_l$. From Lemma 2.2, we have $x^{\text{OPT}}_1 \geq \ldots \geq x^{\text{OPT}}_l \geq p_l$. Since $\sum_{i \in [l]} \frac{x^{\text{OPT}}_i}{p_i} = n$, we get $n \geq \sum_{i \in [l]} \frac{p_l}{p_i} = p_l \sum_{i \in [l]} \frac{1}{p_i}$. That is, $x(l)$ is good.

In particular, this implies that $x(l')$ is good for each $l' \leq l$, so it is now sufficient to show that there is some $l' \leq l$ such that $\|x(l')\|_{(w)} \leq \|x^{\text{OPT}}\|_{(w)}$. Consider the following linear program:

$$\text{(LP2)} \qquad \min w^\top x \qquad \text{s.t.}$$

$$\text{(2.9)} \qquad \sum_i \frac{x_i}{p_i} = n,$$

$$\text{(2.10)} \qquad x_i \geq x_{i+1} \qquad \forall\, i \in [d-1],$$

$$\text{(2.11)} \qquad x_{l+1} = \ldots = x_d = 0.$$

x^{OPT} is feasible for this LP by assumption. Further, by an argument similar to Lemma 2.3, we get that the vertices of the constraint polytope for this LP are $x(1), \ldots, x(l)$. Therefore, there is some $l' \leq l$ such that $\|x(l')\|_{(w)} = w^\top x(l') \leq w^\top x^{\text{OPT}} = \|x^{\text{OPT}}\|_{(w)}$, finishing the proof. ∎

We are now ready to prove Theorem 2.1. We will convert the 2-approximate portfolios of size d for doubling instances to an $\alpha/2$-approximate portfolio of size $\sim \log_{\alpha/4} d$, which implies α-approximate portfolios of size $\sim \log_{\alpha/4} d$ for MLIJ by Corollary 2.1.

Proof. [Proof of Theorem 2.1] We claim that for all indices $l, i \in [d]$ such that $i \leq \frac{\alpha}{4}l$, we have $x(l) \preceq \frac{\alpha}{4}x(i)$. Therefore, $\|x(l)\|_{(w)} \leq \frac{\alpha}{4}\|x(i)\|_{(w)}$ for all ordered norms $\|\cdot\|_{(w)}$ from Lemma 1.3, implying that $\left\{x((\alpha/4)^j) : j \in [0, 1 + \log_{(\alpha/4)} L]\right\}$ is an $(\alpha/2)$-approximate portfolio over doubling instances.

Since $p_1 \leq \ldots \leq p_d$ and $i \leq \frac{\alpha}{4}l$, we have $\sum_{j \in [l]} \frac{1}{p_j} \geq \frac{4}{\alpha} \sum_{j \in [i]} \frac{1}{p_j}$. Therefore, for all $k \leq l$, we have

$$\sum_{j \in [k]} x(l)_j = \frac{kn}{\sum_{i \in [l]} \frac{1}{p_j}} \leq \frac{\alpha}{4} \cdot \frac{kn}{\sum_{j \in [i]} \frac{1}{p_j}} = \frac{\alpha}{4} \cdot \sum_{j \in [k]} x(i)_k.$$

Further, for $k > l$,

$$\sum_{j \in [k]} x(l)_j = \sum_{j \in [l]} x(l)_j = \frac{nl}{\sum_{j \in [l]} \frac{1}{p_j}} \leq \frac{\alpha}{4} \frac{nl}{\sum_{j \in [i]} \frac{1}{p_j}} \leq \frac{\alpha}{4} \sum_{j \in [i]} x(i)_j \leq \frac{\alpha}{4} \sum_{j \in [k]} x(i)_j.$$

Therefore, $x(l) \preceq (\alpha/4)x(i)$. This completes the proof. ∎

2.2 Portfolio Lower Bound. We prove Theorem 2.2 by giving an appropriate doubling instance with d machines where any α-approximate portfolio must have size $O\left(\frac{\log d}{\log \alpha + \log\log d}\right)$. Given d, let $S = S(d)$ be a superconstant that we specify later; assume that S is an integer that is a power of 2. Let L be the largest integer such that $1 + S^2 + \ldots + S^{2L} \le d$, then $L = \Theta(\log_S d)$. The d machines are divided into $L+1$ classes from 0 to L: there are S^{2l} machines in the lth class and the processing time on these machines is $p_l = S^l$. The number of jobs n is S^{3L}; it is chosen so as to ensure that all vertices in the constraint polytope for LP1 are good, and can be rounded to an integral solution that is only worse by a factor at most 2 (Lemma 2.4).

There are $L+1$ weight vectors for our instance. The first weight vector is $w(0) = (1, 1, \ldots, 1)$. The second weight vector is $w(1) = \left(1, \frac{1}{S^2}, \frac{1}{S^2}, \ldots, \frac{1}{S^2}\right)$. More generally, for $l \in [0, L]$,

$$w(l) = \Bigg(1, \underbrace{\frac{1}{S^2}, \ldots, \frac{1}{S^2}}_{S^2}, \underbrace{\frac{1}{S^4}, \ldots, \frac{1}{S^4}}_{S^4}, \ldots, \underbrace{\frac{1}{S^{2l-2}}, \ldots, \frac{1}{S^{2l-2}}}_{S^{2l-2}}, \underbrace{\frac{1}{S^{2l}}, \ldots, \frac{1}{S^{2l}}}_{\text{remaining}}\Bigg).$$

With some foresight, we choose S such that $\frac{S}{L} = 5\alpha$. We claim the following: for each $l \in [0, L-1]$,

1. There is a schedule $\hat{x}(l)$ for this instance with $\|\hat{x}(l)\|_{(w(l))} \le nlS^{-l}$.
2. Any schedule y that schedules more than $n/4$ jobs on machines in classes $l+1$ to L has $\|y\|_{(w(l))} \ge \frac{nS}{4} \cdot S^{-l}$. Combined with the above and since $\alpha \le \frac{S}{4L}$, it cannot be an α-approximation for the $w(l)$-norm problem.
3. Any schedule y that schedules more than $n/4$ jobs on machines in classes 0 to $l-1$ has $\|y\|_{(w(l))} \ge \frac{nS}{2} \cdot S^{-l}$. Therefore, it cannot be an α-approximation for the $w(l)$-norm problem either.
4. $L = \Theta(\log_S d) = \Omega\left(\frac{\log d}{\log \alpha + \log\log d}\right)$.

Claims 1, 2, and 3 imply that any α-approximate solution for norm $w(l)$ must schedule at least $n/2$ jobs on machines in class l. Another application of claims 2 and 3 then implies that a portfolio that is α-approximate for weight vectors $\{w(0), \ldots, w(L-1)\}$ must have distinct solutions for each weight vector, and therefore has size at least L. Claim 4 then implies our theorem.

Claim 4 is just computation: $L = \Theta(\log_S d) = \Theta(\log_{\alpha L} d) = \Theta\left(\frac{\log d}{\log \alpha + \log L}\right)$. If $L = \Omega(\log d)$, then we are done since the target size is anyway $\Theta\left(\frac{\log d}{\log \alpha + \log\log d}\right) = O(\log d)$ for constant α. Otherwise, $\log L = O(\log\log d)$ and so $L = \Theta\left(\frac{\log d}{\log \alpha + \log L}\right) = \Theta\left(\frac{\log d}{\log \alpha + \log\log d}\right)$.

We move to claim 1. As alluded to before, $n = S^{3L}$ has been chosen so that each vertex $x(l)$ of the constraint polytope is good (see inequality (2.8)):

$$\frac{n}{1 \cdot \frac{1}{1} + S^2 \cdot \frac{1}{S} + \ldots + S^{2L} \cdot \frac{1}{S^L}} \ge \frac{n}{2S^L} \ge S^L = p_L.$$

With this in hand, it is sufficient to give a fractional solution $x(l)$ with $\|x(l)\|_{(w(l))} = \Theta(nlS^{-l})$, since Lemma 2.4 then implies the existence of an integral solution $\hat{x}(l)$ with norm at most twice. Consider $x(l) = (a, \ldots, a, 0, \ldots, 0)$ where the first $1 + S^2 + \ldots + S^{2l}$ coordinates are non-zero and equal to a; all other coordinates are 0. Since a total of n jobs must be scheduled (constraint (2.9)),

$$n = a\left(1 \cdot \frac{1}{1} + S^2 \cdot \frac{1}{S} + \ldots + S^{2l} \cdot \frac{1}{S^l}\right) \ge aS^l,$$

so that $a \le \frac{n}{S^l}$. Therefore,

$$\|x(l)\|_{(w(l))} = a \times \text{ sum of first } (1 + S^2 + \ldots + S^{2l}) \text{ coordinates of } w(l) = a \cdot l \le nlS^{-l}.$$

We move to claim 2. Let y schedule more than $n/4$ jobs on machines in classes $l+1$ to L. Irrespective of how these $n/4$ jobs are distributed, they contribute a total load of at least $(n/4) \times S^{l+1}$. Since all coordinates of $w(l)$ are at least $\frac{1}{S^{2l}}$, the contribution of these jobs to $\|y\|_{(w(l))}$ is at least

$$\frac{1}{S^{2l}} \times \frac{n}{4} S^{l+1} = \frac{nS}{4} \cdot S^{-l}.$$

Since $l \leq L = o(S)$, we get $\|y\|_{(w(l))} = \omega(nlS^{-l})$.

Finally, we prove claim 3. Consider the restricted instance with only machines from classes $0, \ldots, l-1$ and $n/4$ jobs. Let x be the optimal fractional solution for this instance for top-1 norm $\|\cdot\|_{(\mathbf{1}_1)}$; it is easy to see that x must have equal loads on machines, so that from constraint (2.9):

$$n = \|x\|_{(\mathbf{1}_1)} \left(1 \cdot \frac{1}{1} + S^2 \cdot \frac{1}{S} + \ldots + S^{2l-2} \cdot \frac{1}{S^{l-1}}\right) \leq 2\|x\|_{(\mathbf{1}_1)} S^{l-1},$$

implying $\|x\|_{(\mathbf{1}_1)} \geq \frac{nS^{-l+1}}{2}$. Therefore, any integral optimal solution $\hat{x}$ to this restricted instance must also satisfy

$$\|\hat{x}\|_{(\mathbf{1}_1)} \geq \|x\|_{(\mathbf{1}_1)} \geq \frac{nS^{-l+1}}{2}.$$

Since y is a solution to the larger original instance, we have $\|y\|_{(\mathbf{1}_1)} \geq \|\hat{x}\|_{(\mathbf{1}_1)}$. Finally, since $w(l) = 1$ by assumption, we get $\mathbf{1}_1 \preceq w(l)$, and so $\|y\|_{(w(l))} \geq \|y\|_{(\mathbf{1}_1)}$. Together, we get $\|y\|_{(w(l))} \geq \frac{nS}{2} \cdot S^{-l}$. This completes the proof of the claim and of Theorem 2.2.

Portfolios for Different Classes of Norms. Recall Observation 1: if x^* is a simultaneous α-approximation for each top-k norm, then it is a simultaneous α-approximation for all symmetric monotonic norms. One might naturally wonder if this is true for portfolios: is an α-approximate portfolio for top-k norms also an α-approximate portfolio for all symmetric monotonic norms? Our lower bound on portfolio sizes for ordered norms (see Appendix B) along with $O(\log d)$ upper bounds on portfolio sizes for top-k norms (Observation 2) already implies that this is not the case. We give another proof using the instance constructed for portfolio lower bound for MLIJ. The proof is deferred to Appendix C.

