The Heterogeneous Capacitated k-Center Problem

Deeparnab Chakrabarty
Microsoft Research India
deeparnab@gmail.com

Ravishankar Krishnaswamy
Microsoft Research India
ravishankar.k@gmail.com

Amit Kumar
Comp. Sci. & Engg., IIT Delhi
amitk@cse.iitd.ac.in

Abstract

In this paper we initiate the study of the *heterogeneous capacitated k-center problem*: given a metric space $X = (F \cup C, d)$, and a collection of capacities. The goal is to open each capacity at a unique facility location in F, and also to assign clients to facilities so that the number of clients assigned to any facility is at most the capacity installed; the objective is then to minimize the maximum distance between a client and its assigned facility. If all the capacities c_i's are identical, the problem becomes the well-studied *uniform capacitated k-center problem* for which constant-factor approximations are known [8, 23]. The additional choice of determining which capacity should be installed in which location makes our problem considerably different from this problem, as well the non-uniform generalizations studied thus far in literature. In fact, one of our contributions is in relating the heterogeneous problem to special-cases of the classical *santa-claus problem*. Using this connection, and by designing new algorithms for these special cases, we get the following results for Heterogeneous Cap-k-Center.

- A quasi-polynomial time $O(\log n/\varepsilon)$-approximation where every capacity is violated by $1+\varepsilon$.
- A polynomial time $O(1)$-approximation where every capacity is violated by an $O(\log n)$ factor.

We get improved results for the *soft-capacities* version where we can place multiple facilities in the same location.

1 Introduction

The capacitated k-center problem is a classic optimization problem where a finite metric space (X, d) needs to be partitioned into k clusters so that every cluster has cardinality at most some specified value L, and the objective is to minimize the maximum intra-cluster distance. This problem introduced by Bar-Ilan et al [8] has many applications [28, 29, 30]. One application is deciding placement of machine locations (centers of clusters) in a network scheduling environment where jobs arise in a metric space and the objective function has a job-communication (intra-cluster distance) and machine-load (cardinality) component [31]. The above problem is *homogeneous* in the sizes of the clusters, that is, it has the same cardinality constraint L for each cluster. In many applications, one would ask for a *heterogeneous* version of the problem where we have a different cardinality constraint for the clusters. For instance in the network scheduling application above, suppose we had machines of differing speeds. We could possibly load higher-speed machines with more jobs than lower-speed ones. In this paper, we study this heterogenous version.

Definition 1. (The Heterogeneous Cap-k-Center Problem[1].) *We are given a metric space $(X = F \cup C, d)$ where C and F represent the clients and facility locations. We are also given a collection of* heterogeneous *capacities: $(k_1, c_1), (k_2, c_2), \ldots, (k_P, c_P)$ with k_i copies of capacity c_i. The objective is to install these capacities at unique locations $F' \subseteq F$, and find an assignment $\phi : C \to F'$ of clients to these locations, such that for any $i \in F'$ the number of clients j with $\phi(j) = i$ is at most the capacity installed at i, and $\max_{j \in C} d(j, \phi(j))$ is minimized. A weaker version, which we call Heterogeneous Cap-k-Center with soft capacities, allows multiple capacities to be installed at the same location.*

[1] Technically, we should call our problem the Heterogeneous Capacitated k-Supplier Problem since we can only open centers in F. However, we avoid making this distinction throughout this paper.

Note that when all $c_p = L$ and $\sum_p k_p = k$, we get back the usual capacitated k-center problem. The Heterogeneous Cap-k-Center problem is relevant in many applications where the resources available are heterogenous. The machine placement problem was one example which has applications in network scheduling [33, 21] and distributed databases [29, 35]. Another example is that of vehicle routing problems with fleets of different speeds [18]. A third relevant application may be clustering; often clusters of equal sizes are undesirable [19] and explicitly introducing heterogeneous constraints might lead to desirable clusters. In this paper, we investigate the worst-case complexity of the Heterogeneous Cap-k-Center problem.

Bar-Ilan et al [8] gave a 10-approximation for the homogeneous capacitated k-center problem which was improved to a 6-factor approximation by Khuller and Sussmann [23]. One cannot get a better than 2-approximation even for the *uncapacitated* k-center problem [20]. More recently, the *non-uniform* capacitated k-center problem was considered [14, 1] in the literature: in this problem every facility $v \in F$ has a predetermined capacity c_v if opened (and 0 otherwise). We remark that the non-uniform version and our heterogeneous version seem unrelated in the sense that none is a special case of the other. Cygan et al [14] gave an $O(1)$-approximation for the problem which was improved to a 11-approximation by An et al [1].

Connection to Non-Uniform Max-Min Allocation Problems.

One main finding of this paper is the connection of the Heterogeneous Cap-k-Center problem to *non-uniform* max-min allocation (also known as Santa Claus [7]) problems, which underscores its difficulty and difference from the homogeneous capacitated k-center problems. We use the machine scheduling parlance to describe the max-min allocation problems.

Definition 2 ($Q||C_{min}$ and $Q|f_i|C_{min}$)**.** *In the*[2] $Q||C_{min}$ *problem, one is given m machines with demands $D_1, \ldots, D_m$ and n jobs with capacities $c_1, \ldots, c_n$, and the objective is to find an assignment of the jobs to machines satisfying each demand. In the* cardinality constrained *non-uniform max-min allocation problem, denoted as the $Q|f_i|C_{min}$ problem, each machine further comes with a cardinality constraint f_i, and a feasible solution cannot allocate more than f_i jobs to machine i. The objective remains the same. An α-approximate feasible solution assigns each machine i total capacity at least D_i/α.*

We now show how these problems arise as special cases of the Heterogeneous Cap-k-Center problem, even with soft capacities.

Remark 1 (Reduction from $Q|f_i|C_{min}$)**.** *Given an instance $\mathcal{I}$ of $Q|f_i|C_{min}$, construct the instance of Heterogeneous Cap-k-Center as follows. The capacities available to us are precisely the capacities of the jobs in $\mathcal{I}$. The metric space is divided into m groups $(F_1 \cup C_1), \ldots, (F_m \cup C_m)$ such that the distance between nodes in any group is 0 and across groups is 1. Furthermore, for $1 \le i \le m$, $|F_i| = f_i$ and $|C_i| = D_i$. Observe that the Heterogeneous Cap-k-Center instance has a 0-cost, capacity-preserving solution iff $\mathcal{I}$ has a feasible assignment.*

The $Q||C_{min}$ and $Q|f_i|C_{min}$ problems are strongly NP-hard.[3] Therefore, no non-trivial approximation to Heterogeneous Cap-k-Center, even the soft-version, exists unless we *violate* the capacities. This observation, which is in contrast to the homogeneous version, motivates us to look at bicriteria approximation algorithms.

Definition 3 ((a, b)-Bicriteria Approximation.)**.** *Given an instance of the Heterogeneous Cap-k-Center problem, an (a, b)-approximate feasible solution installs k_p units of c_p capacity, and assigns clients to facilities at most $a \cdot \mathsf{OPT}$ away and the number of clients assigned to a facility where a capacity c_p has been opened*[4] *is $\le \lceil bc_p \rceil$. An (a, b)-bicriteria approximation algorithm always returns an (a, b)-approximate feasible solution.*

[2] (Ab)using Graham's notation

[3] A simple reduction from 3-dimensional matching shows NP-hardness of $Q|f_i|C_{min}$ and $Q||C_{min}$ even when the demands and capacities are polynomially bounded.

[4] We add the ceiling to avoid pesky rounding issues.

Although bicriteria approximation algorithms may be unsatisfactory, sometimes these can give unicriteria approximations for other related problems. We mention one application that we alluded to above, and in fact was the starting point of this research, which may be of independent interest.

Definition 4. (Machine Placement Problem for Network Scheduling.) *The input is a metric space* $(X = F \cup C, d)$ *with jobs with processing times* p_j *at locations* C*. We are also given* P *machines with speeds* $s_1, s_2, \ldots, s_P$*. The goal is to find a placement of these machines on* F *and schedule the jobs on these machines so as to minimize the makespan. A job can be scheduled on a machine only after it reaches the location of the machine. In the "soft" version of the problem, multiple machines may be placed in the same location.*

Although we do not prove it in this paper, any (a, b)-bicriteria approximation algorithm for (soft) Heterogeneous Cap-k-Center problem implies an $O(a+b)$ approximation for the (soft) machine placement problem.

1.1 Results

The reduction in Remark 1 does not rule out arbitrarily small violations to the capacity. Indeed the $Q||C_{min}$ problem has a PTAS [4]. Our first couple of results give logarithmic approximation to the cost with $(1+\varepsilon)$-violations to the capacities.

Theorem 1.1. *Fix an* $\varepsilon > 0$*. There exists an* $(O(\log n/\varepsilon), (1+\varepsilon))$*-bicriteria approximation algorithm for the Heterogeneous Cap-*k*-Center problem running in time* $C_\varepsilon^{\tilde{O}(\log^3 n)}$ *for a constant* C_ε *depending only on* ε*. There exists an* $(O(\log n/\varepsilon), (1+\varepsilon))$*-bicriteria approximation algorithm for the Heterogeneous Cap-*k*-Center problem with soft capapcities running in time* $n^{O(1/\varepsilon)}$*.*

We are not aware of non-trivial results for the $Q|f_i|C_{min}$ problem (although, see Remark 2 below). We therefore call out the special case of the above theorem. This makes it rather improbable for $Q|f_i|C_{min}$ to be APX-hard, and we leave the design of a PTAS as a challenging open problem.

Theorem 1.2. *There is a QPTAS for the* $Q|f_i|C_{min}$ *problem.*

Our main technical meat of the paper is in reducing the logarithmic factor in the approximation to the distance. We can give $O(1)$-approximations if the violations are allowed to be $O(1)$ in the soft-capacity case and $O(\log n)$ in the general case. These algorithms run in polynomial time.

Theorem 1.3. *There is a polynomial time* $(O(1), O(\log n))$*-bicriteria approximation algorithm for the Heterogeneous Cap-*k*-Center problem.*

Theorem 1.4. *For any* $\delta > 0$*, there is a polynomial time* $(\tilde{O}(1/\delta), 2+\delta)$*-bicriteria approximation algorithm for the Heterogeneous Cap-*k*-Center problem with soft capacities.*

In particular we have polynomial time $O(1)$ and $O(\log n)$ approximation algorithms for the machine placement problem of Definition 4. Once again, we call out what we believe is the first polynomial time non-trivial approximation to $Q|f_i|C_{min}$.

Theorem 1.5. *There is a polynomial time logarithmic approximation algorithm for the* $Q|f_i|C_{min}$ *problem.*

We end the section by stating what we believe was the frontier of knowledge for the $Q|f_i|C_{min}$ problem.

Remark 2 (Known algorithms for $Q|f_i|C_{min}$)**.** To our knowledge, $Q|f_i|C_{min}$ has not been explicitly studied in the literature. However, in a straightforward manner one can reduce $Q|f_i|C_{min}$ to *non-uniform*, restricted-assignment max-min allocation problem (which we denote as $Q|restr|C_{min}$) where, instead of the cardinality constraint dictated by f_i, we restrict jobs to be assigned only to a subset of the machines: for every machine i and job j, j can be assigned to i iff $c_j \geq D_i/2f_i$. It is not hard to see that a ρ-approximation for the $Q|restr|C_{min}$ implies a 2ρ-approximation for the $Q|f_i|C_{min}$ instance.

Clearly $Q|restr|C_{min}$ is a special case of the general max-min allocation problem [12] and therefore for any $\varepsilon > 0$, there are $n^{O(1/\varepsilon)}$-time algorithms achieving $O(n^{\varepsilon})$-approximation. We do not know of any better approximations for $Q|restr|C_{min}$. The so-called Santa Claus problem is the *uniform* version $P|restr|C_{min}$ where all demands are the same [7]. This has a $O(1)$-approximation algorithm [16, 3, 32]. However all these algorithms use the configuration LP; unfortunately for the non-uniform version $Q|restr|C_{min}$, the configuration LP has an integrality gap of $\Omega(\sqrt{n})$ (this example is in fact the same example of [7] proving the gap for general max-min allocation – see Appendix 11.)

1.2 Outline of Techniques

We give a brief and informal discussion of how we obtain our results, referring to the formal definition whenever needed. In a nutshell, we obtain our results by reducing the Heterogeneous Cap-k-Center problem *to* the $Q|f_i|C_{min}$ problem (complementing the reduction *from* discussed in Remark 1). We provide two reductions – the first incurs logarithmic approximation to the cost but uses black-box algorithms for $Q|f_i|C_{min}$, the second incurs $O(1)$-approximation to the cost but uses "LP-based" algorithms for $Q|f_i|C_{min}$. Both these reductions proceed via *decomposing* the given instance of the Heterogeneous Cap-k-Center problem.

Warm-up: Weak Decompostion. Given a Heterogeneous Cap-k-Center instance, suppose we *guess* the optimal objective value, which we can assume to be 1 after scaling. Then, we construct a graph connecting client j with facility location i iff $d(i,j) \leq 1$. Then, starting at an arbitrary client and using a simple region-growing technique (like those used for the graph cut problems [25, 17]), we can find a set of clients J_1 of along with their neighboring facility locations $T_1 = \Gamma(J_1)$[5], such that: (a) the diameter of J_1 is $O(\log n/\varepsilon)$, and (b) the additional clients in the boundary $|\Gamma(T_1) \setminus J_1|$ is at most $\varepsilon|J_1|$. Now, we simply *delete* these boundary clients and charge them to J_1, incurring a capacity violation of $(1+\varepsilon)$. Moreover, note that in an optimal solution, *all* the clients in J_1 *must be* assigned to facilities opened in T_1. Using this fact, we define our first demand in the $Q|f_i|C_{min}$ instance by $D_1 = |J_1|$ and $f_1 = |T_1|$. Repeating this process, we get a collection of $\{(J_i, T_i)\}$ which naturally defines our $Q|f_i|C_{min}$ instance. It is then easy to show that an α-approximation to this instance then implies an $(O(\log n/\varepsilon), \alpha(1+\varepsilon))$-bicriteria algorithm for Heterogeneous Cap-k-Center.

LP-Based Strong Decompostion. It is not a-priori clear how to modify the above technique to obtain better factors for the cost. To get $O(1)$-approximations, we resort to linear programming relaxations. One can write the natural LP relaxation (L1)-(L6) described in Section 3 – the relaxation has y_{ip} variables which denote opening a facility with capacity c_p at i. Armed with a feasible solution to the LP, we prove a *stronger decomposition theorem* (Theorem 6.2): we show that we can delete a set of clients C_{del} which can be charged to the remaining ones, and then partition the remaining clients and facilities into *two* classes. One class $\mathcal{T}$ is the so-called *complete neighborhood sets* of the form $\{(J_i, T_i)\}$ with $\Gamma(J_i) \subseteq T_i$ as described above — we define our $Q|f_i|C_{min}$ instance using these sets. The other class $\mathcal{S}$ is of, what we call, *roundable* sets (Definition 5). Roundable sets have "enough" y-mass such that installing as many capacities as prescribed by the LP (rounded down to the nearest integer) supports the total demand incident on the set (with a $(1+\varepsilon)$-factor capacity violation). Moreover, the diameter of any of these sets constructed is $\tilde{O}(1/\varepsilon)$.

Technical Roadblock. It may seem that the above decomposition theorem implies a reduction to the $Q|f_i|C_{min}$ problem – for the class $\mathcal{T}$ form the $Q|f_i|C_{min}$ instance and use black-box algorithms, while the roundable sets in $\mathcal{S}$ are taken care of almost by definition. The nub of the problem lies in the *supply* of capacities to each of these classes. Sure, the $Q|f_i|C_{min}$ instance formed from $\mathcal{T}$ must have a solution if the Heterogeneous Cap-k-Center problem is feasible, *but only if all the k_p copies of capacity c_p are available to it.* However, we have already used up some of these copies to take care of the $\mathcal{S}$ sets, and what we actually have available for $\mathcal{T}$ is what the *LP prescribes.* And this can be very off (compared to the case when the $Q|f_i|C_{min}$ instance had all the k_p copies to itself). In fact, this natural LP relaxation has bad integrality gap (Remark 3), that is, although the LP is feasible, any assignment will violate capacities to $\Omega(n)$ factors.

The Supply Polyhedra. The above method would be fine if the supply prescribed by the LP to the complete-neighborhood sets in $\mathcal{T}$ would satisfy (or approximately satisfy) the demands of the machines in

[5]For $S \subseteq C \cup F$, $\Gamma(S)$ denotes the neighboring vertices of S.

the corresponding $Q|f_i|C_{min}$ instance. This motivates us to define *supply polyhedra* for $Q|f_i|C_{min}$ and other related problems. Informally, the supply polyhedron (Definition 8) of a $Q|f_i|C_{min}$ instance is supposed to capture all the vectors $(s_1, \ldots, s_n)$ such that s_j copies of capacity c_j can satisfy the demands of all the machines. Conversely, any vector in this polyhedron should also be a feasible (or approximately feasible) supply vector for this instance.

If such an object $\mathcal{P}$ existed, then we could strengthen our natural LP relaxation as follows. For *every* collection $\mathcal{T}$ of complete-neighborhood sets, we add a constraint (described as (L7)) stating that the fractional capacity allocated to the facilities in $\mathcal{T}$ should lie in the supply polyhedron of the corresponding $Q|f_i|C_{min}$ instance. Note that this LP has exponentially many constraints, and it is not clear how to solve it. However, we can use the "round-and-cut" framework exploited earlier in many papers [10, 11, 2, 15, 26, 27]. Starting with a solution (x, y), we use the strong decomposition theorem to obtain the set $\mathcal{T}$ and check if the restriction of y to the facilities in $\mathcal{T}$ lies in the supply polyhedron of the corresponding $Q|f_i|C_{min}$ instance. If yes, then we are done. If no, then we have obtained a separating hyperplance for the super-large LP(L1)-(L7), and we can run the ellipsoid algorithm. In sum, we obtain an algorithm which reduces the Heterogeneous Cap-k-Center problem to obtaining good supply polyhedra for the $Q|f_i|C_{min}$ problem (Theorem 6.1).

Supply Polyhedron for $Q|f_i|C_{min}$ and $Q||C_{min}$. Do good supply polyhedra exist for $Q|f_i|C_{min}$ or even the simpler $Q||C_{min}$ problem? Unfortunately, we show (Theorem 5.1) that there cannot exist *arbitrarily good* supply polyhedra. More precisely, there exists an instance of the $Q||C_{min}$ problem such that for *any* convex set which contains all feasible supply vectors, it also contains integer supply vectors which can't satisfy all demands even when a violation of 1.001 in capacities is allowed. This observation exhibits the limitation of our approach: we cannot hope to obtain $(1+\varepsilon)$-violation to the capacities for arbitrarily small ε.

Nevertheless, for $Q||C_{min}$ we describe a 2-approximate supply polyhedron (Theorem 5.2) based on the natural assignment LP, which along with our reduction proves Theorem 1.4. In fact, we show (Lemma 6.7) that for the Heterogeneous Cap-k-Center problem with soft capacities, the strong inequalities (L7) that we add for this 2-approximate supply polyhedron are already implied by (L1)-(L6).

