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

Let the instance $\mathcal{I}$ of $Q||C_{min}$ have m machines M with demands $D_1 \geq \cdots \geq D_m$ and cardinality constraints $f_1, \ldots, f_m$, and n types of jobs J with capacities $c_1 \geq \cdots \geq c_n$. A supply vector $(s_1, \ldots, s_n)$ indicates the number of jobs of each type available; a supply vector is feasible if together they can satisfy all the demands. An α-approximate supply polyhedra $\mathcal{P}$ has the following properties: any feasible supply vector lies in $\mathcal{P}$, and given any integral vector $(s_1, \ldots, s_n) \in \mathcal{P}$ there is an allocation of the s_j jobs of capacity c_j which satisfies every demand up to an α-factor.

A feasible supply vector $(s_1, \ldots, s_n)$ must lie in the following polytope. Let Supp be a set indicating infinitely many copies of all jobs. For every machine i, let $\mathcal{F}_i := \{S \in \mathsf{Supp} : |S| \leq f_i \text{ and } \sum_{j \in S} c_j \geq D_i\}$ denote all the feasible sets that can satisfy machine i. Let $n(S, j)$ denote the number of copies of job of type j.

$$
\begin{aligned}
\mathcal{P}_{\mathsf{conf}} \; = \{(s_1, \ldots, s_n) : & \\
\forall i \in M, \quad & \textstyle\sum_S z(i, S) = 1 && (1)\\
\forall j \in J, \quad & \textstyle\sum_{i \in M, S} z(i, S) n(S, j) \leq s_j && (2)\\
\forall i \in M, S \notin \mathcal{F}_i, \quad & z(i, S) = 0\} && (3)
\end{aligned}
$$

Theorem 1.1. *Given $(s_1, \ldots, s_n) \in \mathcal{P}_{\mathsf{conf}}$ for an instance $\mathcal{I}$ of $Q|f_i|C_{min}$, there is 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. Throughout the proof we fix $(s_1, \ldots, s_n)$. We start with a feasible solution z to (1)-(3) above. Our objective is to find an assignment of the s_j jobs of capacity c_j to the m machines so that machine i obtains total capacity $\geq D_i/\alpha$ for $\alpha = O(\log D)$. 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 1. *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 $c_j \geq \frac{\bar{D}_i}{16 \log_2 D}$, or (b) $|S| \leq f_i$, $c_j \leq \alpha \cdot \frac{\bar{D}_i}{16 \log_2 D}$, 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 $c_j \geq \frac{\bar{D}_i}{16 \log_2 D}$, 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 1.2. *After* **Step 1**, *$\bar{z}$ satisfies (1) and (2) 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: $\mathsf{FixLargeMachine}(i)$**.** 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 (2). 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 $c_{j_2}>\frac{\bar{D}_{i'}}{16\log_2 D}$, 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 1.3. $\mathsf{FixLargeMachine}$*(i) terminates. Upon termination, the solution $\bar{z}$ satisfies (1) and (2), 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, (1) is satisfied by the modification. (2) 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 (1). Furthermore, since $T\subseteq S\cup j_2$, it can only decrease the LHS of (2) (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 $c_{j_2}\geq\frac{\bar{D}_{i'}}{16\log D}$ and so $T\in\mathcal{F}_i^{(\alpha,\beta)}$. Otherwise, since $S\in\mathcal{F}_i^{(\alpha,\beta)}$, $c_{j_2}<\frac{\bar{D}_{i'}}{16\log D}$, and $c_{j_2}\geq c_{j_1}$ we have $T\in\mathcal{F}_i^{(\alpha,\beta)}$. So at every step $\bar{z}$ maintains (1) and (2) 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 $\bar{z}(i,\{j_1\})=1$. $\square$

Subroutine: $\mathsf{FixBucket}(t)$**.** 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 (1) and (2) 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 1.4. $\mathsf{FixBucket}$*(t) terminates. Upon termination, the solution $\bar{z}$ satisfies (1) and (2), 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 (1) and (2) 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. $\square$

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 $\mathsf{FixLargeMachine}(i)$. 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 $\mathsf{FixBucket}(t)$. 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 (1) and (2) (Claim 1.2), therefore Claim 1.3 and Claim 1.4 imply that it satisfies after the while loop. We encapsulate the above discussion in the following claim about Step 2.

Claim 1.5. **Step 2** *terminates. Upon termination, the modified LP solution $\bar{z}$ is $(1,1)$-feasible, satisfies (1) and (2), 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 (1) 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 1.6. *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 (2), 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 (2). 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. $\square$

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 (2) 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 $c_j \le \frac{\bar{D}_i}{16\log_2 D}$.

$$\sum_{j\in S\setminus J_M} c_j \ge \sum_{j\in S} c_j - |J_M| \cdot \frac{\bar{D}_i}{16\log_2 D} \ge \frac{15\bar{D}_i}{16}$$

Therefore, we have proved the following claim.