Theorem 2.3. *For all large enough d, there exists a set of vectors $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ such that:*

1. *there is an $O(1)$-approximate portfolio X of size 2 for* **Top***, and*
2. *any $O(1)$-approximate portfolio X' for* **Ord** *has size $\widetilde{\Omega}\left(\frac{\log d}{\log\log d}\right)$.*

3 `OrderAndCount` for Covering-Polyhedra

In this section, we extend `OrderAndCount` to Covering-Polyhedra in d-dimensions, which is defined by $\mathcal{P} = \{x \in \mathbb{R}^d : Ax \geq b, x \geq 0\}$ with nonnegative constraint matrix $A \in \mathbb{R}^{r \times d}_{\geq 0}$ with r rows and $b \in \mathbb{R}^r_{\geq 0}$. As alluded to before, such polyhedra can model workload management in settings with r splittable jobs split among d machines that can run all r jobs concurrently. MLIJ corresponds to $r = 1$ since it had a single constraint of the form $a^\top x \geq b$. We prove the following theorem that upper bounds the portfolio size for covering polyhedra:

Theorem 3.1. *Given $\epsilon \in (0, 1]$ and a covering polyhedron $\mathcal{P} = \{x \in \mathbb{R}^d : Ax \geq b, x \geq 0\}$ where $A \in \mathbb{R}^{r \times d}_{\geq 0}$ and $b \in \mathbb{R}^d_{\geq 0}$, there is an algorithm to compute a portfolio $X_{1+\epsilon}$ of size*

$$|X_{1+\epsilon}| = O\left(\left(\frac{\log(d/\epsilon)}{\epsilon}\right)^{3r^2-2r}\right)$$

that is $(1+\epsilon)$-approximate for ordered norms **Ord***. This portfolio can be computed in time $\mathrm{poly}(d, \log^{r^2}(d)))$. Consequently, using Lemma 1.4, there is a polynomial-time algorithm to compute a portfolio of size $O\left((\log d)^{3r^2-2r}\right)$ that is $O(\log d)$-approximate for symmetric monotonic norms* **Sym***.*

We focus on the result for **Ord** since the result for **Sym** follows from Lemma 1.4. Further, we assume that $b = \mathbf{1}_r = (1, \ldots, 1)^\top$, without loss of generality by rescaling rows of A if necessary (and removing rows with $b = 0$ since they will be feasible anyway).

For any order π on $[d]$, define the restriction $\mathcal{P}_\pi := \mathcal{P} \cap \{x \in \mathbb{R}^d : x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0\}$. Recall our high-level plan: any ordered norm $\|\cdot\|_{(w)}$ is a linear function on each $\mathcal{P}_\pi$. Therefore, the minimum norm point $x(w) := \arg\min_{x \in \mathcal{P}} \|x\|_{(w)}$ must be one of the vertices of some $\mathcal{P}_\pi$. Call X the union of vertices of $\mathcal{P}_\pi$ across all orders π; then X is an *optimal* portfolio for **Ord**. As we outlined, two main issues potentially blow up the size $|X|$:

1. Each $\mathcal{P}_\pi$ can have too many vertices. For each vertex of $\mathcal{P}_\pi$, d out of $r+d$ constraints $Ax \geq \mathbf{1}_r, x_{\pi(1)} \geq \ldots \geq x_{\pi(d)} \geq 0$ must be tight. Therefore, $\mathcal{P}_\pi$ may have $\binom{d+r}{d} \sim d^r$ vertices.

2. There are $d!$ orders $\pi \in \mathrm{Perm}(d)$. Since we are taking a union over all such orders, the size $|X|$ is bounded by:

$$\binom{\text{number of vertices}}{\text{in each } \mathcal{P}_\pi} \times \binom{\text{number of orders}}{\pi} \sim d^r \times d!.$$

We start with sparsification (Section 3.1) that aims to reduce the number of unique coordinates of each $x(w)$ to $\left(\frac{\log(d/\epsilon)}{\epsilon}\right)^r$; showing that this bound the first factor above to $\left(\frac{\log(d/\epsilon)}{\epsilon}\right)^{r^2}$. Bounding the second factor requires bounding the number of orders that $x(w)$ can satisfy, which we accomplish using our primal-dual counting technique (Section 3.2).

Algorithm 1 `SparsifyPolyhedron`($\mathcal{P}$)

input: covering polyhedron $\mathcal{P} = \{x \in \mathbb{R}^d : Ax \geq \mathbf{1}_r, x \geq 0\}$, error parameter $\epsilon \in (0,1]$
output: another covering polyhedron $\widetilde{\mathcal{P}} = \{x \in \mathbb{R}^d : \widetilde{A}x \geq \mathbf{1}_r, x \geq 0\}$

define $\mu = \frac{3d^2}{\epsilon}$ and initialize $\widetilde{A} = \mathbf{0}_{r\times d}$
for each row $i \in [r]$ **do**
  define $a_i^* = \max_{j\in[d]} A_{i,j}$ to be the largest entry in the row
  for column $j \in [d]$ **do**
    if $A_{i,j} < \frac{a_i^*}{\mu}$ **then**
      set $\widetilde{A}_{i,j} = 0$
    else
      let $l \in [0, \lfloor \log_{(1+\epsilon/2)} \mu \rfloor]$ be the unique integer such that

$$\frac{a_i^*}{\mu}\left(1+\frac{\epsilon}{2}\right)^l \leq A_{i,j} < \frac{a_i^*}{\mu}\left(1+\frac{\epsilon}{2}\right)^{l+1}$$

      set $\widetilde{A}_{i,j} = \frac{a_i^*}{\mu}\left(1+\frac{\epsilon}{2}\right)^l$
return $\widetilde{A}$, $\widetilde{P} = \{x \in \mathbb{R}^d : \widetilde{A}x \geq \mathbf{1}_r, x \geq 0\}$

3.1 Sparsification. Denote $N = O\big(\frac{\log(d/\epsilon)}{\epsilon}\big)$. We give a sparsification procedure (Algorithm 1, `SparsifyPolyhedron`) that reduces the number of distinct columns in A to N^r. For each row of matrix A, this sparsification (1) removes 'small' entries in the row and (2) restricts the number of unique entries in the row to N. Since there are r rows, the number of distinct columns after sparsification is N^r. In the process, we lose a factor $(1+\epsilon)$ in the approximation.

LEMMA 3.1. *The columns of matrix $\widetilde{A} \in \mathbb{R}^{r\times d}_{\geq 0}$ output by Algorithm* `SparsifyPolyhedron` *take one of N^r values, i.e., $[d]$ can be partitioned into $S_1, \ldots, S_{N^r}$ such that for any $j, j' \in S_l$, the jth and j'th columns of $\widetilde{A}$ are equal.*

Proof. Fix row $i \in [r]$. By construction, each entry in the ith row of $\widetilde{A}$ is in the set $\{0\} \cup \left\{\frac{a_i^*}{\mu}\left(1+\frac{\epsilon}{2}\right)^l : l \in [0, \lfloor \log_{(1+\epsilon/2)} \mu \rfloor]\right\}$. These are $O(\log_{(1+\epsilon/2)} \mu) = O(\log_{(1+\epsilon/2)}(d^2/\epsilon)) = O\left(\frac{\log(d/\epsilon)}{\epsilon}\right) = N$ distinct numbers. Since each column is composed of r entries, one from each row, we get a total of N^r possible values for a column. ∎

The next lemma gives the approximation guarantee for the sparsified polyhedron; its proof is relatively straightforward and we defer it to Appendix D.

LEMMA 3.2. *$\widetilde{\mathcal{P}} = \{x : \widetilde{A}x \geq \mathbf{1}_r, x \geq 0\}$ output by Algorithm* `SparsifyPolyhedron` *is a $(1+\epsilon)$-approximate portfolio for* **Sym** *over $\mathcal{P}$.*

Lemma 3.2 allows us to work with $\widetilde{P} = \{x : \widetilde{A}x \geq \mathbf{1}_r, x \geq 0\}$ with the nice property that columns of $\widetilde{A}$ take at most N^r distinct values (from Lemma 3.1). We will give an optimal portfolio for **Ord** over $\widetilde{\mathcal{P}}$. This portfolio will have size $O(N^{3r^2-2r})$. Using Lemma 1.5, this is sufficient to prove Theorem 3.1. Hereafter, we will only work with the sparsified matrix $\widetilde{A}$ and polyhedron $\widetilde{\mathcal{P}}$. For ease of notation, we drop the symbol $\widetilde{A}$ and assume that the original matrix A and corresponding polyhedron $\mathcal{P}$ are already given to us in the sparsified form.

Let $S_1, \ldots, S_{N^r}$ denote the partition of $[d]$ based on the value of columns of A, i.e., for each $l \in [N^r]$ and $j, j' \in S_l$, jth and j'th columns of A are the same. Further, define $\mathcal{Q} = \{x \in \mathbb{R}^d_{\geq 0} : x_j = x_{j'} \forall j, j' \in S_l, \forall l \in [N^r]\}$, i.e., the set of all non-negative vectors that attain the same value for all $j \in S_l$, for all $l \in [N^r]$. Define $\mathcal{P}^= = \mathcal{P} \cap \mathcal{Q}$. Recall that for weight vector w, we define $x(w) := \arg\min_{x \in \mathcal{P}} \|x\|_{(w)}$. Our first lemma shows that $x(w) \in \mathcal{P}^=$:

Lemma 3.3. *Given a weight vector w, we can assume without loss of generality that for all $l \in [N^r]$ and $j, j' \in S_l$, $x(w)_j = x(w)_{j'}$. That is, $\mathcal{P}^=$ is an optimal portfolio for* **Sym** *over $\mathcal{P}$.*

Proof. Suppose $x(w)_j \neq x(w)_{j'}$, say $x(w)_j > x(w)_{j'}$. Then consider $\overline{x} \in \mathbb{R}^d$ such that $\overline{x}_k = x(w)_k$ for all $k \neq j, j'$, and $\overline{x}_j = \overline{x}_{j'} = \frac{x(w)_j + x(w)_{j'}}{2}$. Then it can be seen that $\overline{x} \preceq x(w)$ and so by Lemma 1.3 we get that $\|\overline{x}\|_{(w)} \leq \|x(w)\|_{(w)}$.

It remains to show that $\overline{x} \in \mathcal{P}$. Clearly $\overline{x} \geq 0$ since $x(w) \geq 0$. Denote by $A^{(j)}, A^{(j')}$ the jth, j'th columns of A. Since they are equal,

$$A(x(w) - \overline{x}) = A^{(j)}(x(w)_j - \overline{x}_j) + A^{(j')}(x(w)_{j'} - \overline{x}_{j'}) = A^{(j)}((x(w)_j + x(w)_{j'}) - (\overline{x}_j + \overline{x}_{j'})) = 0.$$

Therefore, $A\overline{x} = Ax(w) \geq \mathbf{1}_r$, or that $\overline{x} \in \mathcal{P}$. ∎

Given the above lemma, it is now sufficient to consider orders over $[N^r]$ instead of orders over $[d]$. We call these *reduced orders*:

Definition 3. (Reduced orders) *An order ρ on $[N^r]$ is called a* reduced order. *For $x \in \mathcal{Q}$, define vector $z(x) \in \mathbb{R}^{N^r}$ of unique coordinates of x, i.e., for $l \in [N^r]$, define $z(x)_l = x_j$ for $j \in S_l$. $x \in \mathcal{Q}$ is said to* satisfy *reduced order ρ if $z_{\rho(1)} \geq \ldots \geq z_{\rho(N^r)} \geq 0$. Given a reduced order ρ, define polyhedron*

$$\mathcal{P}^=_\rho = \{x \in \mathcal{P} \cap \mathcal{Q} : x \text{ satisfies reduced order } \rho\}.$$

Suppose now that we are given some reduced order ρ. Then for $x \in \mathcal{P}^=_\rho$, $\|x\|_{(w)}$ is a linear function of x. Therefore, given a weight vector w, if $x(w)$ satisfies reduced order ρ, then $x(w)$ is one of the vertices of polyhedron $\mathcal{P}^=_\rho$. With this observation, the rest of the proof is organized as follows:

- For each reduced order ρ, $\mathcal{P}^=_\rho$ has at most $N^{r^2} + 1$ vertices (Lemma 3.4).
- Consider the set Π of reduced orders such that for any weight vector w, $x(w)$ satisfies some reduced order $\rho \in \Pi$, i.e, $\Pi = \{\text{reduced order } \rho : \exists\, w \text{ where } x(w) \text{ satisfies } \rho\}$. Then we will show that $|\Pi| \leq N^{2r(r-1)}$ (Lemma 3.5).