For $Q|f_i|C_{min}$ we describe a supply polyhedron based on the *configuration LP* and prove that is $O(\log D)$-approximate (Theorem 5.4) where D is the ratio of maximum and minimum demand. This also implies a *polynomial time* $O(\log D)$-approximation algorithm for the $Q|f_i|C_{min}$ problem. As remarked in Remark 2, this is considerably better than any polynomial time algorithm implied before. We complement this by showing (Theorem 5.5, Section 9.1) that the integrality gap of the configuration LP is $\Omega(\log n/\log\log n)$. On the other hand, using fairly standard tricks of enumeration and rounding, we can provide a QPTAS for $Q|f_i|C_{min}$ (Theorem 1.2). We leave the complexity of $Q|f_i|C_{min}$ as an interesting open question.

1.3 Related Work

Capacitated Location problems have a rich literature although most of the work has focused on versions where each facility arrives with a predetermined capacity and the decision process is to whether open a facility or not. We have already mentioned the state of the art for capacitated k-center problems. For the capacitated facility location problem a 5-approximation is known via local search [6], while more recently an $O(1)$-approximate *LP-based* algorithm was proposed [2]. All these are true approximation algorithms in that they do not violate capacities. It is an outstanding open problem to obtain true approximations for the capacitated k-median problem. The best known algorithm is the recent work of Demirci and Li [15] who for any $\varepsilon > 0$ give a poly$(1/\varepsilon)$-approximate algorithm violating the capacities by $(1+\varepsilon)$-factor. The technique of this algorithm and its precursors [2, 26, 27] are similar to ours in that they follow the round-and-cut strategy to exploit exponential sized linear programming relaxations.

The $Q|f_i|C_{min}$ problem is a cardinality constrained max-min allocation problem. There has been some work in the scheduling literature on cardinality-constrained min-max problem. When all the machines are identical, the problem is called the k_i-partitioning problem [5]. When the number of machines is a constant, Woeginger [37] gives a FPTAS for the problem, and the best known result is a 1.5-approximation due to Kellerer and Kotov [22]. To our knowledge, the related speeds case has not been looked at. When the

machines are unrelated, Saha and Srinivasan [34] showed a 2-approximation; in fact this follows from the Shmoys-Tardos rounding of the assignment LP [36].

As we have discussed above, the Heterogeneous Cap-k-Center problem behaves rather differently than the usual homogeneous capacitated k-center problem. This distinction in complexity when we have heterogeneity in resource is a curious phenomenon which deserves more attention. A previous work [13] of the first two authors (with P. Goyal) looked at the (uncapacitated) k-center problem where the heterogeneity was in the radius of the balls covering the metric space. As in our work, even for that problem one needs to resort to bicriteria algorithms where the two criteria are cost and *number* of centers opened. That paper gives an $(O(1), O(1))$-approximation algorithm. In contrast, we do not wish to violate the number of capacities available at all (in fact, the problem is considerably easier if we are allowed to do so – we do not expand on this any further).

2 Roadmap

In Section 3, we set up the notation and key definitions which we will subsequently use in the remaining sections. Then in Section 4, we give our simpler weak-decomposition theorem which (upto a logarithmic factor in the distance objective) effectively reduces Heterogeneous Cap-k-Center to $Q|f_i|C_{min}$. To overcome this logarithmic loss in the distance objective, we turn to an LP-based approach and a stronger decomposition theorem. But to help us along the way, we introduce and state our main results about the so-called *supply polyhedra* for $Q|f_i|C_{min}$ in Section 5. In Section 6 we then state our strong decomposition theorem and show how it can be combined with good supply polyhedra to get Theorems 1.3 and 1.4. In the next Section 7, we prove the strong decomposition theorem. Subsequently, in Sections 8 and 9, we prove the existence of good supply polyhedra for $Q||C_{min}$ and $Q|f_i|C_{min}$. Finally in Section 10 we show that $Q|f_i|C_{min}$ admits a QPTAS, thereby proving Theorem 1.2.

3 Technical Preliminaries

Given an Heterogeneous Cap-k-Center instance, we start by guessing OPT. We either prove OPT is infeasible, or find an (a, b)-approximate allocation of clients to facilities. We define the bipartite graph $G = (F \cup C, E)$ where $(i, j) \in E$ iff $d(i, j) \leq \mathsf{OPT}$. If OPT is feasible, then the following assignment LP(L1)-(L6) must have a feasible solution. In this LP, we have opening variables y_{ip} for every $i \in F, p \in [P]$ indicating whether we open a facility with capacity c_p at location i. Recall that the capacities available to us are $c_1, c_2, \ldots, c_P$ – a facility with capacity c_p installed on it will be referred to as a *type p facility*. We have connection variables x_{ijp} indicating the fraction to which client $j \in C$ connects to a facility at location i where a type p facility has been opened. We force $x_{ijp} = 0$ for all pairs i, j and type p such that $d(i, j) > \mathsf{OPT}$.

$$\forall j \in C, \quad \sum_{i \in F} \sum_{p \in [P]} x_{ijp} \geq 1 \quad \text{(L1)}$$

$$\forall i \in F, p \in [P], \quad \sum_{j \in C} x_{ijp} \leq c_p y_{ip} \quad \text{(L2)}$$

$$\forall p \in [P], \quad \sum_{i \in F} y_{iq} \leq k_p \quad \text{(L3)}$$

$$\forall i \in F, j \in C, p \in [P], \quad x_{ijp} \leq y_{ip} \quad \text{(L4)}$$

$$\forall i \in F, \quad \sum_{p \in [P]} y_{ip} \leq 1 \quad \text{(L5)}$$

$$\forall i \in F, j \in C, p \in [P], \quad x_{ijp}, y_{ip} \geq 0 \quad \text{(L6)}$$

We say a solution (x, y) is (a, b)-feasible if it satisfies (L1), (L3)-(L6), and (L2) with the RHS replaced by $bc_p y^{\mathsf{int}}_{ip}$, and $x_{ijp} > 0$ only if $d(i, j) \leq a \cdot \mathsf{OPT}$, We desire to find an integral solution $(x^{\mathsf{int}}, y^{\mathsf{int}})$ which is (a, b)-feasible. The following lemma shows that it suffices just to round the y-variables.

Claim 3.1. *Given an (a, b)-feasible solution (x, y^{int}) where $y^{\mathsf{int}}_{ip} \in \{0, 1\}$, we can get an (a, b)-approximate solution to the Heterogeneous Cap-k-Center problem.*

Proof. Consider a bipartite graph with client nodes C on one side, and nodes of the form (i, p) with $y^{\mathsf{int}}_{ip} = 1$ on the other. The node (i, p) has capacity bc_p. Since (x, y^{int}) satisfies the conditions of the lemma, there is a fractional matching in this graph so that each client j is fractionally matched to an (i, p) so that $d(i, j) \leq a \cdot \mathsf{OPT}$, and the total fractional load on (i, p) is $\leq bc_p$. The theory of matching tells us that there

is an *integral* assignment of clients j to nodes (i,p) such that $d(i,j) \le a \cdot \mathsf{OPT}$ and the number of nodes matched to (i,p) is $\le \lceil bc_p \rceil$. Therefore opening a capacity c_p facility at i for all (i,p) with $y^{\mathsf{int}}_{ip} = 1$ gives an (a,b)-approximate solution to Heterogeneous Cap-k-Center. □

Henceforth, we focus on rounding the y-values. To this end, we make the following useful definition.

Definition 5 (Roundable Sets)**.** *A set of facilities $S \subseteq F$ is said to be (a,b)-roundable w.r.t (x,y) if*

(a) $\mathrm{diam}_G(S) \le a$
(b) there exists a rounding $y^{\mathsf{int}}_{ip} \in \{0,1\}$ for all $i \in S, p \in [P]$ such that
  1. $\sum_{q \ge p} \sum_{i \in S} y^{\mathsf{int}}_{iq} \le \lfloor \sum_{q \ge p} \sum_{i \in S} y_{iq} \rfloor$ *for all p, and*
  2. $\sum_{j \in C} d_j \sum_{i \in S, p \in [P]} x_{ijp} \le b \cdot \sum_{i \in S} \sum_{p \in [P]} c_p y^{\mathsf{int}}_{ip}$

If (x,y) were feasible, then for any (a,b)-roundable set, we can integrally open facilities to satisfy all the demand that was fractionally assigned to it taking a hit of a in the cost and a factor of b in the capacities. Furthermore, the number of open facilities is at most what the LP prescribes. Therefore, if we would be able to decompose the instance into roundable sets, we would be done. Unfortunately, that is not possible, and in fact the above LP has a large integrality gap even when we allow arbitrary violation of capacities.

Remark 3 (Integrality Gap for Heterogeneous Cap-k-Center)**.** *Consider the following instance. The metric space X is partitioned into $(F_1 \cup C_1) \cup \cdots \cup (F_K \cup C_K)$, with $|F_k| = 2$ and $|C_k| = K$ for all $1 \le k \le K$. The distance between any two points in $F_i \cup C_i$ is 1 for all i, while all other distances are ∞. The capacities available are $k_1 = K$ facilities with capacity $c_1 = 1$ and $k_2 = K-1$ facilities with capacity $c_2 = K$. It is easy to see that integrally any solution would violate capacities by a factor of $K/2$. On the other hand, there is a feasible solution for the above LP relaxation: for $F_k = \{a_k, b_k\}$, we set $y_{a_k 2} = 1 - 1/K$ and $y_{b_k 1} = 1$, and for all $j \in C_k$, we set $x_{a_k j 2} = 1 - 1/K$ and $x_{b_k j 1} = 1/K$.*

For the version with soft capacities, we do not have the constraint (L5) and the above integrality gap doesn't hold since we can install capacity K facilities on $K-1$ of the sets F_k's, $1 \le k \le K-1$, and K copies of the capacity 1 facilities at F_K. Note that although $|F_K| = 2$, we have opened K capacities.

In particular, note that for the (x,y) solution in the integrality gap example above there are no roundable sets. This motivates the definition of the second kind of sets.

Definition 6 (Complete Neighborhood Sets)**.** *A subset $T \subseteq F$ of facilities is called a* complete neighborhood *if there exists a client-set $J \subseteq C$ such that $\Gamma(J) \subseteq T$. In this case the subset J is said to be* responsible *for T. Additionally, a complete neighborhood T is said to be an α-complete neighborhood if* $\mathrm{diam}(T) \le \alpha$.

Remark 4 (Complete Neighborhood Sets to $Q|f_i|C_{min}$)**.** If we find a complete neighborhood T of facilities with say a set J of clients responsible for it, then we know that the optimal solution must satisfy all the demand in J by suitably opening facilities of sufficient capacity in S. Given a collection $\mathcal{T} = (T_1, \ldots, T_m)$ of disjoint α-complete neighborhood sets with J_i repsonsible for T_i, we can define an instance $\mathcal{I}$ of the $Q|f_i|C_{min}$ problem with m machines with demands $D_i = |T_i|$ and cardinality constraint $f_i = |T_i|$, and P jobs of capacities $c_1, \ldots, c_P$. The facilities opened by the OPT solution corresponds to a valid solution for $\mathcal{I}$; furthermore, any β-approximate solution for $\mathcal{I}$ corresponds to a (α, β)-approximate solution for the Heterogeneous Cap-k-Center problem restricted to clients in $\cup_\ell J_\ell$. Finally note that for Heterogeneous Cap-k-Center with soft-capacities, $\mathcal{I}$ is an instance of the $Q||C_{min}$ problem.

Note that the above integrality gap example is essentially a $Q|f_i|C_{min}$ instance with K machines of demand K each having cardinality constraint 2, and there are K jobs of capacity 1 and $K-1$ jobs with capacity K. This shows the assignment LP has bad integrality gap for the $Q|f_i|C_{min}$ problem (but not for $Q||C_{min}$).

Our final definition is that of (τ, ρ)-*deletable clients* who can be removed from the instance since they can be "ρ-charged" to the remaining clients no further than τ-away.

Definition 7 (Deletable Clients)**.** *A subset $C_{\mathsf{del}} \subseteq C$ of clients is ρ-deletable if there exists a mapping $\phi_{j,j'} \in [0,1]$ for $j \in C_{\mathsf{del}}$ and $j' \in C \setminus C_{\mathsf{del}}$ satisfying (a) $\sum_{j' \in C \setminus C_{\mathsf{del}}} \phi_{j,j'} = 1$ for all $j \in C_{\mathsf{del}}$, and(b) $\sum_{j \in C_{\mathsf{del}}} \phi_{j,j'} \leq \rho$ for all $j' \in C \setminus C_{\mathsf{del}}$. Furthermore, $\phi_{j,j'} > 0$ only if $d(j,j') \leq \tau \cdot \mathsf{OPT}$.*

The following claim shows we can remove C_{del} from consideration.

Claim 3.2. *Let C_{del} be a (ρ,τ)-deletable set. Given an (a,b)-approximate feasible solution (x', y^{int}) where x'_{ijp} is defined only for $j \in C \setminus C_{\mathsf{del}}$, we can extend x' to a general (x, y^{int}) solution which is $(a+\tau, b(1+\rho))$-approximate feasible.*

Proof. For any $j \in C_{\mathsf{del}}$, define $x_{ijp} = \sum_{j' \in C \setminus C_{\mathsf{del}}} x_{ij'p}\phi_{j,j'}$. We get for all $j \in C_{\mathsf{del}}$, $\sum_{i \in F} \sum_{p \in [P]} x_{ijp} = \sum_{i,p} \sum_{j' \in C \setminus C_{\mathsf{del}}} x_{ij'p}\phi_{j,j'} = \sum_{j' \in C \setminus C_{\mathsf{del}}} \phi_{j,j'} \left(\sum_{i,p} x_{ij'p}\right) \geq \sum_{j' \in C \setminus C_{\mathsf{del}}} \phi_{j,j'} = 1$, and for all $i \in F, p \in [P]$, $\sum_{j \in C_{\mathsf{del}}} x_{ijp} = \sum_{j \in C_{\mathsf{del}}} \sum_{j' \in C \setminus C_{\mathsf{del}}} x_{ij'p}\phi_{j,j'} = \sum_{j' \in C \setminus C_{\mathsf{del}}} x_{ij'p} \left(\sum_{j \in C_{\mathsf{del}}} \phi_{j,j'}\right) \leq \rho \sum_{j' \in C \setminus C_{\mathsf{del}}} x_{ijp} \leq b\rho c_p$. Therefore, in all we have $\sum_{j \in C} x_{ijp} \leq bc_p(1+\rho)$. □

4 Reduction to Max-Min Allocation via Region Growing

In this section, we give a reduction to $Q|f_i|C_{min}$ when we allow logarithmic approximations. We then show how we get Theorem 1.1 using this result.

Theorem 4.1. *Given an β-approximation algorithm for $Q|f_i|C_{min}$ (respectively, $Q||C_{min}$), for any $\varepsilon > 0$ there exists an $(O(\log n/\varepsilon), \beta(1+\varepsilon))$-approximate algorithm for the Heterogeneous Cap-k-Center problem (respectively, for the Heterogeneous Cap-k-Center problem with soft capacities).*

The main crux of the above proof is the following decomposition theorem obtained by the technique of region growing which was first used in the context of sparsest and multi cut problems [25, 17].

Theorem 4.2. *Given a guess OPT for Heterogeneous Cap-k-Center problem and any $\varepsilon > 0$, there is an algorithm which partitions the facilities F into a collection $\mathcal{T} = (T_1, \ldots, T_L)$ of $O(\log n/\varepsilon)$-complete neighborhoobd sets with J_ℓ responsible for T_ℓ, and the client set $C = C_{\mathsf{del}} \cup \bigcup_{\ell=1}^{L} J_\ell$ such that C_{del} is an $(O(\log n/\varepsilon), \varepsilon)$-deletable set.*

Proof. Recall G is the graph with $d(i,j) \leq \mathsf{OPT}$ for $(i,j) \in G$. Initially $\mathcal{T}$ and C_{del} are empty. We maintain a set of alive clients C' which is initially C. We maintain a working graph H which is initialized to G and is always a subgraph of G. Given a node j and an integer t, let $N_H^{(t)}(j)$ denote all the modes j' s.t. $d_H(j,j') < t$ and $\Gamma_H^{(t)}(j)$ denote all the nodes j' with $d_H(j,j') = t$. Note that for even t, we have $\Gamma_H^{(t)}(j) \subseteq C$, and for odd t, $\Gamma_H^{(t)}(j) \subseteq F$.

Till C' is empty, we perform the following operation. Select an arbitrary active client $j \in C'$. Find the smallest even t such that $|\Gamma_H^{(t)}(j)| < \varepsilon \cdot |N_H^{(t)}(j) \cap C|$. Since for all $s < t$ we have $|N^{(s+2)}(j) \cap C| \geq (1+\varepsilon)|N_H^{(s)}(j) \cap C|$, $|N^{(s+2)}(j) \cap C| > (1+\varepsilon)^{\frac{s}{2}}$. Therefore, $t \leq (2\ln n)/\varepsilon$, where $n = |C'|$. We define $T_\ell := N_H^{(t)}(j) \cap F$ and $J_\ell := N_H^{(t)} \cap C$; note that T_ℓ is an $O(\log n/\varepsilon)$-complete neighborhood which is responsible for J_ℓ. Furthermore, we add $J_{\mathsf{ext}} := \Gamma_H^{(t)}(j)$ to C_{del}, and since $|J_{\mathsf{ext}}| < \varepsilon |J_\ell|$ and $\mathrm{diam}(J_\ell) = O(\log n/\varepsilon)$, there exists a mapping $\phi_{j,j'}$ for $j \in J_{\mathsf{ext}}$ and $j' \in J_\ell$ such that $\sum_{j' \in J_\ell} \phi_{j,j'} = 1$ for all $j \in J_{\mathsf{ext}}$, and $\sum_{j \in J_{\mathsf{ext}}} \phi_{j,j'} \leq \varepsilon$ for all $j \in J_\ell$, and $\phi_{j,j'} > 0$ only if $d(j,j') = O(\log n/\varepsilon)$. That is, J_{ext} is a valid $(O(\log n/\varepsilon), \varepsilon)$-deletable set. Finally, we delete $T_\ell \cup J_\ell \cup J_{\mathsf{ext}}$ from H and $J_\ell \cup J_{\mathsf{ext}}$ from C'. We continue this procedure till C' is empty. □

Proof of Theorem 4.1. Given $\mathcal{T}$ we form the instance $\mathcal{I}$ of $Q|f_i|C_{min}$ (or $Q||C_{min}$ in case of soft-capacities) described in Remark 4. We provide k_p copies of job with capacity c_p. If OPT is feasible, then there must exist a feasible solution to $\mathcal{I}$. Furthermore, a β-approximate solution to $\mathcal{I}$ gives an $(O(\log n/\varepsilon), \beta)$-approximate solution to the clients in $C \setminus C_{\mathsf{del}}$. The theorem follows from Claim 3.2. □

As a corollary to Theorem 4.1, and using the fact that $Q||C_{min}$ has a PTAS [4], and our result (Theorem 1.2 proved in Section 10) that $Q|f_i|C_{min}$ has a quasipolynomial time approximation scheme (QPTAS), we get Theorem 1.1.

In Section 6 we state a much stronger decomposition theorem than Theorem 4.2 which exploits the LP solution. To exploit it for Heterogeneous Cap-k-Center problem, however, and prove an analogous theorem as Theorem 4.1, we need to understand certain polyhedra with respect to the $Q|f_i|C_{min}$ problem. We first do this in the next section.

