GENERALIZED UNRELATED MACHINE SCHEDULING PROBLEM

Shichuan Deng Jian Li Yuval Rabani
Presentation prepared by: Anthony Kervin
COMP5112 – Nov 24, 2025

The Problem – Generalized load-balancing (GLB)

- m machines for each $i \in M$
- n jobs for each $j \in J$.
- $p_{ij} > 0$ is the time it takes machine i to do job j
- σ is the assignment of all jobs to machines $J \to M$.
- $p_i[\sigma] = \{p_{ij} \cdot 1[\sigma(j) = i]\}_{j \in J}$ is machine i's load vector for each assigned job j
  - $p_i[\sigma] = p_{ij}$ if j is assigned to machine i under σ, 0 otherwise.
- We must assign jobs in such a way that workload is minimized
- How do we measure workload?

Symmetric Monotone Norms

- Let ψ be a symmetric monotone norm
- **Symmetric**: $\psi_i(p_i[\sigma]') = \psi_i(p_i[\sigma])$
  - A load does not change if its measured vector is ordered differently
- **Monotone**: $\psi(u) \leq \psi(v)$, when $0 \leq u \leq v$
  - Increasing an vector's entry can only increase its total load
- **Norm**: a vector load-measuring function
  - ψ can be
    - $\mathcal{L}_1 \leftarrow$ sum of all loads in a vector
    - $\mathcal{L}_\infty \leftarrow$ returns the maximum load in a vector
    - $Top_k \leftarrow$ Returns a sum of the largest k loads in a vector
- So, ψ is a function that measures the load of a vector where its output is the same regardless of what order its entries are in and only grows when the vector's entries grow

Measuring Load and our Goal

- $load_i(\sigma) = \psi_i(p_i[\sigma])$
  - Load of machine i under assignment σ
  - ψ_i is an inner norm, it measures each machine's load
- $load(\sigma) = \{load_i(\sigma)\}_{i \in M}$
  - Load vector of all machine loads under assignment σ
- $\Phi(\sigma) = \phi(\mathrm{load}(\sigma))$
  - The "generalized makespan"
  - Final measured load of all machines under assignment σ
  - ϕ is an outer norm, measures the total load of all machines
- **Our goal is to find an assignment σ that minimizes $\Phi(\sigma)$**
  - Paper shows WHP, an approximation factor of $O(\log n)$ of optimal can be achieved

Lower Bound → NP-hard $(1-\epsilon)\ln n$ approximation[2]

- Translate GLB to unweighted set cover:
  - $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ of $[n]$ (n jobs)
    - Each S_i contains the jobs that are assigned to machine i
  - Let $I \subseteq \mathcal{S}$ be a subset that contains the minimum number of sets from $\mathcal{S}$ to cover all jobs
    - We want to find the minimum size of I, such that $\bigcup_{i \in I} S_i = [n]$
  - For machine i's load vector $p_i[\sigma]$, each entry $p_{ij} = 1$ if $j \in S_i$, and ∞ if $j \notin S_i$
  - Set $\psi_i = \mathcal{L}_\infty$ and $\phi = \mathcal{L}_1$
  - Under any finite assignment σ, $\psi_i(p_i[\sigma]) = 1$, so $\phi\big(load(\sigma)\big) = |I_\sigma|$
    - Here I_σ is the subset of machines that get ≥ 1 job assignment from σ
    - Notice that we find the minimum number of machines to do all jobs
- Here, GLB → Set Cover. Thus, we get:
  - **THEOREM 1.1** For every fixed constant $\epsilon > 0$, it is NP-hard to approximate GLB within a factor of $(1-\epsilon)\ln n$, even when $\phi = \mathcal{L}_1$ and $\psi_i = \mathcal{L}_\infty$ for each $i \in M$
  - This provides a lower bound for our approximation of GLB

Finding the upper bound

THEROREM 1.2:

There exists a polynomial time randomized algorithm for GLB that, with high probability, achieves an approximation factor of $O(\log n)$

Preliminaries

- $u^{\downarrow}$ is the non−increasingly sorted version of the vector u
  - $u = (1,3,2) \rightarrow u^{\downarrow} = (3,2,1)$
- Let $\sigma^*: \mathcal{J} \rightarrow \mathcal{M}$ be an optimal job assignment of job sets to machines
- Let $o = load(\sigma^*) \in \mathbb{R}_{\geq 0}^{m}$ be the optimal outer load vector
- Let $opt = \phi(o)$ be the optimal solution to the GLB problem

Preliminaries ctd.