Together, these observations mean that $X := \bigcup_{\rho \in \Pi} \left(\text{vertices of } \mathcal{P}^=_\rho\right)$ is an optimal portfolio for **Ord** over $\mathcal{P}^=$. By Lemma 3.3, $\mathcal{P}^=$ is an optimal portfolio for **Ord** over $\mathcal{P}$. Therefore, Lemma 1.5 implies that X is an optimal portfolio for **Ord** over $\mathcal{P}$. Further,

$$\begin{aligned} |X| &= \Big|\bigcup_{\rho \in \Pi} \left(\text{vertices of } \mathcal{P}^=_\rho\right)\Big| \leq \sum_{\rho \in \Pi} \left|\left(\text{vertices of } \mathcal{P}^=_\rho\right)\right| \\ &\leq \sum_{\rho \in \Pi} (N^{r^2} + 1) = |\Pi|(N^{r^2} + 1) \leq N^{2r(r-1)}(N^{r^2} + 1) = O(N^{3r^2 - 2r}). \end{aligned}$$

This implies Theorem 3.1. We prove Lemma 3.4 next and defer Lemma 3.5 to the next subsection.

Lemma 3.4. *For each reduced order ρ, $\mathcal{P}^=_\rho$ has at most $N^{r^2} + 1$ vertices*

Proof. For simplicity, assume (after possibly relabeling indices) that $\rho(l) = l$ for all $l \in [N^r]$, and that $S_1 = \{1, \ldots, |S_1|\}, S_2 = \{|S_1|+1, \ldots, |S_1|+|S_2|\}$ etc. Then the polyhedron $\mathcal{P}^{=}_{\rho}$ is the set of all x such that $A_i^\top x \geq 1$ for all $i \in [r]$ and

$$x_1 = \ldots = x_{|S_1|} \geq x_{|S_1|+1} = \ldots = x_{|S_1|+|S_2|} \geq \ldots \geq x_{d-|S_{N^r}|+1} = \ldots = x_d \geq 0.$$

Any vertex corresponds to a set of d (linearly independent) inequalities. The constraints of the polytope have $d - N^r$ equalities and $N^r + r$ inequalities. Therefore, each vertex corresponds to some N^r of the $N^r + r$ inequalities being tight. The number of such choices is $\binom{N^r+r}{N^r}$. Then,

$$\binom{N^r + r}{N^r} = \binom{N^r + r}{r} \leq \left(1 + \frac{N^r}{r}\right)^r.$$

For $r = 1$, this is at most $1 + N^r$. For $r \geq 2$, $1 + \frac{N^r}{r} \leq N^r$ and so this is at most N^{r^2}. ∎

At this point, a natural first attempt at bounding the portfolio size is to count the number of ordered norms in the space of 'reduced' vectors $\{z(x) : x \in \mathcal{P}^{=}\} \subseteq \mathbb{R}^{N^r}$. After all, the result from [6] result shows that there are at most $\text{poly}(N^{r/\epsilon})$ ordered norms in $\mathbb{R}^{N^r}$ up to a $(1+\epsilon)$-approximation. However, to the best of our knowledge, this approach does not directly work because ordered norms on $\mathbb{R}^d$ cannot be translated appropriately into an ordered norm on the smaller space $\mathbb{R}^{N^r}$.

For example, consider the covering polyhedron $\mathcal{P} = \{x \in \mathbb{R}^3_{\geq 0} : x_1 \geq 2, x_2 + x_3 \geq 4, 2x_1 + x_2 + x_3 \geq 10\}$. The point $(3, 2, 2) \in \mathcal{P}$ is the (unique) minimizer of the L_1 norm, which corresponds to weight vector $w = (1, 1, 1)$. The constraint polytope for $\mathcal{P}$ has two unique columns, and the corresponding 'reduced covering polyhedron' is $\mathcal{P}' = \{z \in \mathbb{R}^2 : z_1 \geq 2, z_2 \geq 2, z_1 + z_2 \geq 5\}$. A point $(a, b, b) \in \mathcal{P}$ corresponds to the point $(a, b) \in \mathcal{P}'$. However, by a majorization argument, the point $(5/2, 5/2) \in \mathcal{P}'$ minimizes *all ordered norms* on $\mathcal{P}'$, but the corresponding point $(5/2, 5/2, 5/2) \in \mathcal{P}$ with L_1 norm 7.5 is sub-optimal for the L_1 norm. Therefore, it is not sufficient to count ordered norms in $\mathbb{R}^{N^r}$, and we need an alternate approach that we describe next.

3.2 Primal-Dual Counting. In this section, we study the set Π of reduced orders such that for any weight vector w, $x(w)$ satisfies some reduced order $\rho \in \Pi$, i.e, $\Pi = \{\text{reduced order } \rho : \exists\, w \text{ where } x(w) \text{ satisfies } \rho\}$. We prove that

LEMMA 3.5. $|\Pi| \leq N^{2r(r-1)}$.

The main idea is to count reduced orders not on $x(w)$, but in a *dual space*. We write the following modified primal and dual, and denote $\lambda(w) = \arg\min_{\lambda \in \mathbf{\Delta}_r} \|A^\top \lambda\|^*_{(w)}$:

(Primal') $\quad \min \|x\|_{(w)}$ s.t. $Ax \geq \mathbf{1}_r, x \in \mathcal{Q}$. $\qquad$ (Dual) $\quad \min \|A^\top \lambda\|^*_{(w)}$ s.t. $\lambda \in \mathbf{\Delta}_r$

Note that $(A^\top \lambda)_j$ is simply the dot product of the jth column of A with λ. Further, recall for all $j, j' \in S_l$ for any $l \in [N^r]$, the jth and j'th columns of A are equal. Therefore, we have $(A^\top \lambda)_j = (A^\top \lambda)_{j'}$ for any λ. By definition, this means that $A^\top \lambda \in \mathcal{Q}$ for all $\lambda \geq 0$.

The next lemma establishes the crucial connection between reduced orders in Primal' and Dual. It uses Lemma 1.2 (Ordered Cauchy-Schwarz) along with a Lagrangian function; we defer its proof to Appendix D.

LEMMA 3.6. *Given a weight vector w, $\|x(w)\|_{(w)} \|A^\top \lambda(w)\|^*_{(w)} = 1$. Further, there is a reduced order ρ such that both $x(w), A^\top \lambda(w)$ satisfy ρ.*

As a consequence of this lemma, we get that it is sufficient to count reduced orders in the dual:

$$\begin{aligned} \Pi &= \{\text{reduced order } \rho : \exists\, w \text{ where } x(w) \text{ satisfies } \rho\} \\ &= \{\text{reduced order } \rho : \exists\, w \text{ where } A^\top \lambda(w) \text{ satisfies } \rho\} \\ &\subseteq \{\text{reduced order } \rho : \exists\, \lambda \in \mathbf{\Delta}_r \text{ where } A^\top \lambda \text{ satisfies } \rho\}. \end{aligned}$$

Denote $\Pi^* = \{\text{reduced order } \rho : \exists\, \lambda \in \mathbf{\Delta}_r \text{ where } A^\top \lambda \text{ satisfies } \rho\}$; our goal is to show that $|\Pi^*| \leq N^{2r(r-1)}$. From the above, this is sufficient to prove Lemma 3.5. Our final lemma is a geometric counting inequality.

Lemma 3.7. *T hyperplanes partition $\mathbf{\Delta}_r$ into at most $T^{r-1}+1$ regions.*

Proof. The result is trivially true for $r = 1$ since $\mathbf{\Delta}_1$ is a point. For $r = 2$, $\mathbf{\Delta}_2$ is a line segment, and T 'hyperplanes' partition it into $\leq T+1$ regions. For $r \geq 3$, we use induction on T. 1 hyperplane clearly divides any convex body into at most $2 \leq 1^{r-1}+1$ regions. Suppose $T > 1$. Let the Tth hyperplane be $\mathcal{H}$. By the induction hypothesis, the first $T-1$ hyperplanes divide $\mathbf{\Delta}_r$ into at most $(T-1)^{r-1}+1$ regions. If $\mathbf{\Delta}_r \subseteq \mathcal{H}$, then $\mathcal{H}$ does not add any new regions, and we are done.

Otherwise, the number of new regions $\mathcal{H}$ adds is the number of regions that the first $T-1$ hyperplanes partition $\mathbf{\Delta}_r \cap \mathcal{H}$ into. But $\mathbf{\Delta}_r \cap \mathcal{H}$ can be linearly transformed into $\mathbf{\Delta}_{r-1}$ in this case, and so the number of new regions is at most $(T-1)^{r-2}+1$. Therefore, by the induction hypothesis, the total number of regions with T hyperplanes is at most

$$((T-1)^{r-1}+1) + ((T-1)^{r-2}+1) \leq T^{r-1}+1 \quad \forall\, T \geq 1, r \geq 3.$$

□

We are ready to finish the proof of Lemma 3.5. Partition $\mathbf{\Delta}_r$ into regions $\{R_\rho : \rho \in \Pi^*\}$, where $R_\rho := \{\lambda \in \mathbf{\Delta}_r : A^\top\lambda \text{ satisfies } \rho\}$. The size $|\Pi^*|$ is exactly the number of such regions. Pick $j, j' \in [d]$ such that j, j' belong to different sets $S_l, S_{l'}$. Then these regions are separated by hyperplanes of the form $\{\lambda : (A^\top\lambda)_j = (A^\top\lambda)_{j'}\}$, i.e., different reduced orders exist on different sides of these hyperplanes. There are $\binom{N^r}{2}$ such hyperplanes, each corresponding to a pair of sets $S_l, S_{l'}$. By the above lemma, these partition $\mathbf{\Delta}_r$ into at most

$$\binom{N^r}{2}^{r-1} + 1 = \left(\frac{N^r(N^r-1)}{2}\right)^{r-1} + 1 \leq N^{2r(r-1)}.$$

regions. Thus, $|\Pi| \leq |\Pi^*| = |\{R_\rho : \rho \in \Pi^*\}| \leq N^{2r(r-1)}$. This finishes the proof of Lemma 3.5, and therefore the proof of Theorem 3.1.

We finally remark that this can be converted into an algorithm that runs in time $\text{poly}(N^{r^2}, d)$: tracing back, find the set Π^* using the above hyperplane argument, and then simply output the union of vertices of $\mathcal{P}_\rho^=$ for all $\rho \in \Pi^*$.

4 k-Clustering and Uncapacitated-Facility-Location

In this section, we consider k-Clustering and Uncapacitated-Facility-Location. Recall that we are given a metric space (X, dist) on $|X| = n$ points (also called *clients*) and are required to choose a subset $F \subseteq X$ of *open facilities*. The induced distance vector $x_F \in \mathbb{R}^X$ is defined as the vector of distances between point j and its nearest open facility, i.e., $x_F(j) = \min_{f \in F} \text{dist}(j, f)$ for all $j \in X$. Given a norm $\|\cdot\|$ on $\mathbb{R}^n$, k-Clustering seeks to open set F of at most k facilities to minimize $\|x_F\|$, while Uncapacitated-Facility-Location allows any number of facilities to open to minimize the combined objective $|F| + \|x_F\|$.

4.1 k-Clustering. The main result in this section is a bicriteria simultaneous approximation for k-Clustering: a solution $F \subseteq X$ is *bicriteria* (α, β)-approximation for k-Clustering if its objective value is within factor α of the optimal and it opens at most βk facilities.

Theorem 4.1. *For k-Clustering,*

1. *there exists a simultaneous bicriteria $\left(1+\epsilon, O\left(\frac{\log n}{\epsilon}\right)\right)$-approximation, and*

2. *a simultaneous bicriteria $\left(3+\epsilon, O\left(\frac{\log n}{\epsilon}\right)\right)$-approximation can be found in polynomial time.*

We note that the part 1 of the theorem can also be obtained using Observation 2: take optimal k-Clustering solutions corresponding to top-l vectors for $l = \lfloor 1+\epsilon \rfloor, \lfloor (1+\epsilon)^2 \rfloor, \ldots$ and combine these facilities to obtain a single solution with $O(k \log_{1+\epsilon}(n))$ facilities; this was noted for $\epsilon = 1$ in [16]. For a polynomial-time bound, their technique of combining *fractional* solutions and rounding based on the techniques of [21] achieves a simultaneous $(6+\epsilon)$-approximation with $O(k \log_{1+\epsilon} n)$ facilities. We improve this to a $3+\epsilon$ with a simple combinatorial algorithm `IterativeClustering`, described next.

