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 $c_1 \geq \cdots \geq c_n$. We assume $D_1/D_m \leq n^C$[deepc: i think this can be made wlog with some std trick] 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. To Do!

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.

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

Theorem 1.1. *Given $(s_1, \ldots, s_n) \in \mathcal{P}_{\mathsf{conf}}$, there is an of assignment ϕ of the s_j jobs of capacity c_j to the machines such that for all $i \in M$, $\sum_{j:\phi(j)=i} c_j \geq D_i/\alpha$ for $\alpha = O(\log n)$.*

Proof. Given a feasible fractional solution $\{z(i, S)\}$ to the configuration LP above, we want to efficiently round it to obtain an integer solution which is an α-approximation for the given $Q|k_i|C_{min}$ instance.

We call a job of capacity c_j *large* for machine i if $c_j \geq \frac{D_i}{16C \log n}$, otherwise it is said to be *small* for machine i. For a machine i, we define a relaxed collection of feasible sets $\mathcal{F}_i^{\mathsf{rel}}$ where $S \in \mathcal{F}_i^{\mathsf{rel}}$ if either (a) $S = \{j\}$ and j is large for i, or (b) $c_j < \frac{D_i}{8C \log n}$ for all $j \in S$, $|S| \leq f_i$, and $\sum_{j \in S} c_j \geq D_i/2$.

Partitioning Configurations and Bucketing Demands. Our first step is to modify z such that its support $z(i, S) > 0$ for only $S \in \mathcal{F}_i^{\mathsf{rel}}$ for all i. For every machine i, if $z(i, S) > 0$ and S contains any large job j for i, then we replaces S by $\{j\}$. To be precise, we set $z(i, \{j\}) = z(i, S)$ and $z(i, S) = 0$. We call such singleton configurations *large* for i; all others are small. Note that after this step, $z(i, S) > 0$ only for $S \in \mathcal{F}_i^{\mathsf{rel}}$. Let $\mathcal{F}_i^L$ be the collection of large configurations for i; the rest $\mathcal{F}_i^S$ being small configurations. Define $z^L(i) := \sum_{S \in \mathcal{F}_i^L} z(i, S)$ be the total large contribution to i, and let $z^S(i) := 1 - z^L(i)$ the small contribution.

The next step of our algorithm is to partition the demands into buckets depending on their requirement values D_i. To this end, we say that demand i belongs to *bucket* t if $2^{t-1} \leq D_i < 2^t$ (we assume wlog by scaling that the smallest demand is 1). We let $B^{(t)}$ to denote the bucket t. Note that the number of buckets $K \leq C \log n$; this drives the approximation factor. We make one observation.

Claim 1.2. *For any t, let i and i' be two machines in $B^{(t)}$ and let $f_i \leq f_{i'}$. Let $z(i, T) > 0$ for some small configuration for i. Then $T \in \mathcal{F}_{i'}^{\mathsf{rel}}$ and $\sum_{j \in T} c_j \geq D_{i'}/2$.*

Proof. Note that since $z(i, T) > 0$, we have $\sum_{j \in T} c_j \geq D_i \geq 2^{t-1} \geq D_{i'}/2$. Furthermore, for any $j \in T$, we have $c_j \leq \frac{D_i}{8C \log n} \leq \frac{2^t}{8C \log n}$. Therefore any other machine $i' \in B^{(t)}$, T satisfies two conditions of being in $\mathcal{F}_{i'}^{\mathsf{rel}}$. Now if $f_{i'} \geq f_i$, we get $|T| \leq f_{i'}$ as well. □

Before describing our subroutines, we make a few definitions. All of these are with respect to a solution z. A machine i is called *rounded* if there exists $S \in \mathcal{F}_i^{\mathsf{rel}}$ with $z(i, S) = 1$. We let $\mathcal{R}$ denote the rounded demands. The remaining machines are of three kinds: *large* ones with $z^L(i) = 1$, *hybrid* ones with $z^L(i) \in (0, 1)$ and *small* ones with $z^L(i) = 0$. Let $\mathcal{L}, \mathcal{H}, \mathcal{S}$ denote these respectively.

Subroutine: $\mathsf{FixBucket}(t)$**.** This takes a bucket t with more than one hybrid machine, and modifies the z-solution such that there is at most one hybrid machine in t. Other machines in other buckets are unaffected.