Claim 1.7. *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 $(1, 16/15)$-feasible, (c) $\bar{z}$ satisfies (2), and satisfies (1) replaced by $\frac{1}{K} \leq \sum_S \bar{z}(i,S) \leq 1$.*

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:

$$\forall j \in J, \quad \textstyle\sum_{i\in\mathcal{S}} \mathsf{z}_{ij} \leq s'_j \tag{4}$$

$$\forall i \in \mathcal{S}, \quad \textstyle\sum_{j\in J} \mathsf{z}_{ij} c_j \geq \frac{15\bar{D}_i}{16\log_2 D} \tag{5}$$

$$\forall i \in \mathcal{S}, \quad \textstyle\sum_{j\in J} \mathsf{z}_{ij} \leq f_i \tag{6}$$

$$\forall i \in \mathcal{S}, j \in J \text{ with } c_j \geq \tfrac{\bar{D}_i}{16\log_2 D}, \quad \mathsf{z}_{ij} = 0 \tag{7}$$

The last equality follows since $\bar{z}$ was $(1, 16/15)$-feasible and so $\bar{z}(i,S) = 0$ for any set S containing a job j with $c_j \geq \frac{\bar{D}_i}{16\log_2 n}$. The first inequality follows since $\bar{z}$ satisfies (2). To see the second and third point, note that for any $i \in \mathcal{S}$,

$$\sum_{j\in J} \mathsf{z}_{ij} c_j = \sum_j \sum_S \bar{z}(i,S)n(S,j)c_j = \sum_S \bar{z}(i,S) \sum_j n(S,j)c_j \geq \frac{1}{\log_2 D} \cdot \frac{15\bar{D}_i}{16}$$

since $\sum_S \bar{z}(i,S) \geq 1/K$ for all $i \in \mathcal{S}$ and since $\bar{z}$ is $(1, 16/15)$-feasible, we have $\sum_{j=1}^n n(S,j)c_j \geq \frac{15\bar{D}_i}{16}$. Similarly,

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

since for any S, $\sum_{j\in S} n(S,j) \leq f_i$ and $\sum_S \bar{z}(i,S) \leq 1$. Now we use Theorem 1.8 to find an integral allocation z^{int} of the jobs J to machines in $\mathcal{S}$ satisfying (4),(6), and $\sum_{j\in J} z^{\mathsf{int}}_{ij} c_j \geq \frac{7\bar{D}_i}{8\log_2 D}$.

To complete the proof of Theorem 1.1, the final integral assignment is follows. For every $i \in \mathcal{R}$, we assign the configuration S with $\bar{z}(i,S) = 1$. Since $\bar{z}$ is $(1, 16/15)$-feasible, every such machine i gets a total capacity of at least $\frac{\bar{D}_i}{16\log_2 D} \geq \frac{\bar{D}_i}{32\log_2 D}$. All the remaining machines $I \in \mathcal{S}$ obtain a set of jobs giving them capacity $\geq \frac{7\bar{D}_i}{8\log_2 D} \geq \frac{7D_i}{16\log_2 D}$. This completes the proof. $\square$

1.1 Assignment LP for $Q|f_i|C_{min}$

Suppose we are given m machines M with cardinality constraints $f_1, \ldots, f_m$, and n types of jobs J with capacities $c_1 \geq \cdots \geq c_n$. Let $(s_1, \ldots, s_n)$ be a supply vector, that is, there are s_j copies of job j. Suppose there exists a feasible solution to the following LP.

$$\forall j \in J, \quad \textstyle\sum_{i\in M} z_{ij} \leq s_j \tag{8}$$

$$\forall i \in M, \quad \textstyle\sum_{j\in J} c_j z_{ij} \geq D_i \tag{9}$$

$$\forall i \in M, \quad \textstyle\sum_{j\in J} z_{ij} \leq f_i \tag{10}$$

$$\forall i \in \mathcal{S}, j \in J_{res} \text{ with } c_j \geq C_i, \quad z_{ij} = 0 \tag{11}$$

Theorem 1.8. *If (8)-(11) is feasible, then there is an integral assignment z^{int}_{ij} which satisfies (8), (10) and (11), and* $\forall i \in M, \quad \sum_{j\in J} c_j z^{\mathsf{int}}_{ij} \geq D_i - C_i$.

Proof. NEEDS BETTER WRITING We repeat the argument of Shmoys and Tardos [**?**]. Form $\lfloor \sum_{j \in J} z_{ij} \rfloor \leq f_i$ copies of every machine; let N_i be the copies of machine i. Order the jobs with multiplicities s.t. $c_1 \geq c_2 \geq \cdots \geq c_N$ where $N = \sum_j s_j$. Modify z_{ij} to get an assigment z_{ij} for $i \in \cup N_i$ and $j \in [N]$ as follows. We do this for one machine i.

Given z_{ij}'s we form $|N_i| + 1$ groups $S_1, \ldots, S_{|N_i|}, S_{|N_i|+1}$ with $\sum_{j \in S_t} z_{ij} = 1$ for all $1 \leq t \leq |N_i|$ and $\sum_{j \in S_t} z_{ij} < 1$ for $t = |N_i| + 1$. Note that $\sum_{j \in S_t} z_{ij} c_j < c_{j'}$ for $j' \in S_{t-1}$. When we do this modification for all machines, we get a fractional matching solution where all the N_i copies get fractional value 1 but the jobs are at most 1. So, there is an integral matching. The total integral load on machine i is at least $\sum_{t>1} \sum_{j \in S_t} z_{ij} c_j \geq D_i - C_i$ since $z_{ij} = 0$ for $c_j > C_i$.

Cardinality constraint vacuously satisfied. $\square$