- $Top_k\colon \mathbb{R}^{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ is the top-k norm that outputs the sum of the k largest values in a vector, where $k \leq |\mathcal{X}|$.
- $a^+ = \max\{a, 0\}$
- By Claim 2.1, for each $u \geq 0$ and $k \in [n]$:

$$Top_k(u) = \min_{t \geq 0} \left\{ kt + \sum_{j \in [n]} \left(u_j - t\right)^+ \right\} = k u_k^{\downarrow} + \sum_{j \in [n]} \left(u_j - u_k^{\downarrow}\right)^+,$$

- Where the minimum is attained at the k-th largest entry of u

Top-k example

Table 2.1: Top-k example where $u = (6,8,2,9)$ and $k = 2$

t	$kt + \sum_{j \in [n]} (u_j - t)^+$	**Output**
0	$0 + (6 + 8 + 2 + 9)$	25
2	$4 + (4 + 6 + 0 + 7)$	21
8	$16 + (\max\{-2,0\} + 0 + \max\{-6,0\} + 1)$	17
9	$18 + (\max\{-3,0\} + \max\{-1,0\} + \max\{-7,0\} + 0)$	18

minimum output is achieved when $t = 8$, which is the k^{th} entry of u, $k = 2$, $u_k = 8$. The sum of the two largest entries in u is $9 + 8 = 17$. So, $u_k^{\downarrow}$ is the optimal value of t

Useful Lemmas

- LEMMA 2.1. ([4]). If $u, v \in \mathbb{R}_{\geq 0}^{\mathcal{X}}$ and $\alpha \geq 0$ satisfy $Top_k(u) \leq \alpha \cdot Top_k(v)$ for each $k \leq |\mathcal{X}|$, one has $\psi(u) \leq \alpha \cdot \psi(v)$ for any symmetric monotone norm $\psi: \mathbb{R}^{\mathcal{X}} \rightarrow \mathbb{R}\_ \geq 0$
  - If $Top_k(u) \leq \alpha \cdot Top_k(v)$, then $\psi(u) \leq \alpha \cdot \psi(v)$
  - States that we only need to bound the Top-k load of the vector to bound the whole vector's workload under any norm
  - Used to bound the LP
- LEMMA 2.2. Let $X_1, \ldots, X_n$ be independent Bernoulli variables with $\mathbb{E}[X_i] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. For $v \geq 6\mu$, one has $Pr[X \geq v] \leq 2^{-v}$
  - WHP a value will not exceed expectation by a factor of 6 or greater
  - Later used to prove approximation WHP

CONFIGURATION LP FOR GLB

Primal LP

- Let $x_{i,J} \in [0,1]$ for each $i \in \mathcal{M}$ and $J \subseteq \mathcal{J}$
  - Indicates if job-set J is assigned to machine i
  - Acts as a probability for fractional configuration assignment

- (P-LB(R, λ, τ)) min 0
- (P-LB.1) s.t. $k\rho_k + \sum_{i \in \mathcal{M}, J \subseteq \mathcal{J}} \left(h\left(\frac{\psi_i(J)}{\tau}\right) - \rho_k \right)^+ x_{i,J} \leq Top_k(\varrho) \quad \forall k \in POS$
- (P-LB.2) $\sum_{J \subseteq \mathcal{J}} x_{i,J} \leq 1 \quad \forall i \in \mathcal{M}$
- (P-LB.3) $\sum_{i \in \mathcal{M}, J \ni j} x_{i,J} \geq \lambda \quad \forall j \in \mathcal{J}$
- (P-LB.4) $\sum_{i,J: \psi_i(J) > \tau} x_{i,J} \leq 0$
- (P-LB.5) $\sum_{i \in \mathcal{M}, J \subseteq \mathcal{J}} x_{i,J} \leq n$

Primal LP ctd.