Algorithm 2 `PartialClustering`$((X, \mathrm{dist}), k, R, \alpha)$

input: A metric space (X, dist), integer $k \geq 1$, radius $R \geq 0$, parameter $\alpha \geq 1$
output: A set of k facilities $C \in \binom{X}{k}$ such that $B(C, \alpha R)$ contains at least as many points as contained by any $B(C', R)$ with $|C'| \leq k$, i.e.,

$$|B(C, \alpha R)| \geq \max_{C' \in \binom{X}{k}} |B(C', R)| .$$

Algorithm 3 `IterativeClustering`$((X, \mathrm{dist}), k, \epsilon, \alpha)$

input: A metric space (X, dist), integer $k \geq 1$, parameter $\epsilon > 0$, parameter $\alpha \geq 1$
output: A set $C \subseteq X$ of $O\left(\frac{k \log n}{\epsilon}\right)$ facilities

```
C ← ∅ R_0 = Dε/n
for l = 0, 1, ..., log_{1+ε}(n/ε) do
    R ← R_0(1 + ε)^l
    C_l ← PartialClustering((X, dist), k, R, α)
    C ← C ∪ C_l
return C
```

At a high level, our algorithm `IterativeClustering` combines several solutions with k facilities each. Each of these solutions corresponds to a radius R, and subroutine `PartialClustering` attempts to get the set of k facilities that covers the largest number of points within radius R. Radius R will increase exponentially across iterations.

For polynomial-time computations, `PartialClustering` cannot be solved exactly since it generalizes the k-center problem. To get efficient algorithms, we allow it to output k facilities that cover as many points within radius αR as those covered by any k facilities within radius R. As [24] note, [8] give an approximation algorithm for `PartialClustering` for $\alpha = 3$, which we state in a modified form:

THEOREM 4.2. (THEOREM 3.1, [8]) *Given metric* (X, dist)*, integer* $k \geq 1$*, and radius* R*, there exists a polynomial-time algorithm that outputs* k *facilities that cover at least as many points within radius* $3R$ *as those covered by any set of* k *facilities within radius* R*. That is, subroutine* `PartialClustering` *runs in polynomial-time for* $\alpha = 3$.

We give some notation: given nonempty $F \subseteq X$ and some radius $R \geq 0$, we denote by $B(F; R)$ the set of all points within distance R of F, i.e., $B(F; R) = \{x \in X : \exists\, y \in F \text{ with } \mathrm{dist}(x, y) \leq R\}$. We say that a set of facilities F *covers* p points within radius R if $|B(F; R)| \geq p$.

Let D denote the k-center optimum for (X, dist). By definition, there are k facilities that can cover all of X within radius D. Therefore, the largest radius we need to consider is D. What is the smallest radius we need to consider? Since all of our objective norms are monotonic and symmetric, points covered within very small radii do not contribute a significant amount to the norm value. Therefore, we can start at a large enough radius, which has been set to $\frac{D\epsilon}{n}$ with some foresight.

We will first prove the following claim:

CLAIM 1. `IterativeClustering` *gives a simultaneous bicriteria* $\left(\alpha(1 + 2\epsilon), O\left(\frac{\log n}{\epsilon}\right)\right)$*-approximation for symmetric monotonic norms* **Sym**.

Proof. We first show that the number of facilities output by the algorithm is $O\left(\frac{k \log n}{\epsilon}\right)$. The number of iterations in the for loop is $\log_{(1+\epsilon)}\left(\frac{n}{\epsilon}\right) = O\left(\frac{\log n}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon}\right)$. When $\epsilon > \frac{1}{n}$, this expression is $O\left(\frac{\log n}{\epsilon}\right)$. Since each iteration adds at most k facilities to C, we are done in this case. When $\epsilon \leq \frac{1}{n}$, then $\frac{k \log n}{\epsilon} \geq n$, that is, all facilities can be opened anyway.

Fix any symmetric monotonic norm $\|\cdot\|$ on $\mathbb{R}^n$, and let OPT denote the optimal solution for this norm and $x^{\mathrm{OPT}} \in \mathbb{R}^n$ denote the corresponding distance vector. Let the distance vector for facilities C output by the algorithm be x. We need to show that $\|x\| \leq \alpha(1 + 2\epsilon)\|x^{\mathrm{OPT}}\|$.

By definition, $(x^{\mathrm{OPT}})^{\uparrow}_1 \le (x^{\mathrm{OPT}})^{\uparrow}_2 \le \ldots \le (x^{\mathrm{OPT}})^{\uparrow}_n$. Let j^* be the smallest index such that $(x^{\mathrm{OPT}})^{\uparrow}_{j^*} > R_0 = \frac{D\epsilon}{n}$. Since $\|\cdot\|$ is symmetric, we have $\|x^{\uparrow}\| = \|x\|$ and $\|(x^{\mathrm{OPT}})^{\uparrow}\| = \|x^{\mathrm{OPT}}\|$. Our twofold strategy is to show that:

1. for all $j \ge j^*$,

$$(x)^{\uparrow}_j \le \alpha(1+\epsilon)(x^{\mathrm{OPT}})^{\uparrow}_j, \tag{4.12}$$

2. the contribution of $x^{\uparrow}_1, \ldots, x^{\uparrow}_{j^*-1}$ to $\|x\|$ is small; specifically,

$$\left\|\left(x^{\uparrow}_1, \ldots, x^{\uparrow}_{j^*-1}, 0, \ldots, 0\right)\right\| \le \alpha\epsilon\|x^{\mathrm{OPT}}\|. \tag{4.13}$$

Consider the first part. We have $R_0(1+\epsilon)^{\log_{1+\epsilon}(n/\epsilon)} = R_0\frac{n}{\epsilon} = D$. That is, in the final iteration of the for loop, $R = D$. Therefore, by definition of D and `PartialClustering`, C_l in this iteration covers all of X within radius αD. That is, $\|x\|_\infty \le \alpha D$ since $C_l \subseteq C$.

fix some $j \ge j^*$, and let $l \ge 0$ be the smallest integer such that $(x^{\mathrm{OPT}})^{\uparrow}_j \le R_0(1+\epsilon)^l$. If $l \ge 1 + \log_{1+\epsilon}(n/\epsilon)$, then $(x^{\mathrm{OPT}})^{\uparrow}_j > R_0(1+\epsilon)^{l-1} = D$. Since $\|x\|_\infty \le \alpha D$, inequality (4.12) holds in this case.

Otherwise, $l \le \log_{1+\epsilon}(n/\epsilon)$. The k facilities in OPT cover at least j points within radius $R = R_0(1+\epsilon)^l$. By definition of `PartialClustering`, in iteration l of the for loop, C_l covers at least j points within radius αR. Since $C_l \subseteq C$, C also covers at least j points within radius αR, so that $x^{\uparrow}_j \le \alpha R = R_0(1+\epsilon)^l$. By definition of l, $(x^{\mathrm{OPT}})^{\uparrow} > R_0(1+\epsilon)^{l-1}$, and so

$$x^{\uparrow}_j \le \alpha R_0(1+\epsilon)^l \le \alpha(1+\epsilon)(x^{\mathrm{OPT}})^{\uparrow}_j.$$

We move to (4.13). By definition of j^*, OPT covers at least j^*-1 points within radius R_0. In iteration 0, by definition of `PartialClustering`, C_0 (and therefore C) covers at least (j^*-1) points within radius αR_0. That is, $x^{\uparrow}_{j^*-1} \le \alpha R_0$.

Denote $(1, 0, \ldots, 0) = \mathbf{e}$. Since $\|\cdot\|$ is monotonic and D is the k center optimum, $\|x^{\mathrm{OPT}}\| \ge (\|x^{\mathrm{OPT}}\|_\infty, 0, \ldots, 0)\,\|\mathbf{e}\| \ge D\|\mathbf{e}\|$. Therefore,

$$\begin{aligned}
\left\|\left(x^{\uparrow}_1, \ldots, x^{\uparrow}_{j^*-1}, 0, \ldots, 0\right)\right\| &\le \sum_{j\in[j^*-1]} x^{\uparrow}_j\|\mathbf{e}\| && \text{(triangle inequality)}\\
&\le \sum_{j\in[j^*-1]} \alpha R_0\|\mathbf{e}\| && (x^{\uparrow}_{j^*-1} \le \alpha R)\\
&< n\alpha\frac{D\epsilon}{n}\|\mathbf{e}\| && (j^* \le n)\\
&\le \alpha\epsilon\|x^{\mathrm{OPT}}\|. && (\|x^{\mathrm{OPT}}\| \ge D\|\mathbf{e}\|)
\end{aligned}$$

Together, inequalities (4.12), (4.13) imply that

$$\begin{aligned}
\|x\| &\le \left\|\left(x^{\uparrow}_1, \ldots, x^{\uparrow}_{j^*-1}, 0, \ldots, 0\right)\right\| + \left\|\left(0, \ldots, 0, x^{\uparrow}_{j^*}, \ldots, x^{\uparrow}_n\right)\right\| && \text{(triangle inequality)}\\
&\le \alpha\epsilon\|x^{\mathrm{OPT}}\| + \alpha(1+\epsilon)\left\|\left(0, \ldots, 0, (x^{\mathrm{OPT}})^{\uparrow}_{j^*}, \ldots, (x^{\mathrm{OPT}})^{\uparrow}_n\right)\right\| && \text{(inequalities (4.12), (4.13))}\\
&\le \alpha\epsilon\|x^{\mathrm{OPT}}\| + \alpha(1+\epsilon)\|x^{\mathrm{OPT}}\| = \alpha(1+2\epsilon)\|x^{\mathrm{OPT}}\|. && (\|\cdot\| \text{ is symmetric monotonic})
\end{aligned}$$

□

With this result in hand, our main theorem is simple to derive: we choose $\alpha = 1$ in the claim with $\epsilon/2$ as the parameter for the existence result. We choose $\alpha = 3$ in the claim with $\epsilon/6$ as the parameter for the polynomial-time result; Theorem 4.2 guarantees that the algorithm is polynomial-time.

4.2 Uncapacitated-Facility-Location. First, we note that a single solution cannot be better than $\Omega(\sqrt{n})$-approximate for even the L_1 and L_∞ norms: suppose the metric is a star metric with n leaves. The distance from the center to each leaf is $\sqrt{n}$. Then the optimal L_1 solution is to open each facility, and the cost of this solution is $n+1$. The optimal L_∞ solution is to open just one facility at the center, the cost of this solution is $1+\sqrt{n}$. Now, any solution that opens fewer than $n/2$ facilities has cost $\geq n/2 + (n/2)\sqrt{n} = \Omega(n\sqrt{n})$ for the L_1 norm and therefore is an $\Omega(\sqrt{n})$-approximation. Any solution that opens $\geq n/2$ facilities is an $\Omega(\sqrt{n})$-approximation for the L_∞ norm. A similar example was noted for the k-clustering variant in [16].

This motivates us to seek larger portfolios and get a smaller approximation. The main theorem of this section gives an $O(\log n)$-approximate portfolio of size $O(\log n)$ for UNCAPACITATED-FACILITY-LOCATION:

THEOREM 4.3. *An $O(\log n)$-approximate portfolio of size $O(\log n)$ for symmetric monotonic norms* **Sym** *over* UNCAPACITATED-FACILITY-LOCATION *can be found in polynomial time.*

Proof. Assume without loss of generality that the number of points n is a power of 2. Choose solutions corresponding to $k = 2^0, 2^1, 2^2, \ldots, 2^{\log_2 n}$ with $\epsilon = 1$ in Theorem 4.1 part 2. There are clearly $O(\log n)$ of these, and the theorem asserts that they can be found in polynomial time. We claim that these form an $O(\log n)$-approximate all-norm portfolio.

