

Lecture 18: Factor 2 approximation for $(1|prec|\sum w_j C_j)$

July 9th, 2009

To recall, in the $(1|prec|\sum w_j C_j)$ problem, we are given n jobs, one machine, a digraph D which captures the precedence constraints; if $(j \rightarrow k)$ is an arc in D then job j must be completed before job k .

We start with the following lemma which will be crucial to obtain the linear programming relaxation for the above problem. The lemma, however, holds for any single machine problem (precedence constraints are not necessary). We need the following definitions.

Given a feasible schedule on a single machine, let C_1, C_2, \dots, C_n be the completion times of the n jobs. Given a subset S of jobs, let $p(S) := \sum_{j \in S} p_j$ and let $p^2(S) := \sum_{j \in S} p_j^2$. We also use the shorthand $p(S)^2$ to denote $(p(S))^2$.

Lemma 0.1. *The C_j 's satisfy the following inequality*

$$\sum_{j \in S} p_j C_j \geq \frac{1}{2}(p^2(S) + p(S)^2) \quad (1)$$

Proof. Assume the jobs are labelled so that $C_1 \leq C_2 \leq \dots \leq C_n$ and let $S \subseteq J$. For any job j , $C_j \geq \sum_{k \leq j} p_k$ because the jobs must be feasibly scheduled. So certainly, $C_j \geq \sum_{k \leq j, k \in S} p_k$ and $p_j C_j \geq p_j \sum_{k \leq j, k \in S} p_k$. Summing over all $j \in S$,

$$\sum_{j \in S} p_j C_j \geq \sum_{j \in S} p_j \sum_{k \leq j, k \in S} p_k = \frac{1}{2}(p^2(S) + p(S)^2)$$

□

We can now state an LP relaxation for $(1|prec|\sum w_j C_j)$. We have variables C_j to denote the completion time of job j . The LP relaxation is as follows

$$\begin{aligned} \min \quad & \sum w_j C_j \\ & C_k \geq C_j + p_k \quad \text{for each } j, k \text{ such that } j \rightarrow k \\ & \sum_{j \in S} p_j C_j \geq \frac{1}{2}(p^2(S) + p(S)^2) \quad \text{for each } S \subseteq J \\ & C_j \geq p_j \quad \text{for } k = 1, \dots, n \end{aligned}$$

Note the LP has a constraint for every subset of jobs, and therefore has exponentially many constraints. In the end of this section, we will tell how this LP can still be solved. For the time being assume we can solve the LP, and let C_j^L 's be the completion times returned by the LP. Note that we are not guaranteed that these are feasible completion times. However, these do imply the following algorithm.

Let the jobs be numbered $\{1, 2, \dots, n\}$ such that $C_1^L \leq \dots \leq C_n^L$. Schedule the jobs in this order.

Theorem 0.2. *The above algorithm is a 2-approximation for $1|prec|\sum w_j C_j$.*

Proof. Let C_j^A be completion time of job j in the above algorithm. We claim that $C_j^A \leq 2C_j^L$. This will prove the theorem.

To see this look at the set of jobs $S = \{1, 2, \dots, j\}$. Note that $C_j^A = \sum_{j \in S} p_j = p(S)$.

Now since $C_1^L \leq \dots \leq C_j^L$, we have

$$C_j^L p(S_j) \geq \sum_{k=1}^j p_k C_k^L \geq \frac{1}{2}(p^2(S_j) + p(S_j)^2)$$

where the last inequality follows from the LP constraint. The above implies, $C_j^L \geq \frac{1}{2}p(S_j) = \frac{1}{2}C_j^A$. \square