(P-LB(R, λ, τ)) min 0

- Why min 0?
- This LP is not looking for its solution
  - It is checking the feasibility of the guessed set R
  - Cannot find solution because there are exponentially many constraints ($m \cdot 2^n$)
- Here, $R = \{\rho_k\}_{k \in POS}$ where each ρ_k is a guess of $o_k^{\downarrow}$ (the k-th value of opt load vector)
  - So R is a guess of the optimal load vector $o^{\downarrow}$
- How do we guess R?
  - Before LP is constructed, we scale R's entries by enumerating $(i^*, j^*) \leftarrow$ heaviest machine-job load
    - Done by looking at each (i, j) and treating it as heaviest load
  - We set this load to be $\frac{1}{n}$, the optimal load vector's range becomes $[0,1]$
  - Then the sorted guesses of the entries of R fall within the range of $[\frac{1}{2mn}, 1]$

(P-LB.1) $$k\rho_k + \sum_{i\in\mathcal{M}, J\subseteq\mathcal{J}} \left(h\left(\frac{\psi_i(J)}{\tau}\right) - \rho_k\right)^+ x_{i,J} \le Top_k(\varrho) \quad \forall k \in POS$$

- From LEMMA 2.1:
  - Bound Top-k norm to bound whole vector
- Instead of Top-k for all k in $[m]$
  - We check $\mathrm{k} \in POS$ = set of powers of 2 up to m
    - If $m = 10$, $POS = \{1,2,4,8,10\}$
- ϱ is the extended version of R
  - Where it fills the gaps of POS indices with the next power of 2, so there are m entries
  - Later in the paper, $Top_k(\varrho)$ is proven to be $\le 8Top_k(o)$
- τ is used to remove configurations of $\psi_i(J)$ that are too big
- We use $h(\psi_i(J))$ to round loads up to the next $k \in POS$
  - Simplifies analysis

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

- Each machine may be fractionally matched to multiple configurations
- The sum of all these weights per machine ≤ 1

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

- Each job must be assigned to an extent of at least $\lambda \leq 1$

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

- Each assignment of a job-set J to a machine i with a load > τ is set to 0 (pruned)

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

- The number of fractional matchings of job sets to machines cannot exceed a total of n

DUAL LP

Why a dual?

- Number of variables in P-LB(R,λ,τ) is exponential ($m \cdot 2^n$)
  - Cannot write an LP wit that many variables
- Number of constraints is polynomial
  - $O(n$ jobs $+m$ machines $+\log n$ guesses)
- So, the dual will have polynomial many variables (small enough to work)
  - It will have exponentially many constraints, but we use the ellipsoid method later

The Dual

(D-LB(R, λ, τ)) $\max - \sum_{k \in POS}(Top_k(\varrho) - k\rho_k)\, r_k - \sum_{i \in \mathcal{M}} y_i + \sum_{j \in \mathcal{J}} z_j - nt$

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

$$r, s, t, y, z \geq 0.$$

- r_k = cost associated with the Top_k load of $x_{i,J}$ exceeding the threshold ρ_k
  - Forces primal to decide if a configuration (i, J) is too heavy to be in the solution
- y_i = penalty assigned to using machine i
  - The higher y_i is, the lower the weight is in the primal for configurations of machine i
- z_j is a reward for covering job j
  - Encourages the primal to assign job j to a machine
- t = price/cost
  - Restricts the primal from using too many configurations
- s accounts for the configurations from primal that were forced to 0
  - All configurations $> \tau$ are accounted for in the dual