Fix a norm $\|\cdot\|$, and suppose the optimal solution OPT for this norm opens $k^* \in [n]$ facilities. Let l be the unique integer such that $2^{l-1} < k^* \leq 2^l$, i.e., $l = \lceil \log_2 k^* \rceil$. We show that the solution corresponding to $k = 2^l$ in our portfolio is an $O(\log n)$-approximation for $\|\cdot\|$. Add arbitrary $2^l - k^*$ facilities to OPT; this only decreases the induced distance vector x^{OPT}. For this new set of facilities, we have the guarantee from Theorem 4.1 that $\|x\| \leq 4\|x^{\mathrm{OPT}}\|$. Therefore, the objective value of the portfolio solution is

$$O(\log n) \cdot 2^l + \|x\| = O(\log n)\left(k^* + \|x^{\mathrm{OPT}}\|\right) = O(\log n) \cdot \mathrm{OPT}.$$

This completes the proof. ∎

5 Discussion and Open Problems

In this work, we gave the first characterization of trade-off between portfolio size and the approximation factors for certain scheduling problems. However, questions about the design of portfolios can be asked for any setting in optimization with a class of objectives: at their core, portfolios simply ask if the set of feasible solutions can be represented by a smaller subset and still enjoy some guarantees for optimization for a given class of functions. We state some open questions here:

1. **General covering polyhedra:** For covering polyhedra in dimension d, we improved portfolio sizes from the general bound of poly(d) when the number of constraints $r = o(\sqrt{\log d}/(\log\log d))$. We conjecture that this is tight up to polylogarithmic factors, i.e, that there exist covering polyhedra in dimension d with $O(\log d)$ constraints such that any $O(\log d)$-approximate portfolios for symmetric monotonic norms must have polynomial size.

   Conversely, this raises the question of whether there are good characterizations of polyhedra that admit small-sized portfolios for ordered norms.

2. **Scheduling with unidentical jobs:** We show $O(1)$-approximate portfolios of size $O(\log d)$ for MLIJ, i.e., machine load minimization on d machines with identical jobs. It is open if there exists a similar-sized portfolio for the more general problem of machine-load minimization with *unidentical* jobs. We believe that this may not be true.

3. **Approximation gap between ordered and symmetric monotonic norms**: Our bounds using `OrderAndCount` have a factor $O(\log d)$ approximation gap between ordered norms and symmetric monotonic norms. `OrderAndCount` does not tackle symmetric monotonic norms directly but instead uses the $O(\log d)$-approximation by some ordered norm. Improving upon the $O(\log d)$ approximation factor for symmetric monotonic norms would be interesting.

4. **Class of equity objectives:** Our work focused on understanding portfolios for various families of symmetric monotonic norms. However, many more notions of equity have been proposed in the literature, such as lexicographically optimal solutions [24], for which such questions are largely open.

References

[1] Yossi Azar and Shai Taub. All-Norm Approximation for Scheduling on Identical Machines. In *Algorithm Theory - SWAT*, pages 298–310, 2004.

[2] Jane Battersby and Jonathan Crush. Africa's Urban Food Deserts. *Urban Forum*, 25(2):143–151, June 2014.

[3] Annette Bernhardt, Reem Suleiman, and Lisa Kresge. Data and algorithms at work: the case for worker technology rights. 2021.

[4] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Symposium on Theory of Computing (STOC)*, pages 163–171, 1994.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[6] Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Symposium on Theory of Computing (STOC) 2019*, pages 126–137, June 2019.

[7] Ashok K. Chandra and C. K. Wong. Worst-Case Analysis of a Placement Algorithm Related to Storage Allocation. *SIAM Journal on Computing*, 4(3):249–263, September 1975.

[8] Moses Charikar, Samir Khuller, David M. Mount, and Gin Narasimhan. Algorithms for facility location problems with outliers. *Symposium on Discrete Algorithms (SODA)*, pages 642–651, 2001.

[9] Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. Approximating Fair Clustering with Cascaded Norm Objectives. *Symposium on Discrete Algorithms (SODA)*, pages 2664–2683, January 2022.

[10] Alexandra Chouldechova. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. 5(2):153–163, June 2017.

[11] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Wormian Vaughan. Group fairness for the allocation of indivisible goods. In *Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference*, pages 1853–1860, January 2019.

[12] Steven Cummins and Sally Macintyre. "Food deserts"—evidence and assumption in health policy making. *BMJ : British Medical Journal*, 325(7361):436–438, August 2002.

[13] Majid Farhadi, Alejandro Toriello, and Prasad Tetali. The Traveling Firefighter Problem. In *Applied and Computational Discrete Algorithms (ACDA)*, pages 205–216. 2021.

[14] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 259–268, August 2015.

[15] Meredith Gartin. Food deserts and nutritional risk in Paraguay. *American Journal of Human Biology*, 24(3):296–301, 2012.

[16] Ashish Goel and Adam Meyerson. Simultaneous Optimization via Approximate Majorization for Concave Profits or Convex Costs. *Algorithmica*, 44(4):301–323, April 2006.

[17] Daniel Golovin, Anupam Gupta, Amit Kumar, and Kanat Tangwongsan. All-Norms and All-Lp-Norms Approximation Algorithms. *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2008.

[18] Swati Gupta, Jai Moondra, and Mohit Singh. Which Lp norm is the fairest? Approximations for fair facility location across all "p". In *Economics and Computation (EC) 2023*, page 817, July 2023.

[19] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[20] G. H. Hardy, J. E. Littlewood, and George Pólya. *Inequalities.* Cambridge University Press, Cambridge, 1952.

[21] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, March 2001.

[22] Jon Kleinberg. Inherent Trade-Offs in Algorithmic Fairness. In *2018 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, page 40, June 2018.

[23] Martin Knapp, Vasiliki Koutsogeorgopoulou, and Justin Davis Smith. *Who volunteers and why?: The key factors which determine volunteering.* Volunteer Centre UK, UK, 1995.

[24] Amit Kumar and Jon Kleinberg. Fairness measures for resource allocation. In *Symposium on Foundations of Computer Science (FOCS)*, pages 75–85, November 2000. ISSN: 0272-5428.

[25] Michael Locke, Angela Ellis, and Justin Davis Smith. Hold on to what you've got: The volunteer retention literature. *Voluntary Action*, 5(3):81–99, 2003.

[26] Wenjun Lyu, Kexin Zhang, Baoshen Guo, Zhiqing Hong, Guang Yang, Guang Wang, Yu Yang, Yunhuai Liu, and Desheng Zhang. Towards fair workload assessment via homogeneous order grouping in last-mile delivery. In *International Conference on Information & Knowledge Management*, pages 3361–3370, 2022.

[27] Vahideh Manshadi and Scott Rodilitz. Online Policies for Efficient Volunteer Crowdsourcing. In *Economics and Computation (EC) 2020*, pages 315–316, July 2020.

[28] Kerryann O'Meara, Dawn Culpepper, Joya Misra, and Audrey Jaeger. Equity-minded faculty workloads: What we can and should do now. *American Council on Education*, 2021.

[29] Kalen Patton, Matteo Russo, and Sahil Singla. Submodular Norms with Applications To Online Facility Location and Stochastic Probing. In *APPROX*, volume 275, pages 23:1–23:22, 2023.

[30] Zahra Rooddehghan, Zohreh ParsaYekta, and Alireza Nikbakht Nasrabadi. Nurses, the oppressed oppressors: A qualitative study. *Global journal of health science*, 7(5):239, 2015.

[31] David B. Shmoys and Eva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(1):461–474, February 1993.

[32] USDA. USDA ERS - Go to the Atlas, 2023.

A Omitted Proofs from Section 1.3

Proof. [Proof of Lemma 1.5]

1. For any $f \in \mathbf{C}$, $\min_{x \in X_2} f(x) \le \alpha_2 \min_{x \in X_1} f(x) \le \alpha_2 \alpha_1 \min_{x \in \mathcal{D}} f(x)$. The first inequality follows since X_2 is an α_2-approximate portfolio for $\mathbf{C}$ over X_1 and the second inequality follows since X_1 is an α_1-approximate portfolio for $\mathbf{C}$ over $\mathcal{D}$.

2. For each $f \in \mathbf{C}$,
$$\min_{x \in \mathcal{D}} f(x) = \min_{i \in [n]} \min_{x \in \mathcal{D}_i} f(x) \le \min_{i \in [n]} \alpha \min_{x \in X_i} f(x) = \alpha \min_{x \in \cup_{i \in [n]} X_i} f(x).$$
Therefore, $\cup_{i \in [n]} X_i$ is an α-approximate portfolio for $\mathbf{C}$ over $\mathcal{D}$.

□

Proof. [Proof of Lemma 1.1] Let $K = \{x \in \mathbb{R}^d : \|x\|_{(w)} \le 1\}$, and let $K^* = \{y \in \mathbb{R}^d : y^\top x \le 1 \,\forall\, x \in K\}$. Also denote $\overline{K} = \left\{ y \in \mathbb{R}^d : \max_{k \in [d]} \frac{(\sigma |y|^{\downarrow})_k}{(\sigma w)_k} \le 1 \right\}$. We will show that $\overline{K} = K^*$.

Suppose $y \in \overline{K}$. Then for any $x \in K$,

$$\begin{aligned}
y^\top x &\le (|y|^{\downarrow})^\top |x|^{\downarrow} && \text{(rearrangment inequality)}\\
&= (\sigma |y|^{\downarrow})^\top (\Delta |x|^{\downarrow}) && \text{(alternating sum)}\\
&\le (\sigma w)^\top (\Delta |x|^{\downarrow}) && (y \in \overline{K})\\
&= \|x\|_{(w)} && \text{(alternating sum)}\\
&\le 1. && (x \in K)
\end{aligned}$$

That is, $y \in K^*$. Conversely, assume $y \in K^*$ so that $y^\top x \le 1$ for each $x \in K$. Since K^* is symmetric, assume without loss of generality that $y_1 \ge \ldots \ge y_d \ge 0$, other cases are handled similarly. It is easy to check that for each $k \in [d]$, $x(k) := \frac{1}{(\sigma w)_k}(\underbrace{1, \ldots, 1}_{k}, 0, \ldots, 0)$ is in K. Therefore $1 \ge y^\top x(k) = \frac{(\sigma y)_k}{(\sigma w)_k} = \frac{(\sigma |y|^{\downarrow})_k}{(\sigma w)_k}$, implying that $y \in \overline{K}$. □

Proof. [Proof of Lemma 1.2] This proof is similar to the previous proof. For any $x, y \in \mathbb{R}^d$, we have

$$\begin{aligned}
y^\top x &\le (|y|^{\downarrow})^\top |x|^{\downarrow} && \text{(rearrangment inequality)}\\
&= \sum_{k \in [d]} (\sigma |y|^{\downarrow})_k (\Delta |x|^{\downarrow})_k && \text{(alternating sum)}\\
&\le \|y\|^*_{(w)} \sum_{k \in [d]} (\sigma w)_k (\Delta |x|^{\downarrow})_k && \text{(definition of } \|y\|^*_{(w)})\\
&= \|y\|^*_{(w)} \|x\|_{(w)} && \text{(alternating sum).}
\end{aligned}$$

Further, the first inequality holds if and only if x, y are order-consistent, i.e., if and only if there exists an order π such that $x^{\downarrow} = x_\pi$ and $y^{\downarrow} = y_\pi$. The second inequality holds if and only if for each k, $(\sigma |y|^{\downarrow})_k (\Delta |x|^{\downarrow})_k = \|y\|^*_{(w)} (\sigma w)_k (\Delta |x|^{\downarrow})_k$, which happens if and only if $\Delta |x|^{\downarrow} = 0$ or $\frac{(\sigma |y|^{\downarrow})_k}{(\sigma w)} = \|y\|^*_{(w)}$. □

B Upper and Lower Bounds on Portfolios for Arbitrary $\mathcal{D}$

Given $\alpha \ge 1$, we give worst-case upper and lower bounds for ordered norms **Ord** and symmetric monotonic norms **Sym**.