5 Max-Min Allocation Problems and Supply Polyhedra

An instance of the $Q|f_i|C_{min}$ problem has m machines M with demands $D_1, \ldots, D_m$ and cardinality constraints $f_1, \ldots, f_m$, and n types of jobs J with capacities $c_1, \ldots, c_n$ respectively. In $Q||c_{min}$, there are no f_i's, or equivalently $f_i = \infty$.

A *supply vector* $(s_1, \ldots, s_n)$ where each s_j is a non-negative integer is called *feasible* for instances of these problems if the ensemble formed by s_j copies of jobs of capacity c_j can be allocated feasibly to satisfy all the demands. The *supply polyhedra* of these instances desires to capture these feasible supply vectors.

Definition 8 (Supply Polyhedron)**.** *Given an instance $\mathcal{I}$ for a max-min allocation problem, a polyhedron $\mathcal{P}(\mathcal{I})$ is called an α-approximate supply polyhedron if (a) all feasible supply vectors lie in $\mathcal{P}(\mathcal{I})$, and (b) given any non-negative integer vector $(s_1, \ldots, s_n) \in \mathcal{P}(\mathcal{I})$ there exists an assignment of the s_j jobs of capacity c_j to the machines such that machine i receives a total capacity of $\geq D_i/\alpha$.*

Ideally, we would like *exactly* supply polyhedra. One guess would be the convex hull of all the supply vectors; indeed this is the tightest polytope satisfying condition (a). Unfortunately, there are instances of $Q||C_{min}$ (and even for the uniform case $P||C_{min}$) where the convex hull of supply vectors contains infeasible integer points. This rules out exact or even $(1+\varepsilon)$-approximate supply polyhedra. In Theorem 9.13 in Section 9.1, we show a stronger lower bound of $\Omega(\log D/\log\log D)$ on the best approximation-factor of any supply polyhedra for $Q|f_i|C_{min}$.

Theorem 5.1. *There cannot exist α-approximate supply polyhedra (or convex sets) for $\alpha < 1.001$ for all $P||C_{min}$ instances.*

Proof. The example is almost similar to the example in [24] which was used to show integrality gap examples for strong LP relaxations for identical machines makespan minimization problem. We just sketch a proof here. Recall the Petersen Graph with 10 nodes and 15 edges which has the following key property: it has six perfect matchings $M_1, \ldots, M_6$ such that each edge (i, j) appears in exactly 2 of these matchings; however, its edge set cannot be partitioned into 3 perfect matchings. The vertices are numbered $0, 1, \ldots, 9$.

Now we can describe the instance. Fix k to be any positive integer. We have 15 types of jobs $p_{ij} = 2^i + 2^j$ for every edge (i, j) of the Petersen graph. We have $3k$ machines each with the same demand $D = \sum_{i=0}^{9} 2^i = 1023$. Consider the six supply vectors $s^{(t)}$ for $1 \leq t \leq 6$, which contains $3k$ copies of the job corresponding to edge (i, j) iff (i, j) is in the matching M_t. These are feasible supply vectors; indeed assign each of the $3k$ machines one jobs p_{ij} for $(i, j) \in M_t$. Now any convex set (in particular polyhedra) containing these six supply vectors must contain any convex combination. However the vector $\frac{1}{6}\sum_{t=1}^{6} s^{(t)}$ is an integer vector with k copies of each (i, j) for all edges of the Petersen Graph. This uses the fact that every edge is in exactly two perfect matchings. Since the edges of the Petersen graph can't be partitioned into 3 perfect matchings, any allocation of this supply vector must give one machine demand ≤ 1022. Therefore, there can't be any α-approximate supply polyhedra for $P||C_{max}$. □

Remark 5. At this point, we should underscore the difference between supply polyhedra and say LP relaxations for solving these allocation problems. Given an instance of say $Q||C_{max}$ *along with* the supply vector (which is one standard way the problems are stated), there does exist a polytope capturing all the feasible allocations. It is the integer hull. However, in general, the description of this integer hull uses the

supply vector in describing these constraints and therefore are non-linear when the supplies are variables. Nevertheless, as we discuss below, many LP relaxations studied in the literature imply supply polyhedra, and their integrality gaps imply the approximation factor for the polyhedra as well.

For our purposes, we need more technical conditions from the supply polyhedra. The first is a natural condition which states that if one moves the supply to higher capacity jobs, then feasibility remains. The second is related to polynomial time algorithms.

Definition 9. *A supply polyhedron $\mathcal{P}(\mathcal{I})$ is* upward-feasible *if the following condition holds. Reorder the jobs so that $c_1 \leq c_2 \leq \cdots \leq c_n$. If $(s_1, \ldots, s_n) \in \mathcal{P}$ and $(t_1, \ldots, t_n)$ is a non-negative vector satisfying $t \succeq_{\mathsf{suff}} s$, that is, $\sum_{k \geq i} t_k \geq \sum_{k \geq i} s_k$, then $(t_1, \ldots, t_n) \in \mathcal{P}$ as well.*

Definition 10 (γ-Approximate Separation.)**.** *A γ-approximate separation oracle for the supply polyhedron $\mathcal{P}(\mathcal{I})$ is a polynomial time procedure which given any $y \in \mathbb{R}^n_{\geq 0}$, either returns a hyperplane separating y from $\mathcal{P}$, or asserts that $y \in \mathcal{P}(\mathcal{I}')$ for the supply polyhedra of the instance $\mathcal{I}'$ where all demands have been reduced by a factor γ.*

5.1 Approximate Supply Polyhedra for $Q||C_{min}$

For $Q||C_{min}$, the following assignment LP acts as a good supply polyhedra.

$$\begin{aligned} \mathcal{P}_{\mathsf{ass}}(\mathcal{I}) = \{(s_1, \ldots, s_n) : & \\ \forall j \in J, \quad & \textstyle\sum_{i \in M} z_{ij} \leq s_j && \text{(A1)} \\ \forall i \in M, \quad & \textstyle\sum_{j \in J} z_{ij} \min(c_j, D_i) \geq D_i && \text{(A2)} \\ \forall i \in M, j \in J, \quad & z_{ij} \geq 0\} && \text{(A3)} \end{aligned}$$

Theorem 5.2. *For any instance $\mathcal{I}$ of $Q||C_{min}$, $\mathcal{P}_{\mathsf{ass}}(\mathcal{I})$ is an upward feasible, 2-approximate supply polyhedron with exact separation oracle.*

We defer the proof of the above theorem to Section 8.

5.2 Approximate Supply Polyhedra for $Q|f_i|C_{min}$

For $Q|f_i|C_{min}$, a candidate supply polyhedra would be (A1)-(A3) along with

$$\forall i \in M, \qquad \textstyle\sum_{j \in J} z_{ij} \;\leq\; f_i \qquad \text{(A4)}$$

which would enforce the cardinality constraint. Unfortunately, an example akin to that in Remark 3 shows that $\mathcal{P}_{\mathsf{ass}}$ is not an α-approximate supply polyhedron for $Q|f_i|C_{min}$ instances with $\alpha = o(n)$. We define a stronger supply polyhedron. However, at this juncture we state a theorem regarding (A1)-(A4) which is based on the ideas from Shmoys-Tardos rounding [36].

Theorem 5.3 (Shmoys-Tardos Rounding)**.** *Let (A1)-(A4) have a feasible solution z and let $C_i := \max_{j : z_{ij} > 0} c_j$. There is an integral assignment $z^{\mathsf{int}}_{ij} \in \{0, 1\}$ which satisfies (A1), (A4), and (A2) with the RHS replaced by $D_i - C_i$ for all i.*

Proof. (Sketch) We proceed as in the Shmoys-Tardos rounding [36] of the assignment LP. We convert the instance into a bipartite matching instance where on one side we have the jobs J with multiplicities $s_1, \ldots, s_n$, and on the other side we have the machines where we take $f'_i := \lceil \sum_{j \in J} z_{ij} \rceil \leq f_i$ copies of each machine. The solution z is converted to a (fractional) solution $\bar{z}$ on this bipartite graph where each job j is assigned by $\bar{z}$ to an extent of at most 1. Furthermore, for every machine i, each of its f'_i copies, except perhaps for the last one, gets $\bar{z}$-mass exactly 1. This assignment also has the property that for any machine i and job

j, if $\bar{z}$ assigns (fractionally) j to ℓ^{th} copy of i, then c_j is at least the total fractional demand assigned by z to the $(\ell+1)^{th}$ copy of this machine. Since each copy of a machine (except for the last copy) gets $\bar{z}$-mass exactly 1, there is an assignment of jobs to these copies such that each such copy of machine i gets exactly one job. We give machine i whatever its copies obtain; note that it obtains $f'_i \le f_i$ jobs. The total capacity of jobs allocated is therefore $\ge D_i - \Delta$ where Δ is the fractional capacity assigned to i's first copy. Since $\Delta \le C_1$, this proves the theorem. □

In other words, for instances where c_j's are $\ll D_i$'s, $\mathcal{P}_{\mathsf{ass}}$ is a good supply polyhedron. But in general we need a supply polyhedra with stronger constraints.

Let Supp be a multiset indicating infinitely many copies of jobs in J. For every machine i, let $\mathcal{F}_i := \{S \in \mathsf{Supp} : |S| \le f_i \text{ and } \sum_{j\in S} c_j \ge D_i\}$ denote all the feasible sets that can satisfy machine i. Let $n(S,j)$ denote the number of copies of job of type j.

$$\begin{aligned}
\mathcal{P}_{\mathsf{conf}}(\mathcal{I}) \;= \{(s_1,\ldots,s_n) :& \\
\forall i \in M, \quad & \textstyle\sum_S z(i,S) = 1 && \text{(C1)}\\
\forall j \in J, \quad & \textstyle\sum_{i\in M,S} z(i,S)n(S,j) \le s_j && \text{(C2)}\\
\forall i \in M, S \notin \mathcal{F}_i, \quad & z(i,S) = 0\} && \text{(C3)}
\end{aligned}$$

Theorem 5.4. *For any instance $\mathcal{I}$ of $Q|f_i|C_{min}$, $\mathcal{P}_{\mathsf{conf}}(\mathcal{I})$ is an upward feasible, $O(\log D)$-approximate supply polyhedron with $(1+\varepsilon)$-approximate separation oracle for any $\varepsilon > 0$, where $D := D_{\max}/D_{\min}$.*

As a corollary, we get Theorem 1.5. We complement this with an almost matching integrality gap.

Theorem 5.5. *The integrality gap of $\mathcal{P}_{\mathsf{conf}}$ is $\Omega\left(\frac{\log n}{\log\log n}\right)$. More precisely, there exists an instance $\mathcal{I}$ of $Q|f_i|C_{min}$ and a supply vector $(s_1,\ldots,s_n) \in \mathcal{P}_{\mathsf{conf}}(\mathcal{I})$, but in any feasible allocation of s_j jobs of capacity c_j to the machines, there exists some machine i receiving $\le O\left(D_i \frac{\log\log n}{\log n}\right)$.*

We defer the proofs of Theorem 5.4 and Theorem 5.5 to Section 9.

6 Heterogeneous Cap-k-Center via Supply Polyhedra

In this section, we prove the following theorem. One of the main engines will be a strong decomposition theorem (Theorem 6.2) which we will state here but will prove in the next section.

Theorem 6.1. *Suppose there exists β-approximate, upward feasible supply polyhedra for all instances of $Q|f_i|C_{min}$ (respectively, $Q||C_{min}$) which have γ-approximate separation oracles. Then for any $\delta \in (0,1)$, there is an $\left(\tilde{O}(1/\delta), \gamma\beta(1+5\delta)\right)$-bicriteria approximation algorithm for the Heterogeneous Cap-k-Center problem (respectively, with soft capacities).*

The above theorem and results about supply polyhedra imply the bicriteria algorithms for the Heterogeneous Cap-k-Center problem. Theorem 1.3 follows from the above theorem (instantialted with $\delta = 0.5$, say) and Theorem 5.4 after noting that $D_{\max}/D_{\mathsf{min}} \le n$ in the reduction we describe below. Theorem 1.4 follows from the above theorem and Theorem 5.2.

Before moving to the proof of Theorem 6.1, we state our main technical result which is a decomposition theorem which essentially states that given an Heterogeneous Cap-k-Center instance, we can partition the problem into roundable and complete neighborhood sets. The reader may want to recall the definitions of roundable sets (Definition 5), complete neighborhood sets (Definition 6), deletable sets (Definition 7), and the natural LP relaxation (L1)-(L6). It is perhaps instructive to compare the below theorem with Theorem 4.2. The proof of this theorem is rather technical, and we defer it to the next section.

Theorem 6.2 (Decomposition Theorem). *Given a feasible solution (x, y) to LP(L1)-(L6), and $\delta > 0$, there is a polynomial time algorithm which finds a solution x satisfying (L2) and(L4), and a decomposition as follows.*

1. *The facility set F is partitioned into two families $\mathcal{S} = (S_1, S_2, \ldots, S_K)$ and $\mathcal{T} = (T_1, T_2, \ldots, T_L)$ of mutually disjoint subsets. The client set C is partitioned into three disjoint subsets $C = C_{\mathsf{del}} \cup C_{\mathsf{black}} \cup C_{\mathsf{blue}}$ where C_{del} is a $(\tilde{O}(1/\delta), \delta)$-deletable subset.*
2. *Each $S_k \in \mathcal{S}$ is $(\tilde{O}(1/\delta), (1+\delta))$-roundable with respect to (x, y), and moreover, each client in C_{blue} satisfies $\sum_{i \in \mathcal{S}, p} \mathsf{x}_{ijp} \geq 1 - \frac{\delta}{100}$.*
3. *Each T_ℓ is a $\tilde{O}(1/\delta)$-complete neighborhood with a corresponding set J_ℓ of clients responsible for it, and $C_{\mathsf{black}} = \cup_{\ell=1}^{L} J_\ell$.*

Proof of Theorem 6.1. Let us first describe an approach which fails. Let (x, y) be a feasible solution to LP(L1)-(L6), and apply Theorem 6.2. Although the sets in $\mathcal{S}$ by definition are roundable which takes care of the clients in C_{blue}, the issue arises in assigning clients of C_{black}. In particular, $y_p^{\mathcal{T}} := \sum_{i \in \mathcal{T}} y_{ip}$ for all $1 \leq p \leq P$ which indicates the "supply" of capacity c_p available for the C_{black} clients. However, this may not be enough for serving all these clients (even with violation). That is, the vector $y^{\mathcal{T}}$ may not lie in the (approximate) supply polyhedra of the $Q|f_i|C_{min}$ instance defined by $\mathcal{T}$ as described in Remark 4.

That we fail is not surprising; after all, the LP has a bad integrality gap (Remark 3) and we need to strengthen it. We strengthen the LP by *explicitly requiring $y^{\mathcal{T}}$ to be in the supply polyhedra.* Since we do not know $\mathcal{T}$ before solving the LP (after all the LP generated it), we go ahead and require this for *all* collection of complete-neighborood sets. More precisely, for $\mathcal{T} := (T_1, \ldots, T_L)$ of L disjoint complete neighborhood sets, let $\mathcal{I}_{\mathcal{T}}$ denote the $Q|f_i|C_{min}$ instance a la Remark 4.

$$\forall \mathcal{T} := (T_1, \ldots, T_L) \text{ disjoint neighborhood subsets}, \quad y^{\mathcal{T}} \in \mathcal{P}(\mathcal{I}_{\mathcal{T}}) \tag{L7}$$

Note that this is a feasible constraint to add to LP(L1)-(L6). In the OPT solution, for any $\mathcal{T}$ there must be enough supply dedicated for the clients responsible for these complete neighborhood sets. So we have the following claim.

Claim 6.3. *If OPT is feasible, then there is a feasible solution to LP(L1)-(L6) along with (L7).*

We don't know how (and don't expect) to check feasibility of (L7) for all collections $\mathcal{T}$. However, we can still run ellipsoid method using the "round-and-cut" framework of [10, 11, 26, 27]. To begin with, we start with the LP(L1)-(L6) and obtain feasible solution (x, y). Subsequently, we apply the decomposition Theorem 6.2 to obtain the collection $\mathcal{T} = (T_1, \ldots, T_L)$. We then check if $y^{\mathcal{T}} \in \mathcal{P}(\mathcal{I}_{\mathcal{T}})$ or not. Since we have a γ-approximate separation oracle for $\mathcal{P}(\mathcal{I}_{\mathcal{T}})$, we are either guaranteed that $y^{\mathcal{T}} \in \mathcal{P}(\mathcal{I}'_{\mathcal{T}})$ where the ℓ^{th} demand is now D_ℓ/γ; or we get a hyperplane separating $y^{\mathcal{T}}$ from $\mathcal{P}(\mathcal{I}_{\mathcal{T}})$ which also gives us a hyperplane separating y from LP(L1)-(L7). This can be fed to the ellipsoid algorithm to obtain a new iterate (x, y) and the above process is repeated. The analysis of the ellipsoid algorithm tells us that in polynomial time we either prove infeasibility of the system (L1)-(L7) (implying the OPT guess for Heterogeneous Cap-k-Center is infeasible), or we are have (x, y) satisfying the premise of the following lemma.

Lemma 6.4. *Given (x, y) feasible for LP(L1)-(L6), let us apply the Decomposition Theorem 6.2 to obtain the instance $\mathcal{S}, \mathcal{T}$. Suppose the solution $y_p^{\mathcal{T}} := \sum_{i \in \mathcal{T}} y_{ip}$ lies in $\mathcal{P}(\mathcal{I}'_{\mathcal{T}})$ for the $Q|f_i|C_{min}$ (respectively, $Q||C_{min}$) instance $\mathcal{I}'_{\mathcal{T}}$ with L machines with $D_\ell := |J_\ell|/\gamma$ and $f_\ell := |T_\ell|$ (respectively, no cardinality constraints). Then we can obtain an $(\tilde{O}(1/\delta), \beta\gamma(1+\delta))$-approximate solution to the Heterogeneous Cap-k-Center problem (respectively, with soft-capacities).*

Proof. Since every set $S_k, 1 \leq k \leq K$, is $(\tilde{O}(1/\delta), (1+\delta))$-roundable, there exists a rounding y_{ip}^{int} for $i \in S_k$ such that

$$\forall p, \quad \sum_{q \geq p} \sum_{i \in S_k} y_{iq}^{\mathsf{int}} \leq \lfloor \sum_{q \geq p} \sum_{i \in S_k} y_{iq} \rfloor \tag{1}$$

Ideally, we would like to open a facility of capacity c_p at location i whenever $y^{\mathsf{int}}_{ip} = 1$. Unfortunately, the decomposition theorem doesn't have capacity constraints for individual p's but only their suffix sums. Instead we do the following. Define $y^{\mathcal{S}}_p := \sum_{i\in\mathcal{S}} y_{ip}$; LP(L3) implies that for all p, $y^{\mathcal{S}}_p + y^{\mathcal{T}}_p \le k_p$. For $1 \le p \le P$, define $s_p := \sum_{i\in\mathcal{S}} y^{\mathsf{int}}_{ip}$; (1) implies for all p, $\sum_{q\ge p} s_q \le \lfloor \sum_{q\ge p} y^{\mathcal{S}}_q \rfloor$ (since $\lfloor a\rfloor + \lfloor b\rfloor \le \lfloor a+b\rfloor$.)