Primal Feasibility from Dual

(D-LB(R, λ, τ)) $\max - \sum_{k \in POS} (Top_k(\varrho) - k\rho_k)\, r_k - \sum_{i \in \mathcal{M}} y_i + \sum_{j \in \mathcal{J}} z_j - nt$

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

$$r, s, t, y, z \ge 0.$$

- The Dual has a trivial 0 solution
  - $(r, s, t, y, z) = (0,0,0,0,0)$
- Also has an unbounded, scale-invariant solution
  - Since there is a feasible solution (r, s, t, y, z), (cr, cs, ct, cy, cz) is also feasible for $c \ge 0$
- The primal is only feasible when the dual is bounded
  - For primal to be bounded, dual's optimal solution must be 0

Dual to Polytope

- Dual has exponentially many constraints
  - Ellipsoid method could not separate them all to find an optimal solution
- Dual's optimal value being 0 is equivalent to a polytope $Q(R, \lambda, \tau)$ being empty

$$(Q(R, \lambda, \tau)) \qquad \{(r, s, t, y, z) \geq 0 \mid -\sum_{k \in POS} (Top_k(\varrho) - k\rho_k) r_k - \sum_{i \in \mathcal{M}} y_i + \lambda \sum_{j \in \mathcal{J}} z_j - nt \geq 1;$$

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

- OBSERVATION 3.1 P-LB(R, λ, τ) is feasible if and only if $Q(R, \lambda, \tau)$ is empty
  - Since if Q is empty, D-LB is bounded, and P-LB is feasible

Finding R

- Apply ellipsoid method on $Q(R, \lambda, \tau)$ with different guesses of R
  - Separation oracle finds separating hyperplane (constraint violation)
    - Cuts feasible region
  - Separation oracle fails to find a violation
    - Certifies some polytope $Q(R, \lambda', \tau')$ is non-empty
      - Primal P-LB(R, λ', τ') is not feasible for that R
      - Pick a different R (polynomial many guesses)
- For clarity, authors keep $(\lambda, \tau) \in \{\left(\frac{1}{2}, \frac{3}{2}\right), (1,1)\}$
  - Job coverage and load pruning threshold
- LEMMA 3.1 Fix R. There exists a polynomial time algorithm that, given $(r, s, t, y, z) \geq 0$, either outputs a violated constraint in $Q(R, \frac{1}{2}, \frac{3}{2})$, or certifies that $Q(R, 1, 1)$ is non-empty

Finding R ctd.

- Enumerate all possible R such that the following holds:
- 1. $\rho_k \in \left[o_k^{\downarrow}, 2o_k^{\downarrow}\right)$ for $k \in POS$ s.t. $o_k^{\downarrow} \geq \frac{o_1^{\downarrow}}{2m}$
  - we guess ρ_k to be within a factor of 2 of the true optimal load value
- 2. $\rho_k = 2^{\left\lceil \log_2\left(\frac{o_1^{\downarrow}}{2m}\right)\right\rceil} \in \left[\frac{o_1^{\downarrow}}{2m}, \frac{o_1^{\downarrow}}{m}\right]$ for $k \in POS$ s.t. $o_k^{\downarrow} < \frac{o_1^{\downarrow}}{2m}$
  - Forcing the values smaller than $\frac{o_1^{\downarrow}}{2m}$ into groups of the next largest power of 2
  - Minimizes number of approximations
- These constraints define R^*, an optimal guess vector determined by the opt o
- This guarantees we will eventually run the optimal guessed vector to get our solution

ANALYSIS

Start

- Enumerate (i^*, j^*) to scale range and guess of R
- Begin with a point (r, s, t, y, z) in the polytope
- Repeatedly call separation oracle
- It always returns a violated constraint (a separation hyperplane)
- Eventually proves $Q_{\mathcal{H}}(R, \frac{1}{2}, \frac{3}{2})$ of the polynomial-sized subset $\mathcal{H} \subseteq \mathcal{M} \times 2^{\mathcal{J}}$ is empty
  - $\mathcal{H}$ contains the indexed constraints the oracle returns
- We directly solve P--$LB_{\mathcal{H}}(R, \frac{1}{2}, \frac{3}{2})$ with $\mathcal{H}$ to get a vertex solution $\hat{x}$
- We keep trying guesses of R to find all the fractional matchings $\hat{x}$
- We use $\hat{x}$ to solve P-LB$(R, \frac{1}{2}, \frac{3}{2})$
  - This is done by eliminating all variables not indexed by $\mathcal{H}$ and extending $\hat{x}$ with zeros