Among the hybrid machines $B^{(t)}$, let i be the one with the smallest f_i. Let i' be any other hybrid machine. We know there is at least one more. We now *modify* z as follows. Since $z^L(i') > 0$, there exists a large configuration $\{j'\}$ for i' with $z(i', \{j'\}) > 0$. Similarly, since $z^L(i) < 1$, there must exist a *small* configuration $T \in \mathcal{F}_i^{\mathsf{rel}}$ such that $z(i, T) > 0$. By Claim 1.2, note that $T \in \mathcal{F}_{i'}^{\mathsf{rel}}$ as well. We then perform the following change: increase $z(i, \{j'\})$ and $z(i', T)$ by δ, and decrease $z(i', \{j'\})$ and $z(i, T)$ 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 on performing this process as long as possible; since we always transfer large configuration assignments to demands which appear earlier in the total order, this process will stop at some point. At this point, we add whichever demands are integrally assigned by large configurations to the set $\mathcal{R}$, i.e., all $i \in B^{(t)}$ for which $z(i, S) = 1$ for some $S \in \mathcal{F}_i^{\mathsf{rel}}$ are added to $\mathcal{R}$.

Claim 1.3. *Fix Bucket on t produces a solution with at most one hybrid machine in $B^{(t)}$.*

Subroutine: $\mathsf{FixLargeMachine}(i)$**.** This takes input a large machine $i \in B^{(t)} \setminus \mathcal{R}$ with $z^L(i) = 1$ and modifies z such that at the end i enters $\mathcal{R}$. Since $i \notin \mathcal{R}$ in the beginning, there must exist then two large configurations with $z(i, \{j_1\}) \in (0, 1)$ and $z(i, \{j_2\}) \in (0, 1)$. Let j_1 be the job with the smallest capacity among all large configurations $(i, \{j\})$ with $z(i, \{j\}) > 0$. Two cases arise. In the simple case, there exists no $i' \notin \mathcal{R}$ and $S' \in \mathcal{F}_{i'}^{\mathsf{rel}}$ with $z(i', S') > 0$ 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) and therefore we can round up $z(i, \{j_1\}) = 1$ (zeroing out all other i's $z(i, S)$'s) without violating (2). We then add $(i, \{j_1\})$ to $\mathcal{R}$.

Otherwise, there exists a machine i' (which could be in a different bucket) and a set $S \in \mathcal{F}_{i'}^{\mathsf{rel}}$ such that $z(i', S) \in (0, 1)$ and $j_1 \in S$. Now define the set T as follows. If $c_{j_2} > \frac{D_{i'}}{8C \log n}$, then $T = \{j_2\}$; otherwise $T = S - j_1 + j_2$. Note that in either case $T \in \mathcal{F}_{i'}^{\mathsf{rel}}$. In the first case, j_2 is large for i'. In the second case, $|T| = |S|$ and $\sum_{j \in T} c_j \geq \sum_{j \in S} c_j$ since $c_{j_2} \geq c_{j_1}$ by choice of j_1.

We modify z-as follows. We decrease $z(i, \{j_2\})$ and $z(i', S)$ by δ and increase $z(i, \{j_1\})$ and $z(i', T)$ by δ till one of the values becomes 0 or 1. As before, this preserves the LHS of (1) and can only decrease the LHS of (2) (for jobs $j \in S \setminus j_1$ if $T = \{j_2\}$). This process ends with assigning $z(i, \{j_1\}) = 1$ and we add $(i, \{j_1\})$ to $\mathcal{R}$.

Claim 1.4. *Subroutine Fix Large Machine i modifies the LP solution and adds i to $\mathcal{R}$.*

Note that Fix Large Machine can make a small machine i' hybrid or large for its bucket since $z^L(i')$ could potentially increase. We run the following while loop in Step 1 of te algorithm.

Step 1: Taking care of large machines

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

- If $i \in \mathcal{L}$, then $\mathsf{Fix\text{-}Large\text{-}Machine}(i)$. Note that i enters $\mathcal{R}$ after this. This can increase the number of hybrid machines across buckets.
- For all $1 \leq t \leq K$, if $B^{(t)}$ contains more than one hybrid machine, then $\mathsf{Fix\text{-}Bucket}(t)$. This can increase the number of machines in $\mathcal{L}$.

The above while loop terminates in at most m iterations, since the first bullet point adds a machine to $\mathcal{R}$.

Claim 1.5. *At the end of* **Step 1***, we have for every bucket t, at most one $i \in B^{(t)} \setminus \mathcal{R}$ has $z^L(i) \in (0, 1)$ and the rest have $z^S(i) = 1$. Furthermore, for every $i \in \mathcal{S}$ and $z(i, S) > 0$, we have $\sum_{j \in S} c_j \geq D_i/2$.*

Step 2: Taking care of hybrid machines. Let $\mathcal{H}$ be the set of hybrid machines at this point. We know that $|\mathcal{H}| \leq K \leq C \log n$ since each bucket has at most one hybrid machine. For any machine $i \in \mathcal{H}$ with $z^L(i) \leq 1-1/K$, we zero-out all its large contribution. More precisely, for all j large for i we set $z(i, \{j\}) = 0$. Note that (1) is no longer true, but it holds with RHS $\geq 1/K$. Note that these machines enter $\mathcal{S}$.