Claim 6.5. *Given $(s_1,\ldots,s_P)$ satisfying $\sum_{q\ge p} s_q \le \lfloor \sum_{q\ge p} y^{\mathcal{S}}_q \rfloor$, there exists $(\tilde{s}_1,\ldots,\tilde{s}_P)$ satisfying for all p, (a) $\sum_{q\ge p} s_q \le \sum_{q\ge p} \tilde{s}_q \le \sum_{q\le p} y^{\mathcal{S}}_q$, and (b) $\tilde{s}_p \le k_p$.*

Proof. Simply define $\tilde{s}_p := \lfloor \sum_{q\ge p} y^{\mathcal{S}}_q \rfloor - \lfloor \sum_{q>p} y^{\mathcal{S}}_q \rfloor$. Therefore, $\sum_{q\ge p} \tilde{s}_q = \lfloor \sum_{q\ge p} y^{\mathcal{S}}_q \rfloor$ implying (a). To see (b), note $\tilde{s}_p \le \lceil y^{\mathcal{S}}_p \rceil \le k_p$, where we use the fact $\lfloor a+b\rfloor \le \lfloor a\rfloor + \lceil b\rceil$ for any non-negative a, b. □

The first inequality in (a) implies that at every location with $y^{\mathsf{int}}_{ip} = 1$, we can open a facility of capacity $c_q \ge c_p$. This, along with condition (b) of roundable sets (Definition 5), implies we can find a fractional solution x_{ijp} for $j \in C_{\mathsf{blue}}$ and (i,p) with $y^{\mathsf{int}}_{ip} = 1$ such that (a) $\sum_{i\in\mathcal{S},p\in[P]} x_{ijp} \ge 1$, (b) $x_{ijp} > 0$ only if $d(i,j) \le \mathrm{diam}(S_k) \le \tilde{O}(1/\delta)$, and (c) the capacity violation is $\le (1+\delta)(1-\delta/100)^{-1} \le (1+2\delta)$. Note the second term arises since from the decomposition theorem we have $\sum_{i\in\mathcal{S},p\in[P]} \mathsf{x}_{ijp} \ge 1 - \delta/100$. Thus we have fractionally assigned all C_{blue} clients to open facilities in $\mathcal{S}$.

Define, for $p \in [P]$, $t_p := k_p - \tilde{s}_p$, the number of facilities of capacity c_p we can open in $\mathcal{T}$. Note, by Claim 6.5, t_p's are non-negative.

Claim 6.6. $(t_1,\ldots,t_P) \in \mathcal{P}(\mathcal{I}'_{\mathcal{T}})$

Proof. By the Lemma premise, we have $y^{\mathcal{T}} \in \mathcal{P}(\mathcal{I}'_{\mathcal{T}})$. Now note that for all p,

$$\sum_{q\ge p} t_q = \sum_{q\ge p}(k_q - \tilde{s}_q) \ge \sum_{q\ge p} k_q - \sum_{q\ge p} y^{\mathcal{S}}_q \ge \sum_{q\ge p} y^{\mathcal{T}}_q$$

Since $\mathcal{P}(\mathcal{I}'_{\mathcal{T}})$ is upward-feasible, and $y^{\mathcal{T}} \in \mathcal{P}$, we get the claim. □

Since $\mathcal{P}(\mathcal{I}'_{\mathcal{T}})$ is β-approximate, we can find an allocation of the t_p copies of jobs of capacity c_p to the L machines of $\mathcal{I}'_{\mathcal{T}}$ such that machine ℓ gets at most f_ℓ jobs and total capacity $\ge D_\ell/\beta = |J_\ell|/\beta\gamma$. We install these capacities on the facilities of T_ℓ. Since the diameter of each T_ℓ is $\tilde{O}(1/\delta)$, we can find an x_{ijp} assignment of C_{black}-clients to these such that $\sum_{i\in\mathcal{T},p\in[P]} x_{ijp} \ge 1$ and $x_{ijp} > 0$ iff $d(i,j) = \tilde{O}(1/\delta)$, such that the capacity violation is at most $\alpha\beta$. This takes care of the clients in C_{black}. Finally, Claim 3.2 takes care of all the deleted clients C_{del} with an extra hit of $(1+\delta)$ on the capacity and additive $\tilde{O}(1/\delta)$ on the distance. □

This completes the proof of Theorem 6.1 for the general Heterogeneous Cap-k-Center problem. For the problem with soft capacities, the proof is exactly the same, except in the end, the instance $\mathcal{I}_{\mathcal{T}}$ is a $Q||C_{min}$ instance rather than a $Q|f_i|C_{min}$ one. □

We end this section by noting that for the Heterogeneous Cap-k-Center problem with soft-capacities, if we use the assignment supply polyhedra described in Section 5, then we do not need to run the ellipsoid algorithm. In particular, the inequality (L7) is implied (L1)-(L6) for $\mathcal{P}_{\mathsf{ass}}$ defined in (A1)-(A3).

Lemma 6.7. *Given any (x,y) feasible for LP(L1)-(L6) and any $\mathcal{T} = (T_1,\ldots,T_m)$, we have $y^{\mathcal{T}} \in \mathcal{P}_{\mathsf{ass}}(\mathcal{I}_{\mathcal{T}})$.*

Proof. Fix $\mathcal{T} = (T_1,\ldots,T_m)$ to be a collection of complete neighborhood sets. In the instance $\mathcal{I}_{\mathcal{T}}$ of $Q||C_{min}$, we have m machines with demands $D_\ell = |J_\ell|$, where J_ℓ is the client set responsible for T_ℓ. Recall, $y^{\mathcal{T}}_p := \sum_{i\in\mathcal{T}} y_{ip}$, and we need to find $z_{\ell,p}$ which satisfy the constraints (A1)-(A3) where $s_p := y^{\mathcal{T}}_p$.

The definition is natural: $z_{\ell,p} := \sum_{i\in T_\ell} y_{ip}$. Clearly it satisfies (A1) (indeed with equality). We now show it satisfies (A2). To this end, define for any $j \in J_\ell$, $x_{jp} := \sum_{i\in T_\ell} x_{ijp}$. Since $\Gamma(J_\ell) \subseteq T_\ell$, we get from (L1) that $\sum_p x_{jp} \ge 1$. In particular,

$$\sum_p \sum_{j\in J_\ell} x_{jp} \ge D_\ell \tag{2}$$

From (L4), we know $x_{ijp} \le y_{ip}$ and summing over all $i \in T_\ell$, we get for all $j \in T_\ell$, $x_{jp} \le \sum_{i \in T_\ell} y_{ip} = z_{\ell,p}$. In particular, $\sum_{j \in J_\ell} x_{jp} \le z_{\ell,p} D_\ell$. From (L2) we know for all $i \in T_\ell, p \in [P]$, $\sum_{j \in J_\ell} x_{ijp} \le c_p y_{ip}$. Summing over all $i \in T_\ell$, gives $\sum_{j \in J_\ell} x_{jp} \le c_p z_{\ell,p}$. Putting together, we get

$$\sum_{j \in J_\ell} x_{jp} \le z_{\ell,p} \min(D_\ell, c_p) \tag{3}$$

(2) and (3) imply that z satisfies (A2). □

Therefore, one can use the natural LP relaxation to obtain for any $\delta > 0$, a $\left(\tilde{O}(1/\delta), (2+\delta)\right)$-bicriteria approximation for the Heterogeneous Cap-k-Center problem with soft capacities. As it should be clear, this is a much more efficient algorithm.

7 Proof of Decomposition Theorem 6.2

We now prove the Decomposition Theorem 6.2. We first describe the algorithm which constructs the partitions into roundable and complete neighborhood sets. It is based on the following refinement of the region growing idea used in the proof of Theorem 4.2 – starting with an arbitrary client we first check if there is a small enough neighborhood (i.e., of small diameter) around it which is *non-expanding*, i.e., the number of clients on the boundary are much smaller than the number of clients inside the neighborhood. If so, we can remove the clients on the boundary and obtain a complete neighborhood set. Otherwise, we show that the total y-mass of the facilities in this neighborhood is quite high, and so, we can get a roundable set of facilities. The algorithm is written formally in Algorithm 1. We analyze the algorithm subsequently and show that it has the desired properties. Throughout, we let $\varepsilon := \min(1/12, \delta/100)$.

7.1 Algorithm Description

Our algorithm starts with the collections $\mathcal{S}$ and $\mathcal{T}$, and the clients sets C_{del}, C_{black}, and C_{blue} being empty. Once a facility is assigned into a set in $\mathcal{S}$ or $\mathcal{T}$, it is called an *assigned facility*. Similarly clients are assigned once they are added to $C_{\mathsf{del}} \cup C_{\mathsf{black}} \cup C_{\mathsf{blue}}$. As our algorithm forms these clusters, it changes the connection graph G by deleting all assigned clients and facilities. At any time, we denote the residual graph by H. We make a couple of definitions as in Section 4. Given a node i and an integer t, let $\Gamma_H^{(t)}(i)$ denote the nodes at distance (in H) exactly t from i. We let $N_H^{(t)}(i)$ denote the nodes at distance $< t$ from i. We use the shorthand $\Gamma_H(i)$ to denote $\Gamma_H^{(1)}(i)$. We extend this definition to subsets: $\Gamma_H(S) := \cup_{i \in S} \Gamma(i)$. Since we only delete vertices from the graph over the iterations, $\Gamma_H(S) \subseteq \Gamma_G(S)$ for all sets $S \subseteq V$ of the original set of vertices of G. For each of the partitions $\mathcal{S}$ and $\mathcal{T}$, let $L(\mathcal{S}) = \cup_{1 \le k \le K} S_k$ and $L(\mathcal{T}) = \cup_{1 \le \ell \le L} T_\ell$ denote the set of all locations in them respectively. Each set S_k (resp. T_ℓ) in the partitions $\mathcal{S}$ (resp. $\mathcal{T}$) will have a *root* facility $i_k \in S_k$ (resp. $i_\ell \in T_\ell$). We use $R(\mathcal{S})$ and $R(\mathcal{T})$ to denote the collection of roots $\cup_{1 \le k \le K} \{i_k\}$ and $\cup_{1 \le \ell \le L} \{i_\ell\}$ in $\mathcal{S}$ and $\mathcal{T}$ respectively.

A key definition in our algorithm is that of *effective capacity*. For every $i \notin L(\mathcal{S}) \cup L(\mathcal{T})$ and $p \in [P]$ with $y_{ip} > 0$, define

$$c_{\text{eff}}(i,p) := \frac{\sum_{j \in H \cap C} d_j x_{ijp}}{y_{ip}}$$

Recall that C denotes the set of all clients, and therefore, $H \cap C$ is the set of unassigned clients. Since all sets are initially empty, $c_{\text{eff}}(i,p)$ is well defined for all $i \in F, p \in [P]$. Whenever a facility enters $L(\mathcal{S}) \cup L(\mathcal{T})$, we fix its $c_{\text{eff}}(i,p)$ to be what it was at the iteration it entered. Since the set of clients in H only monotonically decreases, the effective capacity can only decrease over time. Each iteration of the algorithm (Line 3) begins by picking the pair $(i^\star, p^\star)$ with the highest effective capacity (Line 5).

Let $t^\star$ be the smallest *even* integer $> \lceil \frac{8}{\varepsilon} \ln\left(\frac{1}{\varepsilon}\right) \rceil$. We set $\bar{t}$ to be the smallest odd number t in $\{1, \ldots, t^\star\}$ such that $|\Gamma_H^{(t)}(i^\star) \cap C| < \varepsilon \cdot |N_H^{(t)}(i^\star) \cap C|$ if such a number exists, otherwise we set $\bar{t} = t^\star + 1$. That is, as in Section 4 we find the smallest distance at which the "client ball" stops expanding, however, we stop once we cross $t^\star$. Depending on what $\bar{t}$ is (although note that it is alway an odd number), we have two cases.

- (If $\bar{t} = t^\star + 1$): In this case, we have always witnessed expansion. We form a new component $S_k := N_H^{(t^\star)}(i^\star) \cap F$ to be all the facilities in the $t^\star$-ball around $i^\star$. Let $J_{\mathsf{int}} := N_H^{(t^\star-1)}(i^\star)$ be the clients whose neighbors in H lie in S_k. Since we witness expansion at all stages, note that

$$|J_{\mathsf{int}}| \geq (1+\varepsilon)^{t^\star/2} \cdot |\Gamma_H(i^\star)| > \frac{1}{\varepsilon^4} \cdot |\Gamma_H(i^\star)| \tag{4}$$

  It is not too hard to see that $|\Gamma_H(i^\star)| \geq c_{\mathrm{eff}}(i^\star, p^\star)$ (see Claim 7.4). Therefore, the (fractionally opened) facilities in S_k are servicing a large enough demand, in particular, more than the effective capacity of $i^\star$ and by the greedy choice, the effective capacity of any (i, p). This implies there will be considerable mass ($> 1/\varepsilon$) of facilities of the same type in S_k opened fractionally; opening floor-many of them violates capacity by only $(1+\varepsilon)$. So, we add S_k to $\mathcal{S}$, make $i^\star$ its root and add $i^\star$ to $R(\mathcal{S})$ (Line 13). We remove $S_k \cup J_{\mathsf{int}}$ from H and add J_{int} to C_{blue}. Additionally, for a technical reason, we remove from H any other client j with $\sum_{i \in \mathcal{S}, p} x_{ijp} > (1-\varepsilon)$; in this case we set $\mathsf{x}_{ijp} = x_{ijp}$ for all $i \in \mathcal{S}, p \in [P]$ and $\mathsf{x}_{ijp} = 0$ for all $i \notin \mathcal{S}, p \in [P]$ and add j to C_{blue} (Line 18).

- (If $\bar{t} \leq t^\star$): Let $J_{\mathsf{ext}} := \Gamma_H^{(\bar{t})}(i^\star)$ and let $J_{\mathsf{int}} := N_H^{(\bar{t})}(i^\star) \cap C$. We know that $|J_{\mathsf{ext}}| < \varepsilon \cdot |J_{\mathsf{int}}|$. Let $F_{\mathsf{tentative}} := N_H^{(\bar{t})}(i^\star) \cap F$ be the facilities in this ball. We delete J_{ext} from H and add it to C_{del}; we can do so since we can "charge it" to J_{int}. Ideally, we would like to add $(F_{\mathsf{tentative}}, J_{\mathsf{int}})$ as a complete-neighborhood to $\mathcal{T}$. While it is true that $\Gamma_H(J_{\mathsf{int}}) \subseteq F_{\mathsf{tentative}}$, the same may not be true in the original graph G since we delete vertices from it. More precisely, there could be a client $j \in J_{\mathsf{int}}$ and a facility $i \in \mathcal{S} \cup \mathcal{T}$ such that $(i, j) \in G$. Therefore, the algorithm branches into two sub-cases.

  (i) There is some root center $i_r \in R(\mathcal{S})$ close to $i^\star$ (Line 27). In this sub-case, the algorithm considers the closest such root i_r, and *augments* S_r to $S_r \cup F_{\mathsf{tentative}}$. As in the above case, we update C_{blue} by adding to it any client which has more than $(1-\varepsilon)$ of its fractional assignment to facilities in $\mathcal{S}$ (in particular, J_{int} will get added to this set)

  (ii) There is no such root (Line 33). In this case, the set $F_{\mathsf{tentative}}$ gets added as a new set T_ℓ to $\mathcal{T}$. Further, we add J_{int} to C_{black}. One of the invariants of our algorithm is that in later stages when we again encounter this case ($\bar{t} \leq t^\star$), any client $j \in J_{\mathsf{int}}$ at that stage *cannot* be a neighbor in G to a facility $i \in \mathcal{T}$.

This completes the description of the algorithm. We now show that the decomposition has the desired properties; in particular it satisfies the conditions in Theorem 6.2.

7.2 Algorithm Analysis

Firstly, the statement of the algorithm implies the partition $\mathcal{S}, \mathcal{T}$ and $C_{\mathsf{blue}} \cup C_{\mathsf{black}} \cup C_{\mathsf{del}}$. We analyze the algorithm to prove the properties needed. At the beginning of each iteration, we want to show that the algorithm maintains the following invariants:

I1. For any facility $i \in L(\mathcal{T})$, $\Gamma_G(i) \cap V(H) = \emptyset$, i.e., $\Gamma_G(i)$ contains no unassigned clients. Note that this holds even w.r.t. all the neighbors according to the original graph G.

I2. Similarly, for any facility $i \in L(\mathcal{S})$ added in Line 29 in Algorithm 1, $\Gamma_G(i)$ contains no unassigned clients.

Note that in I2, we count only those i which get added to $L(\mathcal{S})$ in Line 29, and so do not consider locations getting added in Line 13.