Randomized Rounding

- Use Randomized Rounding to get feasible assignment of jobs
- Let $\mathcal{I} \leftarrow \emptyset$ and $T = \lceil 6 \ln n \rceil$
- For each bucket (power of 2) $t = 1, \dots, T$ and each $(i, J) \in \widehat{\mathcal{H}}$, set $\mathcal{I} \leftarrow \mathcal{I} \cup \{(i, J)\}$ independently with probability $\hat{x}_{i,J}$
  - Where $\widehat{\mathcal{H}}$ contains all pairs (i, J) for which the LP solution $\hat{x}_{i,J} > 0$
- Expected number of configurations given to each machine:

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

- Applying Chernoff from LEMMA 2.2 for random variables $1[(i, J) \in \mathcal{I}], (i, J) \in \widehat{\mathcal{H}}$:

(3.1) $$\Pr[|\{J: (i, J) \in \mathcal{I}\}| > 6T] = \Pr\left[\textstyle\sum_{J:(i,J)\in\widehat{\mathcal{H}}} 1[(i, J) \in \mathcal{I}] > 6T\right] \leq 2^{-6T} \leq \frac{1}{n^{24}}$$

- WHP number of configurations machine i gets after T rounding steps does not exceed a factor of 6 of T

Job Coverage

- Probability of a job $j \in \mathcal{J}$ not being in any selected $J \in \mathcal{I}$ is also very unlikely

(3.2) $$\prod_{(i,J)\in\widehat{\mathcal{H}}:J\ni j}\left(1-\hat{x}_{i,J}\right)^{T} \leq \prod_{(i,J)\in\widehat{\mathcal{H}}:J\ni j}\exp\left(-T\hat{x}_{i,J}\right) = \exp(-T\sum_{(i,J)\in\widehat{\mathcal{H}}:J\ni j}\hat{x}_{i,J}) \leq \frac{1}{n^3}$$

- (P-LB.3) is used here to get $\sum_{(i,J)\in\widehat{\mathcal{H}}:J\ni j}\hat{x}_{i,J} \geq 1/2$ in the last inequality
- WHP every job is covered by a machine

Ensuring Rounded Load is within a Factor of $O(\log n) \cdot \mathrm{opt}$

- Let $Y_t = |\widehat{\mathcal{H}}_t \cap \mathcal{I}|$ be the number of configurations chosen by rounding that fall into bucket t

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

- Using Markov's inequality with probability $\geq \frac{1}{2}$
  - $\mathrm{P}\left[Y_t \leq 2\mathbb{E}[Y_t]\right] \geq \frac{1}{32n}$
  - WHP number of configurations does not exceed twice its expectation
  - For every bucket t, each configuration has a value of 2^t
  - At each bucket, $Y_t \cdot (2^t - \rho_k)^+$ to the total Top_k load
  - Summed over all buckets, we get the following:

$$\sum_{(i,J) \in \mathcal{I}} \left(h\left(\frac{2\psi_i(J)}{3}\right) - \rho_k\right)^+ = \sum_{t \in PWR} \sum_{(i,J) \in \mathcal{I} \cap \widehat{\mathcal{H}}_t} (2^t - \rho_k)^+ = \sum_{t \in PWR} Y_t \cdot (2^t - \rho_k)^+$$

(3.3) $\leq \sum_{t \in PWR} 2\mathbb{E}[Y_t](2^t - \rho_k)^+ \leq 2T \sum_{t \in PWR} \sum_{(i,J) \in \widehat{\mathcal{H}}_t} \hat{x}_{i,J}(2^t - \rho_k)^+ \leq 2T(Top_k(\varrho) - k\rho_k)$