At this point, we have $\mathcal{H}$ where 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 $z(i, \{j\}) > 0$. Let G be a bipartite graph with $\mathcal{H}$ on one side and J' on the other and we draw an edge (i, j) iff j is large for i.

Claim 1.6. *There is a matching in G saturating 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 $\sum_{j \in J''} s_j \geq \mathcal{H}'$. Since z satisfies (2), we get

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

The 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 this job and add i to $\mathcal{R}$. Let $J_M \subseteq J'$ be the subset of jobs allocated; note $|J_M| \leq K$. After this point we have only small machines remaining. For every $i \in \mathcal{S}$ and every small configuration S with $z(i, S) > 0$, we move this mass to $z(i, S \setminus J_M)$. Note that $\sum_{j \in S \setminus J_M} c_j \geq \sum_{j \in S} c_j - |J_M| \cdot \frac{D_i}{8C \log n} \geq 3D_i/8$ where we use the fact that $\sum_{j \in S} c_j \geq D_i/2$ (by Claim 1.5) and $K \leq C \log n$.

Claim 1.7. *At the end of* **Step 2**, *we have a set of residual machines $\mathcal{S}$ and a set of residual jobs J_{res} and a solution $z(i, S)$ where*

1. *For all $i \in \mathcal{S}$ we have $z(i, S) > 0$ iff $|S| \leq f_i$, $\sum_{j \in S} c_j \geq 3D_i/8$, and $c_j < \frac{D_i}{8C \log n}$ for all $j \in S$.*
2. $\forall i \in \mathcal{S},\ 1 \geq \sum_S z(i, S) \geq 1/K \geq \frac{1}{C \log n}$.
3. $\forall j \in J_{res},\ \sum_i z(i, S) n(S, j) \leq s_j$.

Step 3: Taking care of Small Machines. We convert the LP solution in Claim 1.7 to an assignment LP solution in the standard way. For every $i \in \mathcal{S}$ and $j \in J_{res}$ define $z_{ij} = \sum_S z(i, S) n(S, j)$. Note that this satisfies the constraint of the assignment LP:

$$\begin{aligned} \forall j \in J_{res}, \quad & \textstyle\sum_{i \in \mathcal{S}} z_{ij} \leq s_j \\ \forall i \in \mathcal{S}, \quad & \textstyle\sum_{j \in J_{res}} z_{ij} c_j \geq \frac{3D_i}{8C \log n} \\ \forall i \in \mathcal{S}, \quad & \textstyle\sum_{j \in J_{res}} z_{ij} \leq f_i \\ \forall i \in \mathcal{S}, j \in J_{res} \text{ with } c_j \geq \frac{D_i}{8C \log n}, \quad & z_{ij} = 0 \end{aligned}$$

The last equality follows from point 1 of Claim 1.7. The first inequality follows from point 3 of Claim 1.7. To see the second and third point, note that for any $i \in \mathcal{S}$,

$$\sum_{j \in J_{res}} z_{ij} c_j = \sum_j \sum_S z(i, S) n(S, j) c_j = \sum_S z(i, S) \sum_j n(S, j) c_j \geq \frac{1}{C \log n} \cdot \frac{3D_i}{8}$$

and,

$$\sum_{j \in J_{res}} z_{ij} = \sum_j \sum_S z(i, S) n(S, j) = \sum_S z(i, S) \sum_j n(S, j) \leq f_i$$

since for any S, $\sum_{j \in S} n(S, j) \leq f_i$.

Now we use Theorem 1.8 to find an allocation of J_{res} to machines $\mathcal{S}$ such that every machine gets capacity $\geq \frac{D_i}{4C \log n}$.

$\square$

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

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

$$\forall j \in J, \quad \sum_{i \in M} z_{ij} \leq s_j \tag{4}$$

$$\forall i \in M, \quad \sum_{j \in J} c_j z_{ij} \geq D_i \tag{5}$$

$$\forall i \in M, \quad \sum_{j \in J} z_{ij} \leq f_i \tag{6}$$

$$\forall i \in \mathcal{S}, j \in J_{res} \text{ with } c_j \geq C_i, \quad z_{ij} = 0 \tag{7}$$

Theorem 1.8. *If (4)-(7) is feasible, then there is an integral assignment z_{ij}^{int} which satisfies (4), (6) and (7), and* $\forall i \in M, \quad \sum_{j \in J} c_j z_{ij}^{\mathsf{int}} \geq D_i - C_i$.

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. $c_1 \geq c_2 \geq \dots \geq c_N$ 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 $S_1, \dots, S_{|N_i|}, S_{|N_i|+1}$ 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. ∎