A general upper bound of $\mathrm{poly}(d^{1/\epsilon})$ for **Ord** was established in [6] by a sparsification argument, by grouping together similar ordered norms. We first show that a similar sparsification argument in the set $\mathcal{D}$ of feasible vectors yields the same bound for **Sym** on the size of $(1+\epsilon)$-approximate portfolios, i.e., there always exist $(1+\epsilon)$-approximate portfolios of size $\mathrm{poly}(d^{1/\epsilon})$ for symmetric monotonic norms (Theorem B.1). The subsequent subsection gives nearly-polynomial lower bounds, showing that these upper bounds are not far off from the true bounds.

B.1 Upper Bound. To get the upper bound, we use a sparsification technique that places all vectors in one of polynomially-many 'buckets', with the property that any two vectors in the bucket approximately majorize each other. This implies a portfolio of polynomial size using Lemma 1.3. This technique was used by [6] to (approximately) enumerate **Ord**, and their result essentially implies a polynomial-sized portfolio for **Ord**. Our observation is that we can get a portfolio for **Sym** if we apply this argument to vectors in $\mathcal{D}$ instead:

THEOREM B.1. *Given a set of vectors $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ and $\epsilon \in (0,1]$, there is always a $(1+\epsilon)$-approximate portfolio for* **Sym** *of size at most $d^{O(1/\epsilon)}$.*

Proof. Let $v^* = \min_{x\in\mathcal{D}} \|x\|_\infty$, with the corresponding vector denoted x^*. Let $\overline{\mathcal{D}} = \{x \in \mathcal{D} : \|x\|_\infty \leq dv^*\}$. We first claim that $\overline{\mathcal{D}}$ is an optimal portfolio for all symmetric monotonic norms over $\mathcal{D}$, i.e., for each symmetric monotonic norm $\|\cdot\|$, the corresponding minimum norm point $\arg\min_{x\in\mathcal{D}}\|x\| \in \overline{\mathcal{D}}$. To see this, let $\overline{x} = \arg\min_{x\in\mathcal{D}}\|x\|$. Then,

$$
\begin{aligned}
&\|\overline{x}\|_\infty\|(1,0,\ldots,0)\| \\
\leq&\|\overline{x}\| && (\|\cdot\| \text{ is symmetric}) \\
\leq&\|x^*\| && (\text{optimality of } \overline{x}) \\
\leq&\|x^*\|_\infty\|(1,\ldots,1)\| \\
\leq&v^*d\|(1,0,\ldots,0)\|.
\end{aligned}
$$

This implies that $\|\overline{x}\|_\infty \leq dv^*$, or that $\overline{x} \in \overline{\mathcal{D}}$. Next, we will place all vectors in $\overline{\mathcal{D}}$ in one of $d^{O(1/\epsilon)}$ buckets such that for any two vector x, y in the same bucket, $x \preceq (1+\epsilon)y$ and $y \preceq (1+\epsilon)x$, so that by Lemma 1.3, $\|x\| \simeq_{1+\epsilon} \|y\|$ for all symmetric monotonic norms $\|\cdot\|$. Consequently, it is sufficient to pick just one vector in each bucket to get a $(1+\epsilon)$-approximate portfolio for all symmetric monotonic norms over $\mathcal{D}$.

Denote $T = \lceil\log_{1+\frac{\epsilon}{3}} d\rceil$. Each bucket $B(a_1,\ldots,a_T)$ is specified by an increasing sequence $a_1 \leq a_2 \leq \ldots \leq a_T$ of integers that lie in $[0, 2T]$. The number of such sequences is $\binom{3T}{T} \leq 3^T = d^{O(1/\epsilon)}$, bounding the number of buckets. Let $c_i = \lfloor(1+\epsilon/3)^i\rfloor$ for $i \in [T]$. Then x lies in bucket $B(a_1,\ldots,a_T)$ where $a_i = \left\lfloor\log_{1+\frac{\epsilon}{3}}\left(\frac{1}{v^*}\|x\|_{\mathbf{1}_{c_i}}\right)\right\rfloor$.

First, we show that this assignment is valid, i.e., each $a_i \in [0, 2T]$. Indeed,

$$\frac{1}{v^*}\|x\|_{\mathbf{1}_{c_i}} \leq \frac{1}{v^*}c_i\|x\|_\infty \leq \frac{d\|x_\infty\|}{v^*} \leq d^2.$$

The final inequality follows since $x \in \overline{\mathcal{D}}$. Therefore, $a_i \leq \log_{1+\frac{\epsilon}{3}d^2} \leq 2T$. Next, we claim that for any $x, y \in B(a_1,\ldots,a_d)$, $x \preceq (1+\epsilon)y$. Fix any $k \in [d]$, and let $i \in [0,T]$ such that $c_i \leq k < c_{i+1}$. Note that by definition of a_i, we have $a_i \leq \log_{1+\frac{\epsilon}{3}}\left(\frac{1}{v^*}\|x\|_{\mathbf{1}_{c_i}}\right) \leq a_i + 1$, and the same inequality also holds for y. Then,

$$
\begin{aligned}
\|x\|_{\mathbf{1}_k} \leq \frac{k}{c_i}\|x\|_{\mathbf{1}_{c_i}} &\leq \frac{k}{c_i}\left(v_i\,(1+\epsilon/3)^{a_i+1}\right) \\
&= \frac{k(1+\epsilon/3)}{c_i}\left(v_i\,(1+\epsilon/3)^{a_i+1}\right) \\
&\leq \frac{k(1+\epsilon/3)}{c_i}\|y\|_{\mathbf{1}_{c_i}} \leq \frac{k(1+\epsilon/3)}{c_i}\|y\|_{\mathbf{1}_k}.
\end{aligned}
$$

Finally, $\frac{k}{c_i} \leq \frac{c_{i+1}-1}{c_i} \leq (1+\epsilon/3)$, so that $\frac{\|x\|_{\mathbf{1}_k}}{\|y\|_{\mathbf{1}_k}} \leq (1+\epsilon/3)^2 = 1 + \frac{2}{3}\epsilon + \frac{1}{9}\epsilon^2 \leq 1+\epsilon$ for all $\epsilon \in (0,1]$. ∎

B.2 Lower Bound. We now give lower bounds on worst-case portfolio sizes for **Sym**, **Ord** in dimension d. Theorem B.2 shows that there exist polytopes $\mathcal{D}$ in $\mathbb{R}^d_{\geq 0}$ such that any $O(\log d)$-approximate portfolios for **Ord** must have size $d^{\Omega(1/\log\log d)}$. Since **Ord** $\subseteq$ **Sym**, this bound is also true for **Sym** and recorded in Corollary B.1. Before we prove the theorems, we need a counting lemma:

LEMMA B.1. *Given $L \geq 1$, Let T be the set of integral sequences $a = (a_0,\ldots,a_L)$ such that $a_{i-1} \leq a_i \leq a_{i-1}+1$ for all $i \in [L]$ and $a_0 = 0$. Then there exists a subset $\overline{T} \subseteq T$ such that (1) $|\overline{T}| \geq 2^L/(2L^2)$, and (2) for any two sequences $a, a' \in \overline{T}$, there exists an i such that $a'_i < a_i$, and vice-versa.*

Proof. We first show that $|T| = 2^L$. For any such sequence a, consider $\phi(a) = (a_1 - a_0, \ldots, a_L - a_{L-1})$. Then $\phi(a)$ maps sequences in T to binary sequences $(b_1, \ldots, b_L)$; further, ϕ is bijective. Therefore, $|T|$ is the number of binary sequences $(b_1, \ldots, b_L)$, which is 2^L.

Also note that $\geq$ is a partial order on T: $a \leq a'$ if and only if $a'_i \geq a_i$ for all $i \in [0, L]$. For any distinct a, a' such that $a' \geq a$, we must have that $\sum_{i\in[L]} a'_i \geq 1 + \sum_{i\in[L]} a_i$. Further, $\sum_{i\in[L]} a_i \leq L^2$ for all $a \in T$. Therefore, the length of any chain in order $\geq$ on T is at most $L^2 + 1$. This means that any chain decomposition of $\geq$ on T must have at least $|T|/(L^2 + 1) \geq 2^L/(2L^2)$ chains. By Dilworth's theorem, this is also the size of the largest antichain. But an anti-chain is exactly the set $\overline{T}$ we are looking for. ∎

Theorem B.2. *There exist set of vectors $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ such that any $O(\log d)$-approximate portfolio for* **Ord** *must have size $d^{\Omega(1/\log\log d)}$. Further, this bound is also true for a polytope $\mathcal{D} \subseteq \mathbb{R}^d$. That is, there exist polytopes $\mathcal{D}$ such that any $O(\log d)$-approximate portfolio for* **Ord** *over $\mathcal{D}$ must have size $d^{\Omega(1/\log\log d)}$.*

Corollary B.1. *There exist $\mathcal{D} \subseteq \mathbb{R}^d_{\geq 0}$ such that any $O(\log d)$-approximate portfolio for* **Sym** *must have size $d^{\Omega(1/\log\log d)}$. Further, this bound is also true for a polytope $\mathcal{D}$.*

Proof. [Proof of theorem] Let $S = \log^3 d$, and let L be such that $S^0 + S^1 + \ldots + S^L = d$. Then $L = \Theta\left(\frac{\log d}{\log S}\right) = \Theta\left(\frac{\log d}{\log\log d}\right)$, or that $S/L = \Omega(\log^2 d)$.

Let $\overline{T}$ be the set of integral sequences from the previous lemma, i.e., each sequence $a = (a_0, \ldots, a_L)$ is such that $a_{i-1} \leq a_i \leq a_{i-1} + 1$ for all $i \in [L]$ and $a_0 = 0$, and for any two sequences $a, a' \in \overline{T}$, there exists i such that $a'_i < a_i$. Define

$$x(a) = \Big(\underbrace{S^{-a_0}}_{S^0}, \underbrace{S^{-a_1}, \ldots, S^{-a_1}}_{S^1}, \ldots, \underbrace{S^{-a_L}, \ldots, S^{-a_L}}_{S^L}\Big).$$

Note that since $a_i \geq a_{i-1}$, $x^{\downarrow} = x$. Further, since $a_i \leq a_{i-1} + 1$, we have $a_i - i \leq a_{i-1} - (i-1)$. Define

$$w(a) = \Big(\underbrace{S^{a_0 - 0}}_{S^0}, \underbrace{S^{a_1 - 1}, \ldots, S^{a_1 - 1}}_{S^1}, \ldots, \underbrace{S^{a_L - L}, \ldots, S^{a_L - L}}_{S^L}\Big).$$

Then

$$\|x(a)\|_{(w(a))} = x(a)^\top w(a) = \sum_{i\in[0,L]} S^{-a_i} S^{a_i - i} S^i = L.$$

Further, for any other $a' = (a'_0, \ldots, a'_L) \in \overline{T}$, there exists i such that $a'_i < a_i$, we get

$$\|x(a')\|_{(w(a))} \geq S^{-a'_i} S^{a_i - i} S^i > S.$$

Since $S/L = \Omega(\log^2 d)$, this means that $x(a')$ is an $\omega(\log d)$-approximation for $\|\cdot\|_{(w(a))}$. That is, any $O(\log d)$-approximate portfolio for **Ord** over $\overline{T}$ must have size $|\overline{T}| \geq 2^L/(2L^2)$. However,

$$\frac{2^L}{2L^2} = 2^{\Theta((\log d)/\log\log d)}\Theta\left(\frac{(\log\log d)^2}{(\log d)^2}\right) = d^{\Theta\left(\frac{1}{\log\log d}\right) - O\left(\frac{\log\log d}{\log d}\right)} = d^{\Omega\left(\frac{1}{\log\log d}\right)}.$$

To prove the second part of the theorem, we claim that in fact even for $\mathrm{conv}(\overline{T})$, we have that any $O(\log d)$-approximate portfolio must have size $|\overline{T}| = d^{\Omega(1/\log\log d)}$. Let $x = \sum_{b\in\overline{T}} \lambda_b x(b) \in \mathrm{conv}(\overline{T})$. Fix $a \in \overline{T}$. We will show that for all x such that $1 - \lambda_a > 1/4$, $\|x\|_{w(a)} = \Theta(S/L)\|x(a)\|_{w(a)}$. That is, the any $O(\log d)$-approximate minimizer x of $\|\cdot\|_{w(a)}$ in $\mathrm{conv}(\overline{T})$ must have $\lambda_a \geq \frac{3}{4}$, implying the claim.