- This proves the total rounded load is within a factor of $2(6 \ln n)$ of the LP load

Putting it all together

- With (3.1), (3.2), and (3.3), the union bound, a large n, and with a probability of at least $\frac{1}{32n} - \frac{3n+5}{n^{24}} - \frac{1}{n^2} \geq \frac{1}{64n}$, the following hold true:
- (1) for each $i \in \mathcal{M}, |\{J: (i,J) \in \mathcal{I}| \leq 6T \leq 38 \ln n$
  - For each machine, its number of assignments is fewer than $38 \ln n$
- (2) for each $j \in \mathcal{J}, \exists (i,J) \in \mathcal{I}$ such that $j \in J$
  - For each job, there exists a matching that contains it
- (3) for each $k \in POS$,

$$\sum_{(i,J) \in \mathcal{I}} \left(h\left(\frac{2\psi_i(J)}{3}\right) - \rho_k \right)^+ \leq 14 \ln n \left(Top_k(\varrho) - k\rho_k\right)$$

  - Meaning, after rounding the Top-k norm of our approx load is $\leq O(\log n) \cdot Top_k(o)$
- Repeat randomized rounding $64n$ times to boost success probability:
  - $1 - \left(1 - \frac{1}{64n}\right)^{64n} \geq 1 - e^{-1} \geq 0.6.$
  - Here, success means (1), (2), and (3) occur

Assigning One Job-Set to each Machine

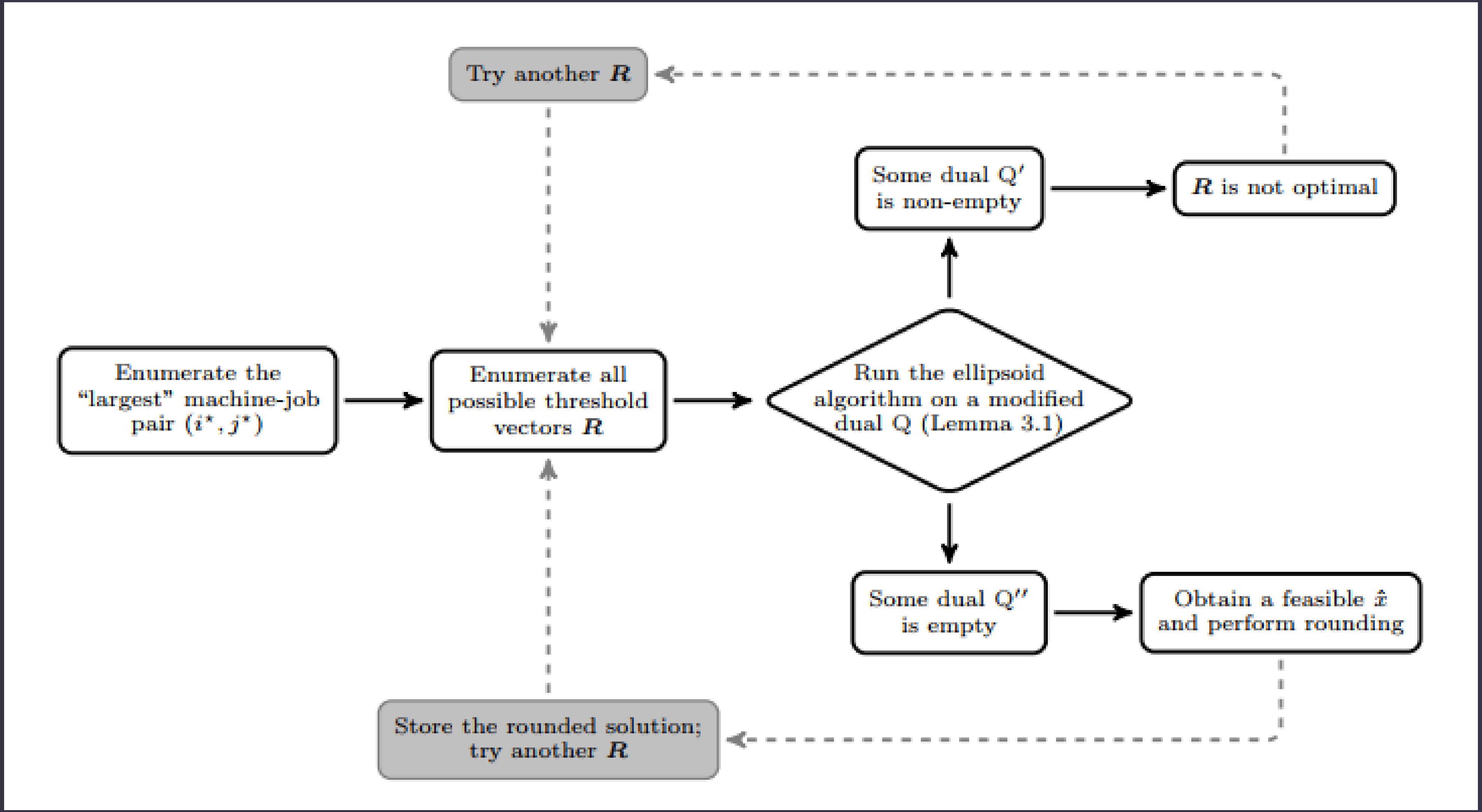
- Rounding assigns job sets fractionally to each machine
  - Each machine may have multiple job-set assignments
- We merge each machine's sets into a single job-set
  - $J_i \leftarrow \bigcup_{J:(i,J)\in\mathcal{J}} J$
  - $\bigcup_{i\in\mathcal{M}} J_i = \mathcal{J}$ (from (2))
- From (1), we are only merging $O(\log n)$ configurations for each machine
- From (3), we find that the load of the merged sets is still within $O(\log n)$ of opt

