Makespan Minimization on Unrelated Machines

Monday, 2 December 2024 7:47 PM

Given: m machines M, n jobs J

job j takes time p_{ij} on machine i

- Let $\sigma : J \to M$ be an assignment of jobs to machines.
- Then load l_i on machine $i = \sum_{j : \sigma(j) = i} p_{ij}$
- Makespan $= \max_i l_i$

Problem: Obtain an assignment of jobs to machines to minimize the makespan.

- NP-hard, by reduction from Subset-Sum.
- NP-hard to approximate better than $3/2$, by reduction from 3-dimensional matching.

We will now give a 2-approximation algorithm, based on LP rounding.

LP: $\min T$

$$\forall j \quad \sum_i x_{ij} \geq 1$$
$$\forall i \quad \sum_j p_{ij} x_{ij} \leq T$$
$$T, x_{ij} \geq 0 \quad \forall i, j$$

This LP isn't very good.
"integrality gap" of m

Example: 1 job, m machines, $p_{ij} = m$ $\forall i, j$
integral soln. $= m$,
fractional soln. $= 1$.

Suppose instead we "guessed" the optimal makespan T.

Consider $LP(T)$: $\min 0$

$$\forall i, \quad \sum_j p_{ij} x_{ij} \leq T$$
$$\forall j, \quad \sum_i x_{ij} \geq 1$$
$$\forall i, j \quad x_{i,j} \geq 0$$
$$\forall i, j : p_{ij} > T, \quad x_{ij} = 0$$

(now if we choose $T < m$, the $LP(T)$ is feasible)

Theorem: If $LP(T)$ is feasible & $\tilde{x}$ is a bfs, we can obtain an integral soln. x w/ makespan $2T$.

Claim 1: In $\tilde{x}$, at least $n-m$ jobs are assigned integrally.

Proof: At a bfs, at least mn constraints are tight.
Total # of constraints $= m+n + mn$.
Thus at most $m+n$ constraints of the type $x_{ij} \geq 0$ are not tight.
i.e., $\boxed{\text{at most } m+n \ \tilde{x}_{ij}\text{'s are NOT zero.}}$
Now say α jobs are set integrally, β fractionally
Then $\alpha + 2\beta \leq m+n$, $\alpha + \beta = n$
$\Rightarrow \beta \leq m$, so $\alpha \geq n-m$ ∎

Now consider the graph $G = (J \cup M, E)$
where $e = \{i, j\} \in E$ if $\tilde{x}_{ij} > 0$.

Claim 2: In every component of G, # edges $\leq$ # vertices.

Proof: Note that in $\tilde{x}$, # edges $= \# \tilde{x}_{ij}\text{'s} \neq 0$
$\leq m+n = $ # vertices.

Now consider a component C of G,
let $\tilde{x}_C$ be the soln. $\tilde{x}$ restricted to C.
& J_C, M_C are jobs & machines in C.
Then $\tilde{x}_C$ is a bfs for $LP(T)$ restricted to J_C, M_C also.
If $\tilde{x}_C$ is not a bfs, it is a convex combination of 2 feasible solutions, say $\tilde{x}_C = \lambda y_C + (1-\lambda) z_C$.
Then let $\tilde{x}_{-C}$ be solution on remaining jobs & machines.
Then $\tilde{x} = \lambda (y_C, \tilde{x}_{-C}) + (1-\lambda)(z_C, \tilde{x}_{-C})$. And $\tilde{x}$ cannot be a bfs. This is a contradiction.
So $\tilde{x}_C$ must be a bfs for the $LP(T)$ restricted to J_C, M_C. Hence, # non-zero variables $=$ # edges $\leq |J_C| + |M_C| = $ # vertices in component C ∎

So now fix a component C in $G = (J \cup M, E)$.

If $j \in J_C$ is assigned integrally, j has degree 1, & is hence a leaf. Remove node j & the incident edge.
Thus from the component C, an equal # of vertices & edges have been removed.
Thus (i) remaining # edges $\leq$ remaining # vertices
(ii) all remaining jobs are assigned fractionally
$\Rightarrow$ all remaining jobs have degree ≥ 2.

So far: assigned jobs assigned integrally in $\tilde{x}$.

A component w/ # edges $\leq$ # vertices is called a pseudo tree. $=$ a tree $+$ one extra edge.

Claim 3: There is a perfect matching in component C after removal of integrally assigned jobs.

Proof: Each job has degree ≥ 2, hence each leaf is a machine. Remove each leaf i w/ adjacent job node j, assigning job j to machine i.
Hence in this step each machine i gets one job j more than in $\tilde{x}$, and $\tilde{x}_{ij} > 0$ (hence $p_{ij} \leq T$)
After removal of all leaves, each node has degree ≥ 2.
In fact, can show that each node has degree $= 2$.
Thus we now have an even cycle (since G is bipartite, no odd cycles).
We just choose alternate edges in the cycle, thus assigning each machine i one extra job j s.t. $\tilde{x}_{ij} > 0$ (hence $p_{ij} \leq T$) ∎

Overall, our assignment:
- assigns jobs assigned integrally in $\tilde{x}$,
- assigns each machine i at most one more job j s.t. $\tilde{x}_{ij} > 0$ (thus $p_{ij} \leq T$)

hence, the makespan achieved is $2T$, proving the theorem ∎

Then our 2-approximation algorithm is straightforward.
Let OPT be the optimal integral makespan.
We use binary search to find the smallest T for which $LP(T)$ is feasible. Note that $T \leq OPT$.
Then use the theorem to obtain a $2T \leq 2\,OPT$ makespan integral assignment in poly time,