Claim 7.1. *The two invariants hold at the beginning of every iteration of the while loop in Line 3.*

Algorithm 1 Rounding algorithm for Theorem 6.2

procedure AlgDecompose(x, y)
  $t \leftarrow 1$; $k \leftarrow 1$; $\ell \leftarrow 1$; $H \leftarrow G$; $t^\star \leftarrow$ smallest *even* integer $> \lceil \frac{8}{\varepsilon} \ln\left(\frac{1}{\varepsilon}\right) \rceil$; $\mathsf{x} \leftarrow x$
  while there are no unassigned clients, i.e., $V(H) \cap C \neq \emptyset$ **do**
    update $c_{\text{eff}}(\cdot,\cdot)$ for all $i \in V(H) \cap F, p \in [P]$
    $(i^\star, p^\star) \leftarrow \arg\max_{i \in H, p \in [P]} c_{\text{eff}}(i,p)$ ▷ *pick location and type with largest effective capacity*
    $\bar{t} \leftarrow 1$ ▷ *Find the smallest odd $\bar{t}$ which is non-expanding*
    while $\bar{t} < t^\star$ **do**
      if $|\Gamma_H^{(t)}(i^\star)| < \varepsilon \cdot |N_H^{(t)}(i^\star) \cap C|$ **then**
        Exit While Loop
      else
        $\bar{t} \leftarrow \bar{t} + 2$
    if $\bar{t} = t^\star + 1$ **then** ▷ $N_H^{(t^\star)}(i^\star) \cap F$ *will be locally roundable*
      $S_k \leftarrow N_H^{(t^\star)}(i^\star) \cap F$ and $\mathcal{S} \leftarrow \mathcal{S} \cup S_k$
      define $i^\star$ to be the root of S_k, i.e., $R(\mathcal{S}) \leftarrow R(\mathcal{S}) \cup \{i^\star\}$
      $H \leftarrow H \setminus S_k$ ▷ *remove assigned facilities from H*
      $J_{\mathsf{int}} \leftarrow N_H^{(t^\star - 1)}(i^\star)$ ▷ *Note* $|J_{\mathsf{int}}| \geq \frac{1}{\varepsilon^4} \cdot |\Gamma(i^\star)|$
      for each $j \in H$ s.t $\sum_{i \in \mathcal{S}, p} x_{ijp} > (1-\varepsilon)$ **do** ▷ *In particular, this contains J_{int}*
        $C_{\mathsf{blue}} \leftarrow C_{\mathsf{blue}} \cup \{j\}$ and $H \leftarrow H \setminus \{j\}$ ▷ *assign clients to C_{blue}*
        $\mathsf{x}_{ijp} \leftarrow 0$ for all $i \notin \mathcal{S}, p \in [P]$ ▷ *Set x to 0 for facilities not in $\mathcal{S}$*
      $k \leftarrow k + 1$
    else ▷ $\bar{t} < t^\star$*, i.e., the ball is non-expanding.*
      $F_{\mathsf{tentative}} \leftarrow N_H^{(\bar{t})}(i^\star) \cap F$ ▷ $F_{\mathsf{tentative}}$ *are the ball's facilities.*
      $J_{\mathsf{ext}} \leftarrow \Gamma_H^{(\bar{t})}(i^\star)$ ▷ *Ball's boundary clients*
      $J_{\mathsf{int}} \leftarrow N_H^{(\bar{t})}(i^\star) \cap C$. ▷ *Ball's internal clients*
      $C_{\mathsf{del}} \leftarrow C_{\mathsf{del}} \cup J_{\mathsf{ext}}$ and define ϕ appropriately ▷ *delete J_{ext} and charge to J_{int}*
      $H \leftarrow H \setminus J_{\mathsf{ext}}$ ▷ *remove deleted clients from H*
      if $\text{dist}_G(i^\star, R(\mathcal{S})) \leq \frac{16}{\varepsilon} \ln\left(\frac{1}{\varepsilon}\right)$ **then** ▷ $i^\star$ *is close to some root in $R(\mathcal{S})$*
        let $i_r = \arg\min_{i \in R(\mathcal{S})} \text{dist}_G(i^\star, i)$ ▷ i_r *is the nearby root from $\mathcal{S}$*
        $S_r \leftarrow S_r \cup F_{\mathsf{tentative}}$ ▷ *add these facilities to S_r*
        for each $j \in H$ s.t $\sum_{i \in \mathcal{S}, p} x_{ijp} > (1-\varepsilon)$ **do** ▷ *In particular, this contains J_{int}*
          $C_{\mathsf{blue}} \leftarrow C_{\mathsf{blue}} \cup \{j\}$ and $H \leftarrow H \setminus \{j\}$ ▷ *assign clients to C_{blue}*
          $\mathsf{x}_{ijp} \leftarrow 0$ for all $i \notin \mathcal{S}, p \in [P]$ ▷ *Set x to 0 for facilities not in $\mathcal{S}$*
      else ▷ $F_{\mathsf{tentative}}$ *will be a $\tilde{O}(1/\varepsilon)$-complete neighborhood of J_{int}*
        add a new part $T_\ell := F_{\mathsf{tentative}}$ to $\mathcal{T}$
        $J_\ell \leftarrow J_{\mathsf{int}}$, $C_{\mathsf{black}} \leftarrow C_{\mathsf{black}} \cup J_{\mathsf{int}}$, and $H \leftarrow H \setminus J_1$ ▷ *assign clients to C_{black}*
        $\ell \leftarrow \ell + 1$
      $H \leftarrow H \setminus F_{\mathsf{tentative}}$ ▷ *remove assigned facilities from H*
    $t \leftarrow t + 1$ ▷ *Iteration Counter*
  return $\mathcal{S}, \mathcal{T}$

Proof. We show this by induction over the number of iterations t. Clearly, at $t = 1$, $L(\mathcal{T})$ and $L(\mathcal{S})$ are empty, so the invariants hold tautologically. Suppose they hold for iterations upto i. We show that they also hold at the end of the t^{th} iteration, and hence they hold at the beginning of the $(t+1)^{th}$ iteration, thus completing the proof. To this end, consider the t^{th} iteration.

We first show that I1 continues to hold at the end of this iteration. Note that we only need to check if I1 holds for any new facilities added to $L(\mathcal{T})$ in this iteration, which only happens in Line 34. In this case, consider any facility $i \in T_\ell$, the set of facilities added to $L(\mathcal{T})$, and consider the neighborhood $\Gamma_G(i)$: in this set, some clients are already in $C_{\mathsf{blue}} \cup C_{\mathsf{black}} \cup C_{\mathsf{del}}$ in which case they would have been deleted from H in earlier iterations. By definition, the remaining clients belong to $J_{\mathsf{int}} \cup J_{\mathsf{ext}}$, since $J_{\mathsf{int}} \cup J_{\mathsf{ext}}$ contains all remaining neighbors of T_ℓ. But clients in J_{int} are added to C_{black}, and those in J_{ext} are added to C_{del}, hence i would have no clients as neighbors in H at the end of this iteration. Applying this to all $i \in T_\ell$ completes the proof.

We now show that I2 continues to hold at the end of this iteration. Similar to the above proof, note that we only need to check if I2 holds for any new facilities added to $L(\mathcal{S})$ in Line 29. In this case, consider any facility $i \in F_{\mathsf{tentative}}$, the set of facilities added to $L(\mathcal{S})$, and consider the neighborhood $\Gamma_G(i)$: in this neighborhood, some clients are already in $C_{\mathsf{blue}} \cup C_{\mathsf{black}} \cup C_{\mathsf{del}}$ in which case they would have been deleted from H in earlier iterations. By definition, the remaining clients belong to $J_{\mathsf{int}} \cup J_{\mathsf{ext}}$, since $J_{\mathsf{int}} \cup J_{\mathsf{ext}}$ contains all remaining neighbors of $F_{\mathsf{tentative}}$. But clients in J_{ext} are added to C_{del}, and we now show that all clients in J_{int} would be colored blue in Line 31, hence showing that i would have no clients as neighbors in H at the end of this iteration. Indeed, consider any client $j \in J_{\mathsf{int}}$: by definition, it was in H at the beginning of this iteration and so by invariant I1, there are no edges in H between j and any location $i' \in L(\mathcal{T})$. So all neighbors in $\Gamma_G(j)$ which have already been deleted belong to $L(\mathcal{S})$. Moreover, $F_{\mathsf{tentative}}$ includes all remaining neighbors of j. Hence, for any such j, we know that $\sum_{i \in L(\mathcal{S}),p} x_{ijp} = 1$, and so it would be added to C_{blue} in Line 31. □

We now show that the deleted clients C_{del} can be charged to C_{blue} and C_{black}.

Claim 7.2. *C_{del} is a $(\tilde{O}(1/\delta), \delta)$-deletable set.*

Proof. We add vertices to C_{del} only in line 25, and at that point it must be that $|J_{\mathsf{ext}}| \le \varepsilon \cdot |J_{\mathsf{int}}|$. As in the proof of Theorem 4.2, we can define the assignment $\phi_{j,j'}$ for $j \in J_{\mathsf{ext}}$ and $j' \in \mathsf{int}$. Furthermore, as in the proof of Claim 7.1, our algorithm makes sure that the client set J_{int} gets added to C_{blue} or C_{black} (in lines 31 and 35). Therefore, these clients in J_{int} will never be images of ϕ again, thus completing the proof. □

We will now show that the sets $\{T_\ell\}$ in the family $\mathcal{T}$ form $\tilde{O}(1/\varepsilon)$-complete neighborhoods supported by the corresponding client-sets $\{J_\ell\}$.

Lemma 7.3. *Consider an iteration when a new set T_ℓ is added to $\mathcal{T}$ in line 34. The set T_ℓ is a $\tilde{O}(1/\varepsilon)$-complete neighborhood supported by the set of clients J_ℓ (defined in line 35).*

Proof. Firstly, the diameter of the new set is at most $\frac{16}{\varepsilon}\ln(1/\varepsilon)$, since for every $i \in T_\ell$ is $d(i, i^\star) \le t^\star$ for the $i^\star$ facility identified in line 5. To complete the proof, we show that T_ℓ is supported by the set J_ℓ defined in line 35, which is same as J_{int}. We establish this by showing that $\Gamma_G(J_{\mathsf{int}}) \subseteq T_\ell$ (recall definition 6).

To this end, consider a client $j \in J_{\mathsf{int}}$. At the beginning of this iteration, j is a client in H. We claim that at the beginning of this iteration, $\Gamma_H(j) = \Gamma_G(j)$ (i.e., no neighboring facility has already been assigned in earlier iterations). Indeed, suppose not, and let i be some facility which is present in $\Gamma_G(j)$ but not in $\Gamma_H(j)$. We first observe that i cannot be in $L(\mathcal{T})$ as that would violate invariant I1 at the beginning of this iteration — (i, j) would form the violated pair. Similarly, we note that i cannot be added to $\mathcal{S}$ in line 13 in an earlier iteration — because then the distance between $i^\star$ and $R(\mathcal{S})$ would be at most $\frac{16}{\varepsilon}\ln(1/\varepsilon)$ (via the path $i^\star \to j \to i \to R(\mathcal{S})$), so this is a contradiction to the fact that the algorithm is in the branch executing line 34. Finally, we note that i cannot be added to $\mathcal{S}$ in line 29 in an earlier iteration, as that would violate invariant I2 at the beginning of this iteration — again (i, j) would form the violated pair. So we can conclude that $\Gamma_H(j) = \Gamma_G(j)$ and thus that the entire neighborhood of j is contained in $\{i^\star\} \cup A$ which is added to T_ℓ in this iteration. Repeating this argument for all j shows that $\Gamma_G(J_{\mathsf{int}}) \subseteq T_\ell$. □

We now turn our attention to proving that the sets in $\mathcal{S}$ are locally roundable. Toward this end, we begin with the following useful claim.

Claim 7.4. *For any set $S_k \in \mathcal{S}$, we have $\sum_{j\in C}\sum_{i\in S_k}\sum_{p\in[P]} d_j \mathsf{x}_{ijp} \geq \frac{1}{\varepsilon^3}\cdot \max_{i\in S_k, p\in[P]} c_{\text{eff}}(i,p)$*

Proof. Given a set $S_k \in \mathcal{S}$, there is a subset S'_k which was formed in line 13 and then got augmented in line 29. Note that the root $i^\star$ of S_k lies in S'_k. First we prove the claim for the set S'_k. By the definition of $(i^\star, p^\star)$ (line 5), we know that $\max_{i\in S'_k, p\in[P]} c_{\text{eff}}(i,p) = c_{\text{eff}}(i^\star,p^\star)$.Furthermore, by definition,

$$c_{\text{eff}}(i^\star,p^\star) = \sum_{j\in H\cap C} d_j \frac{\mathsf{x}_{i^\star jp^\star}}{y_{i^\star p^\star}} \leq \sum_{j\in\Gamma_H(i^\star)} d_j = |\Gamma_H(i^\star)|$$

The inequality follows because we know (a) that $\mathsf{x}_{ijp^\star} = x_{ijp^\star} > 0$ only for $j \in \Gamma_H(i^\star)$ (i.e., only clients which are neighboring $i^\star$ can be serviced by $i^\star$), and (b) that $x_{i^\star jp^\star} \leq y_{i^\star p^\star}$ (using inequality (L4)). Now note that for any $j \in J_{\mathsf{int}}$, since $j \in H$, we have that $\sum_{i\in L(\mathcal{S}),p\in[P]} x_{ijp} \leq 1-\varepsilon$ (otherwise it would have been added to C_{blue} in an earlier iteration and deleted from H), and so $\sum_{i\in S_k,p\in[P]} x_{ijp} \geq \varepsilon$ (because S_k includes all the neighbors of j not already in $L(\mathcal{S})$, and j has no edge to any facility in $L(\mathcal{T})$ even in the original graph G by invariant I1, so the fractional demand from j to vertices in $L(\mathcal{T})$ is 0). Therefore,

$$|J_{\mathsf{int}}| = \sum_{j\in J_{\mathsf{int}}} d_j \leq \frac{1}{\varepsilon}\sum_{j\in J_{\mathsf{int}}} d_j \left(\sum_{i\in S_k,p\in[P]} x_{ijp}\right)$$

Since $|J_{\mathsf{int}}| \geq \frac{1}{\varepsilon^4}\cdot|\Gamma(i^\star)|$ (4), we have the claim for S'_k.

Subsequently, since $i^\star$ got added to H, $c_{\text{eff}}(i^\star,p^\star)$ remains unchanged and since $c_{\text{eff}}(i,p)$ can only decrease, we see $c_{\text{eff}}(i^\star,p^\star) = \max_{i\in S_k,p\in[P]} c_{\text{eff}}(i,p)$. Therefore, even when we add more facilities to S_k later in the algorithm (line 29), the RHS of the inequality in the statement of the Claim does not change. Observe that the LHS can only increase since the set S_k can grow during the algorithm. □

Claim 7.5. *At the end of the algorithm, for all $j \in C_{\mathsf{blue}}$, $\sum_{i\in\mathcal{S},p\in[P]} \mathsf{x}_{ijp} > (1-\frac{\delta}{100})$.*

Proof. This follows since we only add clients to C_{blue} when their fractional allocation to $\mathcal{S}$ exceeds $(1-\varepsilon)$. □

Lemma 7.6. *Each set $S_k \in \mathcal{S}$ is a $\left(\tilde{O}\left(\frac{1}{\delta}\right),(1+\delta)\right)$-roundable set with respect to (x,y).*

Proof. (Diameter) We claim that $\text{diam}(S_k) \leq \frac{50}{\varepsilon}\ln(\frac{1}{\varepsilon})$ for every $S_k \in \mathcal{S}$. We show by induction that for each $S_k \in \mathcal{S}$, $\text{dist}_G(i,i_k) \leq \frac{25}{\varepsilon}\ln(\frac{1}{\varepsilon})$ for every $i \in S_k$, where i_k is the root of S_k. When we add a new set S to $\mathcal{S}$ (in Line 13), S is the set $N_H^{(t^\star)}(i^\star)\cap F$ and so $\text{dist}_G(i,i^\star) \leq t^\star \leq \frac{9}{\varepsilon}\ln(\frac{1}{\varepsilon})$ for any $i \in S$. Now, consider the case when we augment an existing set in $\mathcal{S}$ (as in Line 29). Again, using the notation in the algorithm, suppose $i_k = \arg\min_{i\in R(\mathcal{S})}\text{dist}_G(i,i^\star)$, and let i_k be the root of $S_k \in \mathcal{S}$. Then, $\text{dist}_G(i^\star,i_k) \leq \frac{16}{\varepsilon}\ln(\frac{1}{\varepsilon})$. Since $\text{dist}_G(i^\star,i') \leq \frac{9}{\varepsilon}\ln(\frac{1}{\varepsilon})$ for any $i' \in F_{\mathsf{tentative}}$, we see that $\text{dist}_G(i_k,i') \leq \frac{25}{\varepsilon}\ln(\frac{1}{\varepsilon})$ for any $i' \in F_{\mathsf{tentative}}$. So the desired claim follows by induction. Since $\varepsilon = O(\delta)$, the diameter condition follows.

(Roundability) We now show that there is a rounding of y^{int}_{ip} values for $i \in S_k$ such that

1. $\sum_{q\geq p}\sum_{i\in S_k} y^{\mathsf{int}}_{iq} \leq \lfloor\sum_{q\geq p}\sum_{i\in S} y_{iq}\rfloor$, and
2. $\sum_{j\in C} d_j \sum_{i\in S_k,p\in[P]} \mathsf{x}_{ijp} \leq (1+\delta)\cdot\sum_{i\in S}\sum_{p\in[t]} c_p y^{\mathsf{int}}_{ip}$

For simplicity, let us use $\Delta := (1+\varepsilon)$. Define $A_u := \{(i,p), i \in S_k, p \in [P] : c_{\text{eff}}(i,p) \in [\Delta^u,\Delta^{u+1})\}$ and let $\max_{i\in S_k,p\in[P]} c_{\text{eff}}(i,p) \in [\Delta^U,\Delta^{U+1})$. From Claim 7.4, we have

$$\Delta^U \leq \max_{i\in S_k,p\in[P]} c_{\text{eff}}(i,p) \leq \varepsilon^3\sum_{j\in C} d_j\sum_{i\in S_k,p\in[P]}\mathsf{x}_{ijp}$$

We also assume we have available capacities of value Δ^u available to us; this is without loss of generality by setting there k_p value to 0 if there don't exist any.

Define $\alpha_u := \sum_{(i,p)\in A_u} y_{i,p}$. For all values of u, *arbitrarily* choose $\lfloor \alpha_u \rfloor$ different facilities F_u in S_k; that there are so many is implied by (L4) of the LP. For each u and for each $i \in F_u$, set $y^{\mathsf{int}}_{i\Delta^u} = 1$. For every other (i,p), set $y^{\mathsf{int}}_{ip} = 0$. Note that $\sum_{i\in S,p\in[P]} c_p y^{\mathsf{int}}_{ip} = \sum_u \lfloor \alpha_u \rfloor \Delta^u$.

We claim that y^{int} satisfies the two conditions of the roundability property. We check Condition 1 first. Let $p \in [P]$ and let s be the index such that $\Delta^s < p \le \Delta^{s+1}$. Then

$$\begin{aligned}
\sum_{q\ge p}\sum_{i\in S_k} y^{\mathsf{int}}_{iq} &= \sum_{u\ge s+1}\sum_{i\in S_k} y^{\mathsf{int}}_{i\Delta^u} = \sum_{u\ge s+1}\lfloor\alpha_u\rfloor \le \lfloor\sum_{u\ge s+1}\alpha_u\rfloor && (5)\\
&= \lfloor\sum_{u\ge s+1}\sum_{(i,q)\in A_u} y_{iq}\rfloor \le \lfloor\sum_{q:c_q\ge\Delta^{s+1}}\sum_{i\in S_k} y_{iq}\rfloor && (6)\\
&\le \lfloor\sum_{q\ge p}\sum_{i\in S} y_{iq}\rfloor && (7)
\end{aligned}$$

where in the second-last inequality we have used the fact that $c_q \ge c_{\text{eff}}(i,q)$ for any i,q.

We now need to prove condition 2 is satisfied. Call the parameter u *good* if $\alpha_u \ge \frac{1}{\varepsilon}$ and bad otherwise. Note that if u is good, then $\alpha_u \le (1+\varepsilon)\lfloor\alpha_u\rfloor$. For simplicity, let $D := \sum_{j\in C} d_j \sum_{i\in S_k,p\in[P]} \mathsf{x}_{ijp}$ denote the total fractional demand assigned to S_k. From the definition of $c_{\text{eff}}(\cdot)$, we get $c_{\text{eff}}(i,p)y_{ip} = \sum_{j\in C} d_j \mathsf{x}_{ijp}$ since for $j \notin H, i \in H$ we have $\mathsf{x}_{ijp} = 0$. Therefore,

$$\begin{aligned}
D := \sum_{j\in C} d_j \sum_{i\in S_k,p\in[P]} \mathsf{x}_{ijp} &= \sum_{i\in S_k,p\in[P]} c_{\text{eff}}(i,p)y_{i,p} \le \sum_{u\le U}\Delta^{u+1}\sum_{(i,p)\in A_u} y_{ip} && (8)\\
&\le (1/\varepsilon)\sum_{u\le U:\text{ bad}}\Delta^{u+1} + \sum_{u\le U:\text{ good}}\Delta^{u+1}\alpha_u && (9)\\
&\le (1/\varepsilon)\cdot\frac{\Delta^{U+2}}{\Delta-1} + (1+\varepsilon)\Delta\sum_{u:\text{good}}\Delta^u\lfloor\alpha_u\rfloor && (10)\\
&\le \frac{(1+\varepsilon)^2}{\varepsilon^2}\Delta^U + (1+\varepsilon)^2\sum_{i\in S,p\in[P]} c_p y^{\mathsf{int}}_{ip} && (11)\\
&\le \varepsilon(1+\varepsilon)^2 D + (1+\varepsilon)^2\sum_{i\in S,p\in[P]} c_p y^{\mathsf{int}}_{ip} && (12)
\end{aligned}$$

Therefore, we get $D \le \frac{(1+\varepsilon)^2}{1-\varepsilon(1+\varepsilon)^2}\sum_{i\in S,p\in[P]} y^{\mathsf{int}}_{ip} \le (1+100\varepsilon)\sum_{i\in S,p\in[P]} y^{\mathsf{int}}_{ip} \le (1+\delta)\sum_{i\in S,p\in[P]} y^{\mathsf{int}}_{ip}$. The second inequality uses ε is small enough. Therefore, S_k has the $\left(\tilde{O}(1/\delta),(1+\delta)\right)$-roundability property. □

Claim 7.2, Lemma 7.3, Claim 7.4, and Lemma 7.6 prove that the decomposition has the properties desired by Theorem 6.2.