$$(3.4) \sum_{i\in\mathcal{M}} \left(\frac{2\psi_i(J_i)}{3} - 38\ln n \cdot \rho_k\right)^+ \leq 14\ln n\,(Top_k(\varrho) - k\rho_k)$$

- Therefore, we get an approximated job-assignment σ that is within a factor of $O(\log n)$ of opt

Conclusion

- GLB has too many constraints to simply solve with an LP
- We, instead, form a configuration LP from a guessed heaviest job-machine pair (i^*, j^*) and R
- Bound it's dual to prove feasibility
- To bound the dual, translate it into a feasibility system (polytope)
- Use the ellipsoid method and its separation oracle to prove the polytope is empty
- Polytope is empty → dual is bounded → primal is feasible for this R
- We get a fractional solution $\hat{x}$ and perform randomized rounding
- Marge each machine's fractionally matched configurations together into one set per machine
- By doing this, we find an assignment σ that minimizes the general makespan $\Phi(\sigma)$ within a factor of $O(\log n)$ of the optimal assignment.



From the paper [1]

References

- [1] Deng, S., Li, J., & Rabani, Y. (2023). Generalized Unrelated Machine Scheduling Problem. *Society for Industrial and Applied Mathematics*, 2898-2916. https://doi-org.proxy.library.carleton.ca/10.1137/1.9781611977554.ch110
- [2] STOC '14: Proceedings of the forty-sixth annual ACM symposium on Theory of computing Pages 624 - 633 https://doi.org/10.1145/2591796.2591884
- [3] Sharat Ibrahimpur and Chaitanya Swamy. Minimum-norm load balancing is (almost) as easy as minimizing makespan. In 48th International Colloquium on Automata, Languages, and Programming, volume 198 of LIPIcs, pages 81:1–81:20, 2021. doi:10.4230/LIPIcs.ICALP.2021.81.
- [4] G. H. Hardy, J. E. Littlewood, and G. P´olya. *Inequalities*. Cambridge University Press, 1934.