First, note that for each b, $x(b)^{\downarrow} = x(b)$. Therefore,

$$\begin{aligned}\|x\|_{w(a)} &= \left(\sum_{b\in\overline{T}} \lambda_b x(b)\right)^{\top} w(a) = \sum_{b\in\overline{T}} \lambda_b \|x(b)\|_{w(a)} \\ &= \lambda_a \|x(a)\|_{w(a)} + \sum_{b\neq a} \lambda_b \|x(b)\|_{w(a)} \\ &\geq \lambda_a L + S\sum_{b\neq a}\lambda_b \geq S(1-\lambda_a) \geq S/4.\end{aligned}$$

Where the last inequality follows by the assumption that $1-\lambda_a \geq 1/4$. Therefore, $\|x\|_{w(a)} = \Theta(S/L) = \omega(\log d)$. This finishes the proof. □

C Proof of Theorem 2.3

We show that all instances of MLIJ (d) admit $O(1)$-approximate portfolio of size 2 for all top-k norms. Recall that theorem 2.2 gives instances where any $O(1)$-approximate portfolio for ordered norms must have size $\widetilde{\Omega}(\log d)$. Combined, this implies the result with $m = d$. We will denote the top-k norm by $\|\cdot\|_{\mathbf{1}_k}$

Recall Lemmas 2.5, 2.4: $X' = \{\hat{x}(1), \ldots, \hat{x}(L)\}$ is an $O(1)$-approximate portfolio for all ordered norms where $\frac{1}{2}x(l) \leq \hat{x}(l) \leq 2x(l)$ for all $l \in [L]$. Therefore, $\|\hat{x}(l)\|_{\mathbf{1}_k}$ is within factor 2 of $\|x(l)\|_{\mathbf{1}_k}$ for all $k \in [d]$. Further for all $k \in [d]$,

$$\|x(l)\|_{\mathbf{1}_k} = \begin{cases} \frac{ln}{\sum_{i\in[l]}\frac{1}{p_i}} & \text{if } l \leq k, \\ \frac{kn}{\sum_{i\in[l]}\frac{1}{p_i}} & \text{if } l > k. \end{cases}$$

Fix k. Since $p_i \leq p_{i+1}$ for all i, $\frac{ln}{\sum_{i\in[l]}\frac{1}{p_i}}$ is non-increasing in l. Further, $\frac{kn}{\sum_{i\in[l]}\frac{1}{p_i}}$ is decreasing in l. Therefore, the smallest among $\|x(l)\|_{\mathbf{1}_k}, l \in [L]$ is either $\|x(1)\|_{\mathbf{1}_k}$ or $\|x(L)\|_{\mathbf{1}_k}$. Therefore,

$$\begin{aligned}\min\{\|\hat{x}(1)\|_{\mathbf{1}_k}, \|\hat{x}(l)\|_{\mathbf{1}_k}\} &\leq 2\min\{\|x(1)\|_{\mathbf{1}_k}, \|x(l)\|_{\mathbf{1}_k}\} \\ &\leq 2\min\{\|x(1)\|_{\mathbf{1}_k}, \|x(2)\|_{\mathbf{1}_k}, \ldots, \|x(L)\|_{\mathbf{1}_k}\} \\ &\leq 4\min\{\|\hat{x}(1)\|_{\mathbf{1}_k}, \|\hat{x}(2)\|_{\mathbf{1}_k}, \ldots, \|\hat{x}(L)\|_{\mathbf{1}_k}\}.\end{aligned}$$

Since $\{\hat{x}(1), \ldots, \hat{x}(L)\}$ is an $O(1)$-approximate portfolio for all ordered norms, this implies that $\{\hat{x}(1), \hat{x}(L)\}$ is an $O(1)$-approximate portfolio for all top-k norms.

We can also show that portfolios for ordered norms are not portfolios for L_p norms: consider an instance of identical jobs scheduling with $p_i = \sqrt{i}$ for each $i \in [d]$. Denote $\rho(l) = \sum_{i\in[l]}\frac{1}{p_i} = \sum_{i\in[l]}\frac{1}{\sqrt{i}}$; also denote the dth Harmonic number $H_d = \sum_{i\in[d]}\frac{1}{i} = \Theta(\log d)$. Then for each $l \in [d]$, $x(l) = \Big(\underbrace{\frac{n}{\rho(l)}, \ldots, \frac{n}{\rho(l)}}_{l}, 0, \ldots, 0\Big)$.

Recall (Lemmas 2.4, 2.5) that there exists an $L \in [d]$ such that (1) $\frac{1}{2}x(l) \leq \hat{x}(l) \leq 2x(l)$ for all $l \in [L]$ and (2) $X' = \{\hat{x}(1), \ldots, \hat{x}(L)\}$ is an $O(1)$-approximate portfolio for ordered norms for some $L \in [d]$. We claim that each $x \in Xl$ is an $\Omega(\sqrt{H_d})$-approximation for the L_2 norm.

For each $l \in [L]$, $\rho(l) \leq 1 + 2\int_1^l \frac{2}{\sqrt{x}} \leq 4\sqrt{l}$, so that

$$2\|\hat{x}(l)\|_2 \geq \|x(l)\|_2 = \frac{n}{\rho(l)} \cdot \sqrt{l} \geq \frac{n}{4}.$$

Consider the following assignment: assign $n_i = \frac{n}{iH_d}$ jobs to machine $i \in [d]$, and choose n large enough so that each n_i is integral. Then this is a valid assignment since $\sum_{i\in[d]} n_i = n$ by definition of H_d. The machine loads for this assignment are $x_i = n_i p_i = \frac{n}{H_d\sqrt{i}}$. The L_2 norm of x is

$$\|x\|_2 = \frac{n}{H_d}\sqrt{\sum_{i\in[d]}\frac{1}{i}} = \frac{n}{\sqrt{H_d}}.$$

Therefore, each $\hat{x}(l)$ is an $\Omega(\sqrt{H_d})$-approximation for the L_2 norm.

D Omitted Proofs from Section 3

We prove Lemmas 3.2 and 3.6 in this section.

D.1 Proof of Lemma 3.2. For each $i \in [r], j \in [d]$, by construction we have $\widetilde{A}_{i,j} \le A_{i,j}$, so that if $x \in \mathcal{P}$, then $Ax \ge \widetilde{A}x \ge \mathbf{1}_r$, i.e., $\widetilde{\mathcal{P}} \subseteq \mathcal{P}$.

Suppose $x \in \mathcal{P}$. We will claim that there is some $\widetilde{x} \in \widetilde{\mathcal{P}}$ such that $\widetilde{x} \preceq (1+\epsilon)x$. From Lemma 1.3, this will imply that $\|\widetilde{x}\| \le (1+\epsilon)\|x\|$, and therefore that $\min_{\widetilde{x}\in\widetilde{\mathcal{P}}} \|\widetilde{x}\| \le (1+\epsilon)\min_{x\in\mathcal{P}} \|x\|$. This implies the lemma.

To see the claim, suppose $x \in \mathcal{P}$. Define $\widetilde{x} = \left(1+\frac{\epsilon}{2}\right)\left(x + \frac{\epsilon(\sigma x)_d}{3d}(1,\ldots,1)\right)$. First, we show that $\widetilde{x} \preceq (1+\epsilon)x$. We have that for all $k \in [d]$,

$$(\sigma\widetilde{x}^{\downarrow})_k = \left(1+\frac{\epsilon}{2}\right)\left((\sigma x^{\downarrow})_k + \frac{k\epsilon(\sigma x)_d}{3d}\right).$$

However, $\frac{(\sigma x)_d}{d} \le \frac{(\sigma x^{\downarrow})_k}{k}$, so that the above gives us

$$(\sigma\widetilde{x}^{\downarrow})_k = \left(1+\frac{\epsilon}{2}\right)\left((\sigma x^{\downarrow})_k + \frac{k\epsilon(\sigma x)_d}{3d}\right) \le \left(1+\frac{\epsilon}{2}\right)\left((\sigma x^{\downarrow})_k + \frac{\epsilon(\sigma x^{\downarrow})_k}{3}\right) = \left(1+\frac{\epsilon}{2}\right)\left(1+\frac{\epsilon}{3}\right)(\sigma x^{\downarrow})_k.$$

For all $\epsilon \in (0,1]$, $\left(1+\frac{\epsilon}{2}\right)\left(1+\frac{\epsilon}{3}\right) \le 1+\epsilon$, so that $\widetilde{x} \preceq (1+\epsilon)x$. Next, we show that $\widetilde{x} \in \widetilde{\mathcal{P}}$. Clearly, $\widetilde{x} \ge x \ge 0$; it remains to show that $\widetilde{A}\widetilde{x} \ge \mathbf{1}_r$.

Let $B(i)$ denote the set of columns in row i such that $\widetilde{A}_{i,j} = 0$. Fix $i \in [r]$; denote the ith rows of $A, \widetilde{A}$ respectively by $A_i, \widetilde{A}_i$. From the algorithm, for $j \notin B(i)$, we have $\widetilde{A}_{i,j} \ge \frac{1}{1+\frac{\epsilon}{2}} A_{i,j}$. Therefore,

$$\begin{aligned}
\widetilde{A}_i^\top \widetilde{x} = \sum_{j\in[d]} \widetilde{A}_{i,j}\widetilde{x}_j &= \sum_{j\notin B(i)} \widetilde{A}_{i,j}\widetilde{x}_j && (\widetilde{A}_{i,j} = 0\ \forall\, j \in B(i)),\\
&\ge \frac{1}{1+\frac{\epsilon}{2}} \sum_{j\notin B(i)} A_{i,j}\widetilde{x}_j\\
&= \frac{1}{1+\frac{\epsilon}{2}} \left(\sum_{j\notin B(i)} A_{i,j}\left(1+\frac{\epsilon}{2}\right)\left(x_j + \frac{\epsilon(\sigma x)_d}{3d}\right)\right)\\
&= \sum_{j\notin B(i)} A_{i,j}x_j + \frac{\epsilon(\sigma x)_d}{3d}\sum_{j\notin B(i)} A_{i,j}.
\end{aligned}$$

Now, $\sum_{j\notin B(i)} A_{i,j} \ge a_i^* \ge \frac{\mu}{d}\sum_{j\in B(i)} A_{i,j} = \frac{3d}{\epsilon}\sum_{j\in B(i)} A_{i,j}$. Therefore,

$$\frac{\epsilon(\sigma x)_d}{3d}\sum_{j\notin B(i)} A_{i,j} \ge \frac{\epsilon(\sigma x)_d}{3d}\cdot\frac{3d}{\epsilon}\cdot\sum_{j\in B(i)} A_{i,j} \ge \sum_{j\in B(i)} A_{i,j}x_j.$$

Together, this means that $\widetilde{A}_i^\top \widetilde{x} \ge A_i^\top x \ge 1$. Since this holds for all $i \in [r]$, $\widetilde{x} \in \mathcal{P}$. This completes the proof.

D.2 Proof of Lemma 3.6. We prove Lemma 3.6 here. We restate the relevant convex programs and the lemma here for convenience:

(Primal') $\quad \min \|x\|_{(w)}$ s.t. $Ax \ge \mathbf{1}_r, x \in \mathcal{Q}$. $\qquad$ (Dual) $\quad \min \|A^\top\lambda\|_{(w)}^*$ s.t. $\lambda \in \mathbf{\Delta}_r$

LEMMA 3.6. *Given a weight vector w, $\|x(w)\|_{(w)}\|A^\top\lambda(w)\|_{(w)}^* = 1$. Further, there is a reduced order ρ such that both $x(w), A^\top\lambda(w)$ satisfy ρ.*

For $j \in [d]$, denote the jth column of A as $A^{(j)} \in \mathbb{R}^r$. Recall that $S_1,\ldots,S_{N^r}$ form a partition of $[d]$ such that for $l \in [N^r]$, and for all $j, j' \in S_l$, $A^{(j)} = A^{(j')}$. Also recall that $\mathcal{Q}$ is the set of all vectors $x \ge 0$ such that $x_j = x_{j'}$ for all $j, j' \in S_l$, for all $l \in [N^r]$. From Lemma 3.3, $x(w) \in \mathcal{Q}$.