8 Supply Polyhedra of $Q||C_{min}$: Proof of Theorem 5.2

Throughout the proof we fix $\mathcal{I}$ to be the instance of $Q||C_{min}$ and the supply vector $(s_1,\ldots,s_n)$. For simplicity of presentation, given the supply vector, abusing notation let J denote the multiset of jobs where job j appears s_j times. We know that the LP(A1)-(A3) is feasible with the s_j replaced by 1. We want to find an assignment where machine i gets at least $D_i/2$ capacity.

The algorithm is a very simple greedy algorithm which doesn't look at the LP solution , and the feasibility of LP(A1)-(A3) is only used for analysis. Rename the jobs (with multiplicities) in decreasing order of capacities $c_1 \ge c_2 \ge \cdots \ge c_N$, and rename the machines in decreasing order of D_i's, that is, $D_1 \ge D_2 \ge \ldots \ge D_m$. Starting with machine $i = 1$ and job $j = 1$, assign jobs j to i if the total capacity filled in machine i is $< D_i/2$ and move to the next job. Otherwise, call machine i happy and move to the next machine. Obviously, if all machines are happy at the end we have found our assignment.

The non-trivial part is to prove that if some machine is unhappy, then the LP(A1)-(A3) is infeasible (with s_j replaced by 1). To do so, we take the Farkas dual of the LP; the following LP is feasible iff LP(A1)-(A3) is infeasible. We describe a feasible solution to the system below if we obtain some unhappy agent.

$$\sum_{i=1}^{m} \beta_i D_i > \sum_{j=1}^{n} \alpha_j \tag{F1}$$

$$\forall i \in M, j \in J \quad \beta_i \min(c_j, D_i) \leq \alpha_j \tag{F2}$$

$$\forall i \in M, \quad \beta_i \geq 0 \tag{F3}$$

Suppose machine $i^\star$ is the first machine which is unhappy. Let $S_1, \ldots, S_{i^\star - 1}$ be the jobs assigned to machines 1 to $(i^\star - 1)$ and $S_{i^\star}$ be the remainder of jobs. We have $\sum_{j \in S_{i^\star}} c_j < D_{i^\star}/2$. We also have for all $1 \leq i \leq i^\star$, $\sum_{j \in S_i} \min(c_j, D_i) \leq D_i$. If not, then the machine must receive at least two jobs and would have capacity $> D_i/2$ from all but the last. We now describe a feasible solution to (F1)-(F3).

Given the assignment S_i's, call a machine i *overloaded* if S_i contains a single jobs j_i with $c_{j_i} \geq D_i$. We let $\beta_1 = 1$. For $1 \leq i < i^\star$, we have the following three-pronged rule

- If $i+1$ is not overloaded, $\beta_{i+1} = \beta_i$.
- If $i+1$ is overloaded, and so is i, then $\beta_{i+1} = \beta_i \cdot D_i / D_{i+1}$.
- If $i+1$ is overloaded but i is not, then $\beta_{i+1} = \beta_i \cdot c_{j_{i+1}} / D_{i+1}$, where j_{i+1} is the job assigned to $i+1$.

For any job j assigned to machine i, we set $\alpha_j = \beta_i \min(c_j, D_i)$. Since for any S_i, we have $\sum_{j \in S_i} \min(c_j, D_i) \leq D_i$ and $\sum_{j \in S_{i^\star}} c_j < D_{i^\star}/2$, the given (α, β) solution satisfies (F1). We now prove that it satisfies (F2). From the construction of the β's the following claims follow.

Claim 8.1. $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_m$.

Claim 8.2. $\beta_1 D_1 \geq \beta_2 D_2 \geq \cdots \geq \beta_m D_m$.

Proof. The only non-obvious case is if $i+1$ is overloaded but i is not: in this case $\beta_{i+1} D_{i+1} = \beta_i c_{j_{i+1}}$. But since i is not overloaded, let j be some job assigned to i with $c_j \leq D_i$. By the greedy rule, $c_j \geq c_{j_{i+1}}$, and so $\beta_{i+1} D_{i+1} \leq \beta_i D_i$. □

Now fix a job j and let i be the machine it is assigned to. Note (F2) holds for (i, j) and we need to show (F2) holds for all (i', j) too. I don't see any more glamorous way than case analysis.

Case 1: $c_j \leq D_i$. In this case $\alpha_j = \beta_i c_j$ and i is not overloaded. Let $i' < i$. Then we have $\beta_{i'} \min(c_j, D_{i'}) \leq \beta_{i'} c_j \leq \beta_i c_j$, where the last inequality follows from Claim 8.1.

Now let $i' > i$. If $c_j \leq D_{i'}$, then none of the machines from i to i' can be overloaded. Therefore, $\beta_{i'} = \beta_i$, and so $\beta_{i'} c_j = \beta_i c_j = \alpha_j$. So, we may assume $c_j > D_{i'}$ and we need to upper bound $\beta_{i'} D_{i'}$. Let $i'' > i$ be the first machine which is overloaded with job j'' say. By Claim 8.2, we have $\beta_{i'} D_{i'} \leq \beta_{i''} D_{i''}$. Now note that $\beta_{i''} D_{i''} = \beta_{i''-1} c_{j''} = \beta_i c_{j''} \leq \beta_i c_j = \alpha_j$ where the second equality follows since none of the machines from i to $i'' - 1$ were overloaded.

Case 2: $c_j > D_i$. In this case $\alpha_j = \beta_i D_i$ and i is overloaded. Let $i' > i$. Then, $\beta_{i'} \min(c_j, D_{i'}) = \beta_{i'} D_{i'} \leq \beta_i D_i$ where the last inequality follows from Claim 8.2.

Let $i' < i$. Let $i' \leq i'' < i$ be the smallest entry such that $c_j > D_{i''}$. Note that all machines from i'' to i must be overloaded implying $\beta_{i''} D_{i''} = \beta_i D_i$. Since $c_j \leq D_{i'}$ (in case $i' < i''$), we need to upper bound $\beta_{i'} c_j$. By Claim 8.1, $\beta_{i'} c_j \leq \beta_{i''-1} c_j$. Now, if $i'' - 1$ were overloaded, then $\beta_{i''} D_{i''} = \beta_{i''-1} D_{i''-1} \geq \beta_{i''-1} c_j$ where the last inequality follows from definition of i''. Together, we get $\beta_{i'} c_j \leq \beta_i D_i$.

Lemma 8.3. *$\mathcal{P}_{\mathsf{ass}}$ is upward-feasible.*

Proof. Let $s := (s_1, \ldots, s_n) \in \mathcal{P}_{\mathsf{ass}}$ for a certain instance of $Q||C_{min}$ where the jobs have been renamed so that $c_1 \leq \cdots \leq c_n$. We need to prove any non-negative vector $t := (t_1, \ldots, t_n)$ s.t. $t \succeq_{\mathsf{suff}} s$also lies in $\mathcal{P}_{\mathsf{ass}}$. By the "hybridization argument", it suffices to prove the lemma for s and t differing only in coordinates $\{j-1, j\}$ and $t_j \geq s_j$ and $t_{j-1} \geq \max(0, s_{j-1} + (s_j - t_j))$. Given that, we can move from s to t by changing pairs of coordinates each time maintaining feasibility in $\mathcal{P}_{\mathsf{ass}}$.

Let z be the solution for the supply vector s; we construct a solution $\bar{z}$ for the supply vector t starting with $\bar{z} = z$. If $\bar{z}$ is not already feasible, then it must be because $s_{j-1} \geq \sum_{i \in M} \bar{z}_{i,j-1} > t_{j-1}$. We select an arbitrary $i \in M$ with $\bar{z}_{i,j-1} > 0$ and increase $\bar{z}_{ij}$ and decrease $\bar{z}_{i,j-1}$ by δ. Since $c_j \geq c_{j-1}$, (A2) remains valid. Since the total increase of fractional load of job j is exactly the same as the decrease in that of job $j-1$, and we only need total decrease $(s_{j-1} - t_{j-1}) \leq t_j - s_j$, at the end we get that $\bar{z}$ is feasible wrt supply vector t.

□

9 Supply Polyhedra for $Q|f_i|C_{min}$: Proof of Theorem 5.4

Throughout the proof we fix $\mathcal{I}$ to be the instance of $Q|f_i|C_{min}$ and the supply vector $(s_1, \ldots, s_n)$. Let z be a feasible solution to (C1)-(C3). The proof of Theorem 5.4 follows from Lemma 9.1, Lemma 9.8, and Lemma 9.9

Lemma 9.1. *Given z, we can find an of assignment of the s_j jobs of capacity c_j to the machines such that for all $i \in M$ receives a total capapcity $\geq D_i/\alpha$ for $\alpha = O(\log D)$ where $D = D_{\max}/D_{\min}$.*

Proof. We start by classifying the demands into buckets.

Bucketing Demands. We partition the demands into buckets depending on their requirement values D_i. By scaling data, we may assume without loss of generality that $D_{\min} = 1$. We say that demand i belongs to *bucket* t if $2^{t-1} \leq D_i < 2^t$. We let $B^{(t)}$ to denote the bucket t. The number of buckets $K \leq \log_2 D$. For any bucket t, we round-down all the demands for $i \in B^{(t)}$; define $\bar{D}_i = 2^{t-1}$ for all $i \in B^{(t)}$. Note that any ρ-approximate feasible solution with respect to $\bar{D}$'s is 2ρ-approximate with respect to the original D_i's.

To this end, we modify the feasible solution z to a solution $\bar{z}$ in various stages. Initially $\bar{z} \equiv z$. Our modified solution $\bar{z}$'s support will not be $\mathcal{F}_i$; to this end we define $\mathcal{F}_i^{(\alpha,\beta)}$ for parameters $\alpha, \beta \geq 1$.

Definition 11. *For machine i and parameters $\alpha, \beta > 1$, $\mathcal{F}_i^{(\alpha,\beta)}$ contains the set S if either (a) $S = \{j\}$ is a singleton with $c_j \geq \frac{\bar{D}_i}{3\log_2 D}$, or (b) $|S| \leq f_i$, $c_j \leq \alpha \cdot \frac{\bar{D}_i}{3\log_2 D}$, and $\sum_{j \in S} c_j \geq \frac{\bar{D}_i}{\beta}$. We say $\bar{z}$ is (α,β)-feasible if for all i, $\bar{z}(i,S) > 0$ implies $S \in \mathcal{F}_i^{(\alpha,\beta)}$.*

Step 1: Partitioning Configurations.

We call a job of capacity c_j *large* for machine i if $c_j \geq \frac{\bar{D}_i}{3\log_2 D}$, otherwise we call it *small* for machine i. For every machine i, if $z(i,S) > 0$ and S contains any large job j for i, then we replace S by $\{j\}$. To be precise, we set $\bar{z}(i,\{j\}) = z(i,S)$ and $\bar{z}(i,S) = 0$. We call such singleton configurations *large* for i; all others are *small*. Let $\mathcal{F}_i^L$ be the collection of large configurations for i; the rest $\mathcal{F}_i^S$ being small configurations. Define $\bar{z}^L(i) := \sum_{S \in \mathcal{F}_i^L} \bar{z}(i,S)$ be the total large contribution to i, and let $\bar{z}^S(i) := \sum_{S \in \mathcal{F}_i^S} \bar{z}(i,S)$ the small contribution.

Claim 9.2. *After* **Step 1**, *$\bar{z}$ satisfies (C1) and (C2) and $\bar{z}$ is $(1,1)$-feasible.*

We partition the demands into buckets depending on their requirement values D_i. By scaling data, we may assume without loss of generality that $D_{min} = 1$. We say that demand i belongs to *bucket* t if $2^{t-1} \leq D_i < 2^t$. We let $B^{(t)}$ to denote the bucket t. The number of buckets $K \leq \log_2 D$.

A machine i is called *rounded* if $\bar{z}(i,S) = 1$ for some set S. We let $\mathcal{R}$ denote the rounded machines. The remaining machines are of three kinds: *large* ones with $\bar{z}^L(i) = 1$, *hybrid* ones with $\bar{z}^L(i) \in (0,1)$ and *small* ones with $\bar{z}^L(i) = 0$. Let $\mathcal{L}, \mathcal{H}, \mathcal{S}$ denote these respectively.

Step 2: Taking care of large machines.

The goal of this step is to modify $\bar{z}$ such that (a) the set of large machines becomes empty and (b) the set of hybrid machines is bounded. In particular, we will have at most one hybrid machine in a bucket proving there are at most K hybrid machines. First we need to discuss two sub-routines.

Subroutine: FixLargeMachine(i). This takes input a large machine $i \in \mathcal{L}$, that is, $\bar{z}^L(i) = 1$. We modify $\bar{z}$ such that at the end of the subroutine, among other things, i gets rounded and enters $\mathcal{R}$.

Consider the jobs j large for i such that $\bar{z}(i, \{j\}) \in (0, 1)$. Since $\bar{z}^L(i) = 1$ and $i \notin \mathcal{R}$, there exists at least two such jobs. Let j_1 be the smallest capacity among these, and j_2 be any other such job. Two cases arise. In the simple case, there exists no $i' \notin \mathcal{R}, S' \subseteq \mathsf{Supp}$ with $\bar{z}(i', S) > 0$ and $j_1 \in S$. That is, no other machine fractionally claims the job j_1. Since s_{j_1} is an integer, we have slack in (C2). We round up $\bar{z}(i, \{j_1\}) = 1$, set $\bar{z}(i, T) = 0$ for all other configurations of i, and add i to $\mathcal{R}$ and terminate.

Otherwise, there exists a machine i' and a set S such that $z(i', S) \in (0, 1)$ and $j_1 \in S$. Now define the set T as follows. If $c_{j_2} > \frac{\bar{D}_{i'}}{3 \log_2 D}$, then $T = \{j_2\}$; otherwise $T = S - j_1 + j_2$. Note that in the second case j_2 could already be in S; T then contains one more copy, that is, $n(T, j_2) = n(S, j_2) + 1$. We modify $\bar{z}$-as follows. We decrease $\bar{z}(i, \{j_2\})$ and $\bar{z}(i', S)$ by δ, and increase $\bar{z}(i, \{j_1\})$ and $\bar{z}(i', T)$ by δ till one of the values becomes 0 or 1. If at any point, some configuration gets $\bar{z}$ value 1, we add the corresponding machine to $\mathcal{R}$. We proceed till i enters $\mathcal{R}$.

Claim 9.3. FixLargeMachine*(i) terminates. Upon termination, the solution $\bar{z}$ satisfies* (C1) *and* (C2)*, and if $\bar{z}$ was (α, β)-feasible before the subroutine, it remains (α, β)-feasible afterwards.*

Proof. If at any point we are in the simpler case, then i enters $\mathcal{R}$ and we terminate. Since we modify $\bar{z}(i, S)$ only for machine i, (C1) is satisfied by the modification. (C2) is satisfied for j no other machine fractionally claims it. In the other case, note that the modification by δ's preserve the LHS of (C1). Furthermore, since $T \subseteq S \cup j_2$, it can only decrease the LHS of (C2) (for jobs $j' \in S \setminus T \cup j_1$ when $T = \{j_2\}$). Finally, the new entry to the support of $\bar{z}$ is $\bar{z}(i', T)$ and we need to check $T \in \mathcal{F}_i^{(\alpha,\beta)}$. If T is a singleton (that is j_2), then $c_{j_2} \geq \frac{\bar{D}_{i'}}{3 \log D}$ and so $T \in \mathcal{F}_i^{(\alpha,\beta)}$. Otherwise, since $S \in \mathcal{F}_i^{(\alpha,\beta)}$, $c_{j_2} < \frac{\bar{D}_{i'}}{3 \log D}$, and $c_{j_2} \geq c_{j_1}$ we have $T \in \mathcal{F}_i^{(\alpha,\beta)}$. So at every step $\bar{z}$ maintains (C1) and (C2) and is (α, β)-feasible. To argue termination, note that in the second case the value of $\bar{z}(i, \{j_1\})$ strictly goes up. In the end, we must have $\bar{z}(i, \{j_1\}) = 1$. □

Subroutine: FixBucket(t). This takes input a bucket t with more than one hybrid machine, and modifies the $\bar{z}$-solution such that there is at most one hybrid machine in t. Recall a machine is hybrid if $\bar{z}^L(i) \in (0, 1)$. The $\bar{z}$-value for other machines in other buckets are unaffected.

Among the hybrid machines in $B^{(t)}$, let i be the one with the smallest f_i. Let i' be any other hybrid machine in this bucket. We know there is at least one more. We now *modify* $\bar{z}$ as follows. Since $\bar{z}^L(i') > 0$, there exists a large configuration $\{j'\}$ for i' with $\bar{z}(i', \{j'\}) > 0$. Similarly, since $\bar{z}^L(i) < 1$, there must exist a *small* configuration T with $\bar{z}(i, T) > 0$. We then perform the following change: decrease $\bar{z}(i', \{j'\})$ and $\bar{z}(i, T)$ by δ, and increase $\bar{z}(i, \{j'\})$ and $\bar{z}(i', T)$ by δ, for a $\delta > 0$ such that one of the variables becomes 0 or 1. Note that this keeps (C1) and (C2) maintained.

We keep performing the above step till bucket t contains at most one hybrid machine. If at any point, some configuration gets $\bar{z}$ value 1, we add the corresponding machine to $\mathcal{R}$.

Claim 9.4. FixBucket*(t) terminates. Upon termination, the solution $\bar{z}$ satisfies* (C1) *and* (C2)*, and if $\bar{z}$ was (α, β)-feasible before the subroutine, it remains (α, β)-feasible afterwards.*

Proof. The possibly new entry to the support of $\bar{z}$ is $\bar{z}(i', T)$. Note that $|T| \leq f_i$ since $\bar{z}$ was (α, β)-feasible to begin with, and therefore $|T| \leq f_{i'}$ as well. The other conditions of (α, β)-feasibility are satisfied since $\bar{D}_i = \bar{D}_{i'}$, both being in the same bucket. Also note that the LHS of both (C1) and (C2) remain unchanged. To argue termination, till bucket t contains more than one hybrid machine, note that $\bar{z}^L(i)$ increases for the hybrid machine i with the smallest f_i. □

Now we have the two subroutines to describe **Step 2** of the algorithm. It is the following while loop.

While $\mathcal{L}$ is non-empty:

- If $i \in \mathcal{L}$, then FixLargeMachine(i). Note that i enters $\mathcal{R}$ after this. This can increase the number of hybrid machines across buckets.

– For all $1 \le t \le K$, if $B^{(t)}$ contains more than one hybrid machine, then FixBucket(t). This can increase the number of machines in $\mathcal{L}$.

