22C:253 Lecture 11

Changhui choi

October 21, 2002

Recall from last class that the integer program for Scheduling on Unrelated Parallel Machines (SUPM) is

$$\min t$$

subject to

$$\begin{aligned} t &\geq \sum_{j \in J} x_{ij} \text{ for each machine } i \in M \\ \sum_{i \in M} x_{ij} &= 1 \text{ for each job } j \in J \\ x_{ij} &\in \{0,1\} \text{ for all } i, j \end{aligned}$$

Recall that the difference between MINIMUM MAKESPAN and SUPM is that in SUPM the processing time of each task is machine dependant. The LP relaxation is obtained by replacing the constraint $x_{ij} \in \{0,1\}$ by $x_{ij} \geq 0$ for all $i \in M, j \in J$. We had two observations from last class:

1. The integrality gap of the above IP, LP relaxation pair is $\geq M$.
2. In any feasible solution (x,t) of the IP, if $p_{ij} > t$ then $x_{ij} = 0$.

However, item (2) above need not to be satisfied by a feasible solution of the LP. So we would like to add this constraint to the LP.

(C) if $p_{ij} > t$ then $x_{ij} = 0$ for each $i \in M, j \in J$

We will call the problem obtained by adding (C) to the LP relaxation, LP + (C). Let T^* be the makespan of an optimization solution of LP + (C), then

$$OPT_f \leq T^* \leq OPT.$$

Here, OPT_f is the cost of an optimal solution of the LP relaxation. So T^* is a better lower bound on OPT. But (C) is not a linear constraint. So to obtain T^* we use a technique called *parametic pruning*.

For any T, define $S_T = \{(i,j) \mid i \in M, j \in J, p_{ij} \leq T\}$. If (x,T) is a feasible solution of LP + (C) then $x_{ij} = 0$ for all $(i,j) \notin S_T$. This means that (x,T) satisfies the following constraints:

$$\sum_{j:(i,j)\in S_T} x_{ij} \cdot p_{ij} \leq T \text{ for all } i \in M \quad (1)$$

$$\sum_{i:(i,j)\in S_T} x_{ij} = 1 \text{ for all } j \in J \quad (2)$$

$$x_{ij} \geq 0 \text{ for all } (i,j) \in S_T \quad (3)$$

Conversely, if x is a feasible solution of the above set of constraints, then (x,T) is a feasible solution to LP + (C).

Now, how do we compute T^*? Suppose we know that T^* is in some range $[LB, UB]$. We simply need to find a smallest $T \in [LB, UB]$ such (x, T) is a feasible solution to set of constraints. We do this by binary search. Specifically, we start with an initial T being the midpoint of the range $[LB, UB]$. We then check of constraints (1-3) defined with respect to T has a feasible solution x. Without loss of generality, we can assume that if constraints (1-3) have a feasible solution x, then x is an extreme point of the set of feasible solutions. If constraints (1-3) do have a feasible solution x, we try a smaller value of T, else we try a larger value of T.

In this manner we compute a pair (x^*, T^*) an optimal solution to LP + (C) and we assume. Note that x^* satisfies the property that $x^*_{ij} = 0$ for all $i \in M, j \in J : p_{ij} > T^*$. It is possible that for some pairs (i, j), x^*_{ij} may be fractional and we show how to round these.

Let $r = |S_T|$. This means that x^* is a vertex of an r-dimensional polytope. Hence x^* is the intersection of some r hyperplanes, from among those defined by

$$\begin{aligned}
\sum_{j:(i,j)\in S_T} x_{ij}P_{ij} &= \quad T^* \text{ for all } i \in M && (4)\\
\sum_{i:(i,j)\in S_{T^*}} x_{ij} &=i \quad 1 \text{ for all } j \in J && (5)\\
x_{ij} &= \quad 0 \text{ for all } (i,j) \in S_{T^*} && (6)
\end{aligned}$$

At most $(n + m)$ of these came from the first two sets of hyperplanes. Hence $r - (n + m)$ of them have to be from the set (6). These $r - (n + m)$ constraints make $r - (n + m)$ of the r x_{ij} variables 0. These means that at most $(n + m)$ of the x_{ij} variables can be non-zero.

Let a job $j \in J$ be integral if $x^*_{ij} = 1$ for some $i \in M$. Let a job $j \in J$ be fractional if $x^*_{ij} \in (0, 1)$ for some $i \in M$. How many fractional jobs can there be? Let a be the number of integral jobs and b be the number of fractional jobs. So we have that $a + b = n$. We also know that in addition to the a x_{ij} variables that are set to 1 because of the a integral jobs, there are at least $2b$ x_{ij} variables that are set to positive fractional values. Hence, we also have $a + 2b \le (n + m)$. Solving for a and b, we get that $b \le m$. In other words, there are at most m fractional jobs. If we could take each fractional job j and assign of it to a distinct machine i with $x^*_{ij} > 0$, we are done. This is because for any $x^*_{ij} \in (0, 1)$, we have $p_{ij} \le T^*$. Therefore, if we could find a matching of jobs to machines, our makespan increases at most by T^*. Before these fractional jobs are assigned, the makespan is T^*. After the fractional jobs are assigned, the makespan $\le 2T^* \le 2OPT$.