First, for all $x \in \mathcal{P}$ and $\lambda \in \mathbf{\Delta}_r$, we get by ordered Cauchy-Schwarz (Lemma 1.2) that $\|x\|_{(w)}\|A^\top\lambda\|_{(w)}^* \geq \lambda^\top Aw$. Since $x \in \mathcal{P}$, $Ax \geq \mathbf{1}_r$, and since $\lambda \in \mathbf{\Delta}_r$, $\lambda^\top Ax \geq 1$. Now, suppose that there is some $\lambda \in \mathbf{\Delta}_r$ such that $\|x(w)\|_{(w)}\|A^\top\lambda\|_{(w)}^* = 1$, i.e. equality holds. Then, since $\lambda(w) = \arg\min_{\lambda\in\mathbf{\Delta}_r}\|\lambda\|_{(w)}^*$, we get that

$$1 = \|x(w)\|_{(w)}\|A^\top\lambda\|_{(w)}^* \geq \|x(w)\|_{(w)}\|A^\top\lambda(w)\|_{(w)}^* \geq 1.$$

Then equality must hold everywhere, and in particular $\|x(w)\|_{(w)}\|A^\top\lambda(w)\|_{(w)}^* = 1$. Further, from ordered Cauchy-Schwarz, it is necessary that $x(w), A^\top\lambda(w)$ satisfy some order $\pi \in \mathrm{Perm}(d)$.

From Lemma 3.3, $x(w) \in \mathcal{Q}$, i.e., for all $j, j' \in S_l$, for all $l \in [N^r]$, $x(w)_j = x(w)_{j'}$. Similarly, $(A^\top\lambda(w))_j$ is the dot product of the jth column of A with $\lambda(w)$, and therefore $A^\top\lambda(w) \in \mathcal{Q}$ as well. Since $x, A^\top\lambda(w)$ both satisfy order π, π must induce a reduced order ρ on $S_1, \ldots, S_{N^r}$. This implies the lemma.

It remains to prove that there exists λ such that $\|x(w)\|_{(w)}\|A^\top\lambda\|_{(w)}^* = 1$. Our proof is along the lines of the proof of strong duality using Slater's conditions [5], although we use the properties of ordered norms at several places. We will need the following two lemmas:

Lemma D.1. *For vector $y \in \mathbb{R}^d$ such that $y_1 \geq \ldots \geq y_d \geq 0$, let $t_1 \leq t_2 \leq \ldots \leq t_T = d$ be indices such that*

$$y_1 = \ldots = y_{t_1} \geq y_{t_1+1} = \ldots = y_{t_2} \geq \ldots \geq y_{t_{T-1}+1} = \ldots = y_{t_T}.$$

Then for any weight vector w, $\|y\|_{(w)}^* = \max_{k\in[d]} \frac{(\sigma y)_k}{(\sigma w)_k}$ is achieved at some $k \in \{t_1, \ldots, t_T\}$.

Proof. The proof is straightforward, albeit somewhat involved. Denote $t_0 = 0$. It is sufficient to show that for all $i \in [T]$ and $t_{i-1} \leq k \leq t_i$, we have

$$\max\left\{\frac{(\sigma y)_{t_{i-1}}}{(\sigma w)_{t_{i-1}}}, \frac{(\sigma y)_{t_i}}{(\sigma w)_{t_i}}\right\} \geq \frac{(\sigma y)_k}{(\sigma w)_k}.$$

Denote $z = y_{t_{i-1}+1} = \ldots = y_{t_i}$. Consider $(1-\lambda)(\sigma y)_{t_{i-1}} + \lambda(\sigma y)_{t_i}$ for $\lambda = \frac{k-t_{i-1}}{t_i-t_{i-1}}$. Then $\lambda \in [0,1]$, and

$$(1-\lambda)(\sigma y)_{t_{i-1}} + \lambda(\sigma y)_{t_i} = (\sigma y)_{t_{i-1}} + \lambda z(t_i - t_{i-1}) = (\sigma y)_{t_{i-1}} + (k - t_{i-1})z = (\sigma y)_k.$$

Further,

$$\begin{aligned}(1-\lambda)(\sigma w)_{t_{i-1}} + \lambda(\sigma w)_{t_i} &= (\sigma w)_{t_{i-1}} + \lambda(w_{t_{i-1}+1} + \ldots + w_{t_i}) \\ &= (\sigma w)_{t_{i-1}} + (k - t_{i-1})\frac{w_{t_{i-1}+1} + \ldots + w_{t_i}}{t_i - t_{i-1}}.\end{aligned}$$

Since $w_{t_{i-1}+1} \geq \ldots \geq w_{t_i}$, we get that

$$\frac{w_{t_{i-1}+1} + \ldots + w_{t_i}}{t_i - t_{i-1}} \leq \frac{w_{t_{i-1}+1} + \ldots + w_k}{k - t_{i-1}}.$$

Plugging this back in, we get $(1-\lambda)(\sigma w)_{t_{i-1}} + \lambda(\sigma w)_{t_i} \leq (\sigma w)_k$. Therefore,

$$\frac{(\sigma y)_k}{(\sigma w)_k} \leq \frac{(1-\lambda)(\sigma y)_{t_{i-1}} + \lambda(\sigma y)_{t_i}}{(1-\lambda)(\sigma w)_{t_{i-1}} + \lambda(\sigma w)_{t_i}} \leq \max\left\{\frac{(\sigma y)_{t_{i-1}}}{(\sigma w)_{t_{i-1}}}, \frac{(\sigma y)_{t_i}}{(\sigma w)_{t_i}}\right\}.$$

∎

Lemma D.2. *For $\mu \in \mathbb{R}^r_{\geq 0}$,*

$$\sup_{x\in\mathcal{Q}} \mu^\top Ax - \|x\|_{(w)} = \begin{cases} 0 & \textit{if } \|\mu^\top A\|_{(w)}^* \leq 1, \\ \infty & \textit{otherwise.}\end{cases}$$

Proof. Denote $y = A^\top\mu$. Then $y \in \mathbb{R}^d$, and $y_j = (A^{(j)})^\top\mu$. If $\|y\|_{(w)}^* \leq 1$, we get from Lemma 1.2 (ordered Cauchy-Schwarz) that

$$y^\top x - \|x\|_{(w)} \leq \|y\|_{(w)}^*\|x\|_{(w)} - \|x\|_{(w)} \leq (\|y\|_{(w)}^* - 1)\|x\|_{(w)} \leq 0.$$

However, $0 \in \mathcal{Q}$, and therefore for $x = 0$, $y^\top x - \|x\|_{(w)} = 0$, so that $\sup_{x\in\mathcal{Q}} y^\top x - \|x\|_{(w)} = 0$

Now suppose that $\|y\|^*_{(w)} \geq 1$. Note that since $y_j = (A^{(j)})^\top \mu$, for all $j, j' \in S_l$ for some l, we get $y_j = y_{j'}$.

Relabel the indices $[N^r]$ so that for all $j \in S_l$ and $j' \in S_{l+1}$, $y_j \geq y_{j'}$. Further, relabel indices $[d]$ so that $S_1 = \{1, \ldots, |S_1|\}$, $S_2 = \{|S_1| + 1, \ldots, |S_1| + |S_2|\}$ etc. That is,

$$y_1 = \ldots = y_{|S_1|} \geq y_{|S_1|+1} = \ldots = y_{|S_1|+|S_2|} \geq \ldots \geq y_{d-|S_{N^r}|+1} = \ldots = y_d \geq 0.$$

By the previous lemma $\|y\|^*_{(w)} = \max_{k\in[d]} \frac{(\sigma y)_k}{(\sigma w)_k}$ achieved at some $k = |S_1| + \ldots + |S_l|$. For brevity, denote this number as k^*.

Define x such that $x_1 = x_2 = \ldots = x_{k^*} = \frac{\alpha}{(\sigma w)_{k^*}}$ and $x_{k^*+1} = \ldots = x_d = 0$ where α is an arbitrarily large number. Then $x \in \mathcal{Q}$ and $\|x\|_{(w)} = \alpha$. Further,

$$y^\top x = (\sigma y)^\top (\Delta x) = (\sigma y)_{k^*} \frac{\alpha}{(\sigma w)_{k^*}}.$$

Since $\frac{(\sigma y)_{k^*}}{(\sigma w)_{k^*}} = \|y\|^*_{(w)} > 1$, we get that $y^\top x - \|x\|_{(w)} = \alpha \left(\frac{(\sigma y)_{k^*}}{(\sigma w)_{k^*}} - 1\right)$, which can be arbitrarily large as α grows. This proves the second case as well. $\square$

We proceed to prove that there exists λ such that $\|x(w)\|_{(w)}\|A^\top\lambda\|^*_{(w)} = 1$. Let $\mathcal{A}$ be the set of points $(v_1, \ldots, v_r, t)$ such that there exists an $x \in \mathcal{Q}$ with $v_i \geq 1 - A_i^\top x$ for all $i \in [r]$ and $t \geq \|x\|_{(w)}$. It is easy to check that $\mathcal{A}$ is convex. Next, define $\mathcal{B} = \{(\underbrace{0, \ldots, 0}_{r}, s) : s < \|x(w)\|_{(w)}\}$. Clearly, $\mathcal{B}$ is convex. It is easy to see that $\mathcal{A} \cap \mathcal{B} = \emptyset$. Therefore, there is a separating hyperplane between $\mathcal{A}, \mathcal{B}$, i.e. there exist $\mu \in \mathbb{R}^d, \delta, \alpha \in \mathbb{R}$ such that

$$\mu^\top v + \delta t \geq \alpha \;\forall\, (v, t) \in \mathcal{A}, \tag{D.1}$$

$$\delta s < \alpha \;\forall s < \|x(w)\|_{(w)}. \tag{D.2}$$

The second equation implies that $\delta \geq 0$ since otherwise we can choose s to be arbitrarily small and δs becomes arbitrarily large. Then, we get $\delta\|x(w)\|_{(w)} \leq \alpha$.

Further, by a similar argument, $\mu \geq 0$. Applying eqn. (D.1), to point $(1 - A_1^\top x, \ldots, 1 - A_r^\top x, \|x\|_{(w)}) \in \mathcal{A}$ that for all $x \in \mathcal{Q}$, $\sum_{i\in[r]} \mu_i - \mu^\top Ax + \delta\|x\|_{(w)} \geq \alpha \geq \delta\|x(w)\|_{(w)}$.

Case I: $\mu = 0$. Then $\delta\|x\|_{(w)} \geq \alpha \geq \delta\|x(w)\|_{(w)}$. Since not both μ, δ can be zero, $\delta > 0$. Further, $\|x(w)\|_{(w)} > 0$, so if we pick $x = 0 \in \mathcal{Q}$, we get a contradiction.

Case II: $\mu \neq 0$, so we get that all for all $x \in \mathcal{Q}$, $\sum_{i\in[r]} \mu_i - \mu^\top Ax + \delta\|x\|_{(w)} \geq \alpha \geq \delta\|x(w)\|_{(w)}$. If $\delta = 0$, then $\sum_i \mu_i - \mu^\top Ax \geq 0$ for all $x \in \mathcal{Q}$. Pick arbitrarily large x again, giving a contradiction. Therefore, $\delta > 0$; assume without loss of generality that it is 1.

That is, for all $x \in \mathcal{Q}$, $\sum_i \mu_i - \mu^\top Ax + \|x\|_{(w)} \geq \|x(w)\|_{(w)}$. Taking infimum on the left-hand side and applying Lemma D.2, we get that $\sum_i \mu_i \geq \|x(w)\|_{(w)}$ with $\|\mu^\top A\|^*_{(w)} \leq 1$. Then $\lambda := \frac{\mu}{\sum_i \mu_i} \in \boldsymbol{\Delta}_r$. Therefore,

$$1 \geq \|\mu^\top A\|^*_{(w)} = \sum_i \mu_i \|\lambda^\top A\|^*_{(w)} \geq \|x(w)\|_{(w)}\|\lambda^\top A\|^*_{(w)}.$$

This finishes the proof.