Since the FixLargeMachine adds a new machine to $\mathcal{R}$, it cannot run more than m times. Therefore, the while loop terminates. Furthermore, before the loop $\bar{z}$ is $(1,1)$-feasible satisfying (C1) and (C2) (Claim 9.2), therefore Claim 9.3 and Claim 9.4 imply that it satisfies after the while loop. We encapsulate the above discussion in the following claim about Step 2.

Claim 9.5. Step 2 *terminates. Upon termination, the modified LP solution $\bar{z}$ is $(1,1)$-feasible, satisfies (C1) and (C2), and furthermore $\mathcal{L}$ is empty and for every bucket t we have at most one hybrid machine $i \in B^{(t)} \setminus \mathcal{R}$.*

Step 3: Taking care of hybrid machines.

Let $\mathcal{H}$ be the set of hybrid machines at this point. We know that $|\mathcal{H}| \le K \le \log_2 D$ since each bucket has at most one hybrid machine. For any machine $i \in \mathcal{H}$ with $\bar{z}^L(i) \le 1 - 1/K$, we zero-out all its large contribution. More precisely, for all j large for i we set $\bar{z}(i,\{j\}) = 0$. Note that (C1) no longer holds, but it holds with RHS $\ge 1/K$. Note that these machines now leave $\mathcal{H}$ and enter $\mathcal{S}$.

At this point, for every $i \in \mathcal{H}$ has $z^L(i) > 1 - 1/K$. Let $K' := |\mathcal{H}|$. Let J' be the set of jobs j which are large for some machine $i \in \mathcal{H}$ and $\bar{z}(i,\{j\}) > 0$. Let $s'_j := s_j - \sum_{i\in\mathcal{R}} \sum_S \bar{z}(i,S) n(S,j)$ be the remaining copies of j. Note that it is an integer since s_j was an integer and for all $i \in \mathcal{R}$, $\bar{z}(i,S) \in \{0,1\}$. Let G be a bipartite graph with $\mathcal{H}$ on one side and J' on the other with s'_j copies of job j. We draw an edge (i,j) iff j is large for i with $\bar{z}(i,\{j\}) > 0$.

Claim 9.6. *There is a matching in G matching all $i \in \mathcal{H}$.*

Proof. Pick a subset $\mathcal{H}' \subseteq \mathcal{H}$ and let J'' be its neighborhood in G. We need to show $\sum_{j\in J''} s'_j \ge \mathcal{H}'$. Since z satisfies (C2), we get

$$\sum_{j\in J''} s'_j \ge \sum_{j\in J''} \sum_{i\in\mathcal{H}'} z(i,\{j\}) = \sum_{i\in\mathcal{H}'} \sum_{j\in J''} z(i,\{j\}) > (1-1/K)|\mathcal{H}'| \ge |\mathcal{H}'| - 1$$

The first inequality follows since $\bar{z}$ satisfies (C2). The strict inequality follows since J'' is the neighborhood of $\mathcal{H}'$ and the fact that $z^L(i) > 1 - 1/K$ for all $i \in \mathcal{H}$. The claim follows since s'_j's are integers. □

If machine $i \in \mathcal{H}$ is matched to job j, then we assign i a copy of this job, that is, set $\bar{z}(i,\{j\}) = 1$ and $\bar{z}(i,S) = 0$ for all other S, and add i to $\mathcal{R}$. Let $J_M \subseteq J'$ be the sub(multi)set of jobs allocated; note $|J_M| \le K \le \log_2 D$. After this point all machines outside $\mathcal{R}$ are small. For every $i \in \mathcal{S}$ and every small configuration S with $\bar{z}(i,S) > 0$, we move this mass to $\bar{z}(i, S \setminus J_M)$. More precisely, $\bar{z}(i, S\setminus J_M) = \bar{z}(i,S)$ and $\bar{z}(i,S) = 0$ for all i and S. Note that (C2) is satisfied at this point. Furthermore, since $\bar{z}$ was $(1,1)$-feasible, we know that $\sum_{j\in S} c_j \ge \bar{D}_i$ and for every $j \in S \cap J_M$ we have $c_j \le \frac{\bar{D}_i}{3\log_2 D}$.

$$\sum_{j\in S\setminus J_M} c_j \ge \sum_{j\in S} c_j - |J_M| \cdot \frac{\bar{D}_i}{3\log_2 D} \ge \frac{2\bar{D}_i}{3}$$

Therefore, we have proved the following claim.

Claim 9.7. *At the end of* **Step 3**, *we have a solution $\bar{z}$ with (a) $\bar{z}^L(i) = 0$ for all $i \notin \mathcal{R}$, (b) $\bar{z}$ is $(1,3/2)$-feasible, (c) $\bar{z}$ satisfies (C2), and satisfies (C1) replaced by $\frac{1}{K} \le \sum_S \bar{z}(i,S) \le 1$.*

Step 4: Taking care of Small Machines. We now convert the solution $\bar{z}$ to a solution z of the assignment LP in the following standard way. As before, let $s'_j = s_j - \sum_{i\in\mathcal{R}} \sum_S \bar{z}(i,S) n(S,j)$ be the

number of jobs remaining. For every $i \notin \mathcal{R}$ and $j \in J$ define $\mathsf{z}_{ij} = \sum_S \bar{z}(i,S)n(S,j)$. Note that this satisfies the constraint of the assignment LP:

$$\forall j \in J, \quad \textstyle\sum_{i \in \mathcal{S}} \mathsf{z}_{ij} \le s'_j \tag{13}$$

$$\forall i \in \mathcal{S}, \quad \textstyle\sum_{j \in J} \mathsf{z}_{ij} c_j \ge \frac{2\bar{D}_i}{3\log_2 D} \tag{14}$$

$$\forall i \in \mathcal{S}, \quad \textstyle\sum_{j \in J} \mathsf{z}_{ij} \le f_i \tag{15}$$

$$\forall i \in \mathcal{S}, j \in J \text{ with } c_j \ge \tfrac{\bar{D}_i}{3\log_2 D}, \quad \mathsf{z}_{ij} = 0 \tag{16}$$

The last equality follows since $\bar{z}$ was $(1, 8/15)$-feasible and so $\bar{z}(i,S) = 0$ for any set S containing a job j with $c_j \ge \frac{\bar{D}_i}{3\log_2 D}$. The first inequality follows since $\bar{z}$ satisfies (C2). To see the second and third point, note that for any $i \in \mathcal{S}$,

$$\sum_{j \in J} \mathsf{z}_{ij} c_j = \sum_j \sum_S \bar{z}(i,S)n(S,j)c_j = \sum_S \bar{z}(i,S) \sum_j n(S,j)c_j \ge \frac{1}{\log_2 D} \cdot \frac{2\bar{D}_i}{3}$$

since $\sum_S \bar{z}(i,S) \ge 1/K$ for all $i \in \mathcal{S}$ and since $\bar{z}$ is $(1, 3/2)$-feasible, we have $\sum_{j=1}^n n(S,j)c_j \ge \frac{2\bar{D}_i}{3}$. Similarly,

$$\sum_{j \in J} \mathsf{z}_{ij} = \sum_j \sum_S \bar{z}(i,S)n(S,j) = \sum_S \bar{z}(i,S) \sum_j n(S,j) \le f_i$$

since for any S, $\sum_{j \in S} n(S,j) \le f_i$ and $\sum_S \bar{z}(i,S) \le 1$. Now we use Theorem 5.3 to find an integral allocation z^{int} of the jobs J to machines in $\mathcal{S}$ satisfying (13),(15), and $\sum_{j \in J} z^{\mathsf{int}}_{ij} c_j \ge \frac{\bar{D}_i}{3\log_2 D}$.

The final integral assignment is as follows. For every $i \in \mathcal{R}$, we assign the configuration S with $\bar{z}(i,S) = 1$. Since $\bar{z}$ is $(1, 3/2)$-feasible, every such machine i gets a total capacity of at least $\frac{\bar{D}_i}{3\log_2 D}$. All the remaining machines $i \in \mathcal{S}$ obtain a set of jobs giving them capacity $\ge \frac{\bar{D}_i}{3\log_2 D}$. This completes the proof of Lemma 9.1. □

Lemma 9.8. *$\mathcal{P}_{\mathsf{conf}}$ is upward-feasible.*

Proof. Let $s := (s_1, \ldots, s_n) \in \mathcal{P}_{\mathsf{conf}}$ for a certain instance of $Q|f_i|C_{min}$ where the jobs have been renamed so that $c_1 \le \cdots \le c_n$. We need to prove any non-negative vector $t := (t_1, \ldots, t_n)$ s.t. $t \succeq_{\mathsf{suff}} s$also lies in $\mathcal{P}_{\mathsf{conf}}$. By the "hybridization argument",it suffices to prove the lemma for s and t differing only in coordinates $\{j-1, j\}$ and $t_j \ge s_j$ and $t_{j-1} \ge \max(0, s_{j-1} + (s_j - t_j))$. Given that, we can move from s to t by changing pairs of coordinates each time maintaining feasibility in $\mathcal{P}_{\mathsf{conf}}$.

Let z be the solution for the supply vector s; we construct a solution $\bar{z}$ for the supply vector t starting with $\bar{z} = z$. If $\bar{z}$ is not already feasible, then it must be because $s_{j-1} \ge \sum_{i,S} z(i,S)n(S, j-1) > t_{j-1}$. Therefore, we need to decrease the fractional utilization of job $(j-1)$ by $s_{j-1} - t_{j-1} \le t_j - s_j$. For any machine i and any set $S \in \mathcal{F}_i$ with $z(i,S) > 0$ and $n(S, j-1) \ge 1$ (and this must exist since $t_{j-1} \ge 0$), define $T := S - \{j-1\} + \{j\}$. Note that T could already have a copy of job j; we have $n(T,j) = n(S,j) + 1$. Also note since $c_j \ge c_{j-1}$, if $S \in \mathcal{F}_i$ then so is $T \in \mathcal{F}_i$. We let $\bar{z}(i,S) = z(i,S) - \delta$ and $\bar{z}(i,T) = z(i,T) + \delta$ till either $\bar{z}(i,S) = 0$ or $\bar{z}(i,T) = 1$. Since the total increase of fractional load of job j is exactly the same as the decrease in that of job $j-1$, and we only need total decrease $(s_{j-1} - t_{j-1}) \le t_j - s_j$, at the end we get that $\bar{z}$ is feasible wrt supply vector t. □

Lemma 9.9. *$\mathcal{P}_{\mathsf{conf}}$ has an $(1+\varepsilon)$-approximate separation oracle.*

Proof. Fix $\varepsilon > 0$. Given a supply vector $s = (s_1, \ldots, s_n)$, we give a polynomial time algorithm which either returns a hyperplane separating s and $\mathcal{P}_{\mathsf{conf}}$, or we can assert that $s \in \mathcal{P}_{\mathsf{conf}}(\mathcal{I}')$, where $\mathcal{I}'$ is an instance where machine i has demand $D_i/(1+\varepsilon)$. To this end, for every machine i, define $\mathcal{F}_i^{(\varepsilon)} := \{S : |S| \le f_i, \sum_j c_j n(S,j) \ge D_i/(1+\varepsilon)\}$. To prove $s \in \mathcal{P}_{\mathsf{conf}}(\mathcal{I}')$, we need to find $z(i,S)$ defined for all $i \in M, S \in \mathcal{F}_i^{(\varepsilon)}$

satisfying (C1)-(C2). For every $j \in J$, define $\tilde{c}_j := (1+\varepsilon)c_j$. Note for every $S \in \mathcal{F}_i^{(\varepsilon)}$ iff $|S| \le f_i$ and $\sum_{j\in S} \tilde{c}_j n(S,j) \ge D_i$.

Consider the following system of inequalities.

$$\forall j \in J, \quad \alpha_j \ge 0 \tag{D1}$$

$$\sum_{j\in J} s_j \cdot \alpha_j < \sum_{i\in M} \beta_i \tag{D2}$$

$$\forall i \in M, S \in \mathcal{F}_i, \quad \sum_{j\in J} \alpha_j n(S,j) \ge \beta_i \tag{D3}$$

We also need a stronger set of inequalities.

$$\forall i \in M, S \in \mathcal{F}_i^{(\varepsilon)}, \quad \sum_{j\in J} \alpha_j n(S,j) \ge \beta_i \tag{D4}$$

If there exists a feasible solution (α,β) to (D1)-(D3), then this forms the hyperplane separating s and $\mathcal{P}_{\mathsf{conf}}$ as follows. This is because for all $t \in \mathcal{P}_{\mathsf{conf}}$, if $z(i,S)$ is the solution feasible for $\mathcal{P}_{\mathsf{conf}}$ with t_j's in the RHS of (C2), then $\sum_{i\in M}\beta_i = \sum_{i\in M}\sum_{S\in\mathcal{F}_i}\beta_i z(i,S) \le \sum_{i\in M, S\in\mathcal{F}_i} z(i,S)\sum_{j\in J}\alpha_j n(S,j) \le \sum_{j\in J}\alpha_j t_j$. The following claim proves the lemma.

Claim 9.10. *In polynomial time, we can either find (α,β) feasible for (D1)-(D3), or we can find variables $z(i,S)$ for $i \in M, S \in \mathcal{F}_i^{(\varepsilon)}$ satisfying (C1)-(C2).*

Proof. We run the ellipsoid algorithm to check feasibility of the stronger system (D1),(D2), and (D4). At any point, we have a running iterate (α,β). For every $i \in M$, maximize $\sum_j \tilde{c}_j n(S,j)$ over all subsets S with $|S| \le f_i$ and $\sum_{j\in J}\alpha_j n(S,j) < \beta_i$. There is an FPTAS for this problem [9]. If the maximum value returned by the approximation scheme is *smaller* than D_i, then we know that the true optimum is $\le D_i(1+\varepsilon)$. That is, for every S with $|S| \le f_i$ and $\sum_{j\in J}\alpha_j n(S,j) < \beta_i$, we have $\sum_{j\in J}\tilde{c}_j n(S,j) \le D_i(1+\varepsilon)$. Which in turn implies $\sum_{j\in J} c_j n(S,j) \le D_i$. Contrapositively, for every $S \in \mathcal{F}_i$, we must have $\sum_{j\in J}\alpha_j n(S,j) \ge \beta_i$. That is (α,β) satisfies (D1)-(D3) and we exit.

Otherwise, the PTAS returns a set $S^\star$ with $|S^\star| \le f_i$ and $\sum_{j\in J}\tilde{c}_j n(S,j) \ge D_i$, that is $S^\star \in \mathcal{F}_i^{(\varepsilon)}$, for which $\sum_{j\in J}\alpha_j n(S^\star,j) < \beta_i$. We add $(i,S^\star)$ to $\mathcal{C}$, and return (α,β) to the separation oracle for (D4). The ellipsoid algorithm states than in polynomial time we either find an (α,β) feasible for (D1)-(D3), or the polynomially many hyperplanes in $\mathcal{C}$ prove (D1),(D2), and (D4) is infeasible. More precisely, there exists a solution z satisfying (C1)-(C2) with $z(i,S)$ defined for $(i,S) \in \mathcal{C}$. Since $|\mathcal{C}|$ is bounded by a polynomial, we can explicitly find z by solving the LP (C1)-(C2) with variables $z(i,S)$ for $(i,S) \in \mathcal{C}$. □

□

9.1 Integrality Gap

In this section we prove Theorem 5.5. Fix K. We present an instance $\mathcal{I}_K$ for which configuration LP is feasible but any integral allocation must violate the demand of some machine by factor K.

First we describe the machines in $\mathcal{I}_K$.

1. There is 1 machine M_0 with $D(M_0) = 1$ and $f(M_0) = 1$.
2. There are K machines $M_1,\ldots,M_K$ with $D(M_i) = K^{-i}$ and $f_i := f(M_i) = K^{2K+1}\cdot K^{-2i}$.
3. There are K **classes** of machines $\mathcal{M}_1,\mathcal{M}_2,\ldots,\mathcal{M}_K$. Machines in the same class are equivalent. There are f_i machines in $\mathcal{M}_i$ and they are numbered $N_1^{(i)},\ldots,N_{f_i}^{(i)}$. Each machine N in class i has $D(N) = \frac{1}{f_i K^i} = K^{-(2K+1)}\cdot K^i$ and $f(N) = 1$.

Now we describe the jobs.

1. There are K "big jobs" $J_1,\ldots,J_K$ with $c(J_i) = 1$.

2. There are K other types of jobs of the same capacity. Job J of type i has capacity $c(J) = c_i := \frac{1}{f_i K^i} = K^{-(2K+1)}K^i$ and there are $n_i := f_i(1+1/K) = (K+1)K^{2K}K^{-2i}$ of them. We divide these n_i jobs into two sets $S_i \cup T_i$ where $|S_i| = f_i$ and $|T_i| = f_i/K$. We order the jobs in S_i arbitrarily and call them $P_1^{(i)}, \cdots, P_{f_i}^{(i)}$.

So, the total number of machines in $\mathcal{I}_K$ are $1 + K + \sum_{i=1}^{K} f_i \leq K^{2K}$ and the number of jobs is $K + (1 + 1/K)\sum_{i=1}^{K} f_i \approx K^{2K}$.

Lemma 9.11. *The Configuration LP is feasible.*

Proof. We describe a fractional solution.

1. For machine M_0 we satisfy as follows: set $y(M_0, J_i) = 1/K$ for $i = 1, \ldots, K$. Note $c(J_i) \geq D(M_0)$ and $|J_i| = 1 = f(M_0)$.
2. For machine M_i we satisfy as follows: set $y(M_i, J_i) = 1 - 1/K$ and $y(M_i, S_i) = 1/K$. Recall S_i are the f_i jobs of type i.
   - Note $c(J_i) = 1 \geq D(M_i) = K^{-i}$ and $|J_i| = 1 \leq f(M_i) = K^{2K+1}K^{-2i}$ since $i \leq K$.
   - Note $c(S_i) = |S_i| \cdot \frac{1}{f_i K^i} = \frac{1}{K^i} = D(M_i)$. Note $|S_i| = f_i = f(M_i)$.
3. For $1 \leq i \leq K$, for a machine $N_j^{(i)}$ in class i, where $1 \leq j \leq f_i$, we satisfy it as follows: $y(N_j^{(i)}, P_j^{(i)}) = 1 - 1/K$ and $y(N_j^{(i)}, t) = 1/f_i$ for all $t \in T_i$. Since $|T_i| = f_i/K$, the total fractional y-amount that $N_j^{(i)}$ gets is 1. Also note that $N_j^{(i)}$ gets singleton jobs of type i whose capacity is $\frac{1}{f_i K_i} = D(N_j^{(i)})$.

We need to show that no job is over allocated.

1. The big jobs J_i is given $1/K$ to M_0 and $(1 - 1/K)$ to M_i.
2. For $1 \leq i \leq K$, $1 \leq j \leq f_i$, job $P_j^{(i)} \in S_i$ is given $1/K$ to M_i and $(1 - 1/K)$ to $N_j^{(i)} \in \mathcal{M}_i$.
3. For $1 \leq i \leq K$, job $t \in T_i$ is given $1/f_i$ to the f_i machines of $\mathcal{M}_i$.

This completes the description of the feasible solution. □

Lemma 9.12. *The integral optimum must violate some machine by factor $\Omega(K)$.*

Proof. Lets take machines in $\mathcal{M}_i$. Recall all machines here have demand of $\frac{1}{f_i K^i}$ and cardinality constraint of 1. Thus in the integral optimum, they **must** get one job which is either big, or of type i or larger. Now, the total number of jobs of type $j > i$ are

$$\sum_{j>i} f_j(1+1/K) = (K+1)K^{2K}\sum_{j>i} K^{-2j} \leq (K+1)K^{2K}K^{-2i}\sum_{\ell=1}^{\infty} K^{-2\ell} = O\left(f_i/K\right)$$

So, at least $(1 - \Theta(1/K))f_i$ of the machines in $\mathcal{M}_i$ get a job of type i (or a big job but lets assume for now this don't happen – can be ma). Therefore, the number of type i jobs left after satisfying machines $(M_0, \ldots, M_K)$ are only $\Theta(f_i/K)$.

Now take a machine M_i. We have $f(M_i) = f_i$ and $D(M_i) = 1/K^i$. First note that jobs of type $j < i$ are "useless" for M_i. Any f_i of them (best to take them of type $(i-1)$) gives capacity $f_i \cdot c_{i-1} = \frac{f_i}{f_{i-1}K^{i-1}} = \frac{1}{K^{i+1}} = \frac{1}{K} \cdot D(M_i)$. So any subset of these jobs that can fit in M_i gives capacity $\leq D(M_i)/K$. On the other hand, the total capacity of jobs remaining from type $j \geq i$ is $\sum_{j \geq i} \Theta(f_j/K) \cdot \frac{1}{f_j K^j} = \Theta(1/K)\sum_{j \geq i} \frac{1}{K^j} = \Theta(D(M_i)/K)$.

Therefore, any machine M_i can't get more than $D(M_i)/K$ from the "small" jobs. But then they all can't get big jobs. □

The above two lemmas prove Theorem 5.5 after noting that $K = \Theta(\log n / \log\log n) = \Theta(\log D / \log\log D)$ where n is either the number of machines of jobs and D is the ratio of $D_{\max}/D_{\min}$.

Theorem 9.13. *There cannot exist α-approximate supply polyhedra (or convex sets) for $\alpha < \frac{\log D}{\log\log D}$ for $Q|f_i|C_{min}$ instances.*

Proof. The proof follows from the instance constructed in the above Theorem 5.5. Indeed note that we can express the supply vector of the instance as $(1-p)\mathbf{s}_1 + p\mathbf{s}_2$. Here $\mathbf{s}_1$ denotes the following supply vector: there are $K+1$ big jobs with size 1, and there are f_i jobs of size c_i for all $1 \le i \le K$. Similarly $\mathbf{s_2}$ denotes the following supply vector: there is 1 big job with size 1, and there are $2f_i$ jobs of size c_i for all $1 \le i \le K$. Finally, the value p is set to $1 - 1/K$.

Now, we will show that for both $\mathbf{s}_1$ and $\mathbf{s}_2$ are feasible supply vectors. Indeed, for $\mathbf{s}_1$, we will assign the large jobs to the large machines $M_0, M_1, \ldots, M_K$. Then we will use the f_i jobs of size c_i to satisfy the f_i machines in class $\mathcal{M}_i$.

Likewise, for $\mathbf{s}_2$, we will assign the one large job to the large machine M_0. Then, for machine M_i, we will assign f_i items of size c_i. Finally, we will assign the remaining f_i jobs of size c_i to satisfy the f_i machines in class $\mathcal{M}_i$.

But now, note that for the resulting average supply vector, we proved in Theorem 5.5 that any assignment must violate some demand by a factor of $\Theta(\log D / \log\log D)$, thus proving the theorem. □

10 QPTAS for $Q|f_i|C_{min}$: Proof of Theorem 1.2

Let the instance $\mathcal{I}$ given to us have m machines with demands $D_1, \ldots, D_m$ and cardinality constraints $f_1, \ldots, f_m$, and n jobs with capacities $c_1, \ldots, c_n$ (with 1 copy of each job by taking possible duplicates). Let $D := D_{\max}/D_{\min}$, which we assume to be $\le \text{poly}(n)$. Fix $\varepsilon > 0$. Our goal is to either prove there is no feasible solution, or find an assignment giving each machine i a capacity of $\ge D_i(1 - O(\varepsilon))$. We start with a lemma which states that finding solutions satisfying cardinality constraints approximately suffices.

Lemma 10.1. *Given an assignment of jobs such that the load on any machine i is at least $D_i(1-\varepsilon_1)$ such that machine i gets $\le (1+\varepsilon_2)f_i$ jobs, we can find another assignment which satisfies the cardinality constraints and the load of any machine i is $\ge D_i(1-\varepsilon_3)$ for $\varepsilon_3 < 2\varepsilon_1 + \varepsilon_2$.*

Proof. For every machine i, let S_i be the jobs currently allocated to it. We may assume $\varepsilon_1 f_i \ge 1$, otherwise $|S_i| \le f_i$. Remove the $\lfloor 2\varepsilon_1 f_i \rfloor \ge \varepsilon_1 f_i$ least capacity jobs to obtain the set S_i'. Note that the total capacity of S_i' is at least $(1 - 2\varepsilon_1)$ times capacity of S_i, and therefore at least $(1 - 2\varepsilon_1 - \varepsilon_2)D_i$. □

Input Modification and Grouping. We now modify the data so that everything is rounded to the nearest power of $(1+\varepsilon)$. More precisely we round f_i to the *smallest* power of $(1+\varepsilon)$ larger than the original value and D_i to the largest power of $(1+\varepsilon)$ smaller than the original value. If the original instance had a feasible solution, then so does the modified instance. For technical reasons, we round c_p to the smallest value of the form $\varepsilon(1+\varepsilon)^t$ larger than the original value. Let J_p be the set of jobs with modified capacity $c_p = \varepsilon(1+\varepsilon)^p$, and let $n_p = |J_p|$. Furthermore, armed with Lemma 10.1, any $(1-\varepsilon)$-approximate solution to the new instance gives an $(1 - O(\varepsilon))$-approximate solution to the original instance.

We now divide the machines into groups. For $0 \le r \le O(\log n)$ and $0 \le s \le O(\log n)$, let $M^{(r,s)}$ be the number of machines with $D_i = (1+\varepsilon)^r$ and $f_i = (1+\varepsilon)^s$. Call a job p big for machine i if $c_p \ge \varepsilon D_i$. If $i \in M^{(r,s)}$, then p lies in the set $J_r \cup J_{r+1} \cup \cdots$. Otherwise, p is small for machine i. We define a bipartite graph H with jobs and machines on either side, with an edge (i,p) iff p is small for i.

For every $0 \le r, s \le O(\log n)$, we define a set of *feasible configurations* $\Phi^{(r,s)}$. These consist of vectors $\phi \in \mathbb{Z}^K_{\ge 0}$ for $K = O(1/\varepsilon)$ corresponding to big jobs assigned to machines i in $M^{(r,s)}$. To be precise, ϕ_k is supposed to count the number of jobs with $c_p = \varepsilon(1+\varepsilon)^{r+k}$ contained in the configuration ϕ. The last coordinate ϕ_K counts the number of jobs p with $c_p > (1+\varepsilon)^r$. Let $\mathsf{cap}(\phi) := \sum_{k=0}^{K} \phi_k \varepsilon(1+\varepsilon)^{r+k}$ be the

total load of the configuration and $|\phi| = \sum_{k=0}^{K} \phi_k$ be its cardinality. We let $\Phi^{(r,s)}$ be the collection of feasible minimal configurations, that is, ϕ's with (a) $|\phi| \le (1+\varepsilon)^s$ and (b) either $\mathsf{cap}(\phi) \le (1+\varepsilon)^r$ or $\mathsf{cap}(\phi) > (1+\varepsilon)^r$ and $\mathsf{cap}(\phi') \le (1+\varepsilon)^r$ for any ϕ' obtained by decreasing any positive coordinate of ϕ by exactly 1. Note that $|\Phi^{(r,s)}| \le N_0 = (1/\varepsilon)^{(1/\varepsilon)}$. Also note that in any optimal solution, each machine $i \in M^{(r,s)}$ does get one configuration from $\Phi^{(r,s)}$. Our algorithm constructs these classes and arbitrary numbers them. The tth member of $\Phi^{(r,s)}$ is denoted as $\phi^{(r,s,t)}$.

Enumeration. For every $0 \le r, s \le O(\log n)$ and $1 \le t \le N_0$, we *guess* the integer $\mathbf{b}_t^{(r,s)} \in \mathbb{Z}_{\ge 0}$ which indicates the number of machines in $M^{(r,s)}$ who are allocated the configuration $\phi^{(t)}$. These guesses must satisfy

$$\forall 0 \le r, s \le O(\log n), \qquad \sum_{t=1}^{N_0} \mathbf{b}_t^{(r,s)} = |M^{(r,s)}| \tag{17}$$

The number of such guesses is $\le \prod_{r,s} |M^{(r,s)}|^{N_0} \le C_\varepsilon^{O(\log^3 n)}$ for some constant C_ε which is double-exponential in $(1/\varepsilon)$. Since machines in $M^{(r,s)}$ are all equivalent (in terms of demand and cardinality constraint), by symmetry we can assign the $\mathbf{b}_t^{(r,s)}$ copies of $\phi^{(r,s,t)}$ as we like. For a guess to be feasible, for every job of type p, at most n_p copies must be used up in the guessed configurations. For every guess we get a residual problem on the bipartite graph H. Let n'_p be the remaining number of jobs of type p. Let D'_i be the residual demand of machine i, that is, $D_i - \mathsf{cap}(\phi)$ where ϕ is allocated to it by the guess. Let f'_i be the residual cardinality constraint, that is, $f'_i = f_i - |\phi|$.

Rounding. The remaining copies of jobs must satisfy the residual demand. For this we simply write the assignment LP(A1)-(A4) which we rewrite below.

$$\forall p, \qquad \textstyle\sum_{i \sim p} z_{ip} \le n'_p \tag{18}$$

$$\forall i \in [m], \quad \textstyle\sum_{p \sim i} c_p z_{ip} \ge D'_i \tag{19}$$

$$\forall i \in [m], \quad \textstyle\sum_{p \sim i} z_{ip} \le f'_i \tag{20}$$

where $i \sim p$ implies $c_p \le \varepsilon D_i$. If the residual LP has no solution, then our guess of big configurations is infeasible. We are also guaranteed some guess is correct and we get a feasible solution to above LP. Therefore, we apply Theorem 5.3 to get an integral solution z^{int}_{ip} satisfying (18),(20), and $\forall i \in [m]$, $\sum_{p \sim i} c_p z^{\mathsf{int}}_{ip} \ge D'_i - \varepsilon D_i$. Therefore in all every machine receives capacity $\ge D_i(1-\varepsilon)$. The total running time is dominated by the enumeration step. This proves **??**.

11 Integrality Gap for Non-Uniform Santa Claus Problem

We reproduce the integrality gap example for the configuration LP by Bansal and Sviridenko [7] for the general max-min allocation problem, and point out how their instance is in fact a $Q|restr|C_{min}$ instance. Fix integer K. There are K machines with demand $D_i = K$; these are the large machines $L = \{M_1, M_2, \ldots, M_K\}$. There are $K-1$ large jobs with $c_j = K$ which can only be assigned to the machines in L. Let J_B be the set of large jobs. There are K^2 small machines each with $D_i = 1$; these machines are distributed in K classes where the ith class $\mathcal{C}_i$ contains K small machines. We let $m_k^{(i)}$ denote the kth machine in $\mathcal{C}_i$, for $1 \le k \le K$. There are $K^2 + K$ small jobs with $c_j = 1$. These jobs are partitoned into K classes with ith class $\mathcal{J}_i$ containing $K+1$ small jobs. Each class $\mathcal{J}_i$ has one "public" job $j_0^{(i)}$ which can be assigned to any machine $m_k^{(i)} \in \mathcal{C}_i$ and K "private" jobs $j_k^{(i)}$, $1 \le k \le K$ where $j_k^{(i)}$ can be assigned to only $m_k^{(i)} \in \mathcal{C}_i$. Furthermore all the private jobs $j_k^{(i)} \in \mathcal{J}_i$ can be assigned to the large machine $M_i \in L$. This completes the description of the instance. Note that the number of machines and jobs are $\Theta(K^2)$.

The integral optimum solution must give one machine i capacity $\le D_i/K$. Indeed, at least one large machine M_i will not receive a job in J_B. The only other jobs available to M_i are the private jobs in $\mathcal{J}_i$.

Suppose we allocate two such jobs to M_i; wlog these are $j_1^{(i)}$ and $j_2^{(i)}$. Now note that the machines $m_1^{(i)}$ and $m_2^{(i)}$ have only job $j_0^{(i)}$ which can be assigned to them; and so one of them would get capacity 0. Therefore, the machine M_i can receive only one job $j_k^{(i)}$ giving it total capacity $\leq D_i/K$.

On the other hand the configuration LP is feasible. Every large machine M_i gets $z(M_i, j) = 1/K$ for all large jobs $j \in J_B$ and $z(M_i, \{j_1^{(i)}, \ldots, j_K^{(i)}\}) = 1/K$ for the set of private jobs in $\mathcal{J}_i$. For all $1 \leq i, k \leq K$, every machine $m_k^{(i)}$ receives $z(m_k^{(i)}, j_k^{(i)}) = 1 - 1/K$ and $z(m_k^{(i)}, j_0^{(i)}) = 1/K$. One can check all the jobs are fractionally assigned exactly.

12 Conclusion

In this paper we introduced and studied the Heterogeneous Cap-k-Center problem, and highlighted its connection to an interesting special case of the max-min allocation problems, namely $Q|f_i|C_{min}$. In our main result, we showed, using a decomposition theorem and the notion of supply polyhedra, a logarithmic approximation for $Q|f_i|C_{min}$, using which we showed a bicriteria $(O(1), O(\log n))$-approximation for Heterogeneous Cap-k-Center. We believe designing polynomial-time $O(1)$-approximations for $Q|f_i|C_{min}$ and bicriteria $(O(1), O(1))$ algorithms for Heterogeneous Cap-k-Center are very interesting open problems.

References

[1] H. An, A. Bhaskara, C. Chekuri, S. Gupta, V. Madan, and O. Svensson. Centrality of trees for capacitated k-center. In *IPCO 2014, Bonn, Germany, June 23-25, 2014. Proceedings*, pages 52–63, 2014. 2

[2] H. An, M. Singh, and O. Svensson. Lp-based algorithms for capacitated facility location. In *55th IEEE FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 256–265, 2014. 5

[3] A. Asadpour, U. Feige, and A. Saberi. Santa claus meets hypergraph matchings. *ACM Trans. Algorithms*, 8(3):24, 2012. 4

[4] Y. Azar and L. Epstein. Approximation schemes for covering and scheduling on related machines. In *APPROX'98, Aalborg, Denmark, July 18-19, 1998, Proceedings*, pages 39–47, 1998. 3, 9

[5] L. Babel, H. Kellerer, and V. Kotov. The$k$-partitioning problem. *Math. Meth. of OR*, 47(1):59–82, 1998. 5

[6] M. Bansal, N. Garg, and N. Gupta. A 5-approximation for capacitated facility location. In *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, pages 133–144, 2012. 5

[7] N. Bansal and M. Sviridenko. The santa claus problem. In *Proceedings of the 38th Annual ACM STOC, Seattle, WA, USA, May 21-23, 2006*, pages 31–40, 2006. 2, 4, 28

[8] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15(3):385–415, 1993. 1, 2

[9] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123(2):333–345, 2000. 25

[10] R. D. Carr, L. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the Eleventh Annual ACM-SIAM SODA, January 9-11, 2000, San Francisco, CA, USA.*, pages 106–115, 2000. 5, 12

[11] D. Chakrabarty, C. Chekuri, S. Khanna, and N. Korula. Approximability of capacitated network design. In *IPCO 2011, New York, NY, USA, June 15-17, 2011. Proceedings*, pages 78–91, 2011. 5, 12

[12] D. Chakrabarty, J. Chuzhoy, and S. Khanna. On allocating goods to maximize fairness. In *50th Annual IEEE FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 107–116, 2009. 4

[13] D. Chakrabarty, P. Goyal, and R. Krishnaswamy. The non-uniform k-center problem. In *43rd ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 67:1–67:15, 2016. 6

[14] M. Cygan, M. Hajiaghayi, and S. Khuller. LP rounding for k-centers with non-uniform hard capacities. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 273–282, 2012. 2

[15] H. G. Demirci and S. Li. Constant approximation for capacitated k-median with (1+epsilon)-capacity violation. In *43rd ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016. 5

[16] U. Feige. On allocations that maximize fairness. In *Proceedings of the Nineteenth Annual ACM-SIAM SODA 2008*, pages 287–293, 2008. 4

[17] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996. 4, 8

[18] I. L. Gørtz, M. Molinaro, V. Nagarajan, and R. Ravi. Capacitated vehicle routing with nonuniform speeds. *Math. Oper. Res.*, 41(1):318–331, 2016. 2

[19] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. *Inf. Syst.*, 26(1):35–58, 2001. 2

[20] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10(2):180–184, 1985. 2

[21] S. Im and B. Moseley. Scheduling in bandwidth constrained tree networks. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 171–180, 2015. 2

[22] H. Kellerer and V. Kotov. A 3/2-approximation algorithm for 3/2-partitioning. *Oper. Res. Lett.*, 39(5):359–362, 2011. 5

[23] S. Khuller and Y. J. Sussmann. The capacitated K-center problem. *SIAM J. Discrete Math.*, 13(3):403–418, 2000. 1, 2

[24] A. Kurpisz, M. Mastrolilli, C. Mathieu, T. Mömke, V. Verdugo, and A. Wiese. Semidefinite and linear programming integrality gaps for scheduling identical machines. In *IPCO 2016*, pages 152–163, 2016. 9

[25] F. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. 4, 8

[26] S. Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM SODA 2015*, pages 696–707, 2015. 5, 12

[27] S. Li. Approximating capacitated k-median with $(1 + \varepsilon)k$ open facilities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 786–796, 2016. 5, 12

[28] R. Lupton, F. M. Maley, and N. E. Young. Data collection for the sloan digital sky survey - A network-flow heuristic. *J. Algorithms*, 27(2):339–356, 1998. 1

[29] H. L. Morgan and K. D. Levin. Optimal program and data locations in computer networks. *Commun. ACM*, 20(5):315–322, 1977. 1, 2

[30] K. Murthy, J. B. Kam, and M. S. Krishnamoorthy. An approximation algorithm to the file allocation problem in computer networks. In *PODS*, 1983. 1

[31] C. A. Phillips, C. Stein, and J. Wein. Task scheduling in networks. *SIAM J. Discrete Math.*, 10(4):573–598, 1997. 1

[32] L. Polácek and O. Svensson. Quasi-polynomial local search for restricted max-min fair allocation. *ACM Trans. Algorithms*, 12(2):13, 2016. 4

[33] Z. Qiu, C. Stein, and Y. Zhong. Minimizing the total weighted completion time of coflows in datacenter networks. In *Proceedings of the 27th ACM SPAA 2015*, pages 294–303, 2015. 2

[34] B. Saha and A. Srinivasan. A new approximation technique for resource-allocation problems. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 342–357, 2010. 6

[35] G. Sen, M. Krishnamoorthy, N. Rangaraj, and V. Narayanan. Exact approaches for static data segment allocation problem in an information network. *Computers and Operations Research*, 62:282 – 295, 2015. 2

[36] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993. 6, 10

[37] G. J. Woeginger. A comment on scheduling two parallel machines with capacity constraints. 2:269 – 272, 2005. 5