

CSC 2420F 2010, Assignment 1
Due: October 6 at start of class

1. Consider the following local search schema for the makespan problem (on m identical machines).
While there is a job defining the makespan that can be moved so as to decrease the number of machines achieving the current makespan (and in particular, if such a move decreases the makespan), move that job. (A job defines the makespan if it completes at the makespan time.)
 - (a) Show that the approximation ratio (for all inputs) is at most $2 - \frac{1}{m}$ no matter how we choose to which machine a job is moved and no matter what the initial schedule is. Show that this ratio is tight for this local search variant.
 - (b) Complete a description of the local search algorithm by specifying that the job is always moved to the least loaded machine (breaking ties arbitrarily). Show that the algorithm terminates in n iterations.
2. Bonus question: Consider the following local search algorithm:
While there is any job that can be moved so as to decrease the number of machines achieving the current makespan, move that job.
 - (a) Show that this local search variant achieves a $2 - \frac{2}{m+1}$ approximation and show that this bound is tight.
 - (b) What is the best termination bound you can derive using any tie breaking rules you wish to define?
3. Consider the PTAS solution of the makespan problem on m identical machines. Suppose that we have been able to schedule all the large jobs (i.e. the ones with $p_j > T/s$) so that the makespan is at most $(1 + 1/s) * T$ when T was the desired target (i.e. if we know $OPT > T$ then we can report “failure”). Show that we can complete the $(1 + 1/s)$ -approximation algorithm by greedily filling in the small jobs without exceeding makespan $(1 + 1/s) * T$.

4. Consider the following scheduling problem. We have one machine and we are given n jobs $\mathcal{I} = \{J_1, \dots, J_n\}$ where a job J_i is described by a triple (p_i, d_i, v_i) where p_i is the processing time, d_i is the deadline and v_i is the value of job J_i . Consider all input parameters to be positive rational numbers. One classical problem is to maximize the value of a feasible schedule (i.e. the sum of the values of scheduled jobs) where a feasible schedule is defined by a scheduling function $\sigma : \{1, \dots, n\} \rightarrow \mathbb{R} \cup \{\infty\}$ such that:

- If $\sigma(i) = t < \infty$, we have $\sigma(i) + p_i \leq d_i$
- for all $i \neq j$ if $\sigma(i) = t_i$, $\sigma(j) = t_j$ and $t_i \leq t_j < \infty$, then $t_i + p_i \leq t_j$.

That is, scheduled jobs must end by their deadlines and cannot intersect other scheduled jobs. Here $\sigma(i) < \infty$ represents the start time of a scheduled job. The objective is to maximize the sum of v_i , summed over those jobs for which $\sigma(i) < \infty$.

- (a) Suppose there is a feasible schedule for $\mathcal{S} \subseteq \mathcal{I}$. Show that the schedule can be made to satisfy $\sigma(i) < \sigma(j)$ implies $d_i \leq d_j$. That is, in any feasible schedule we can assume that jobs are scheduled in order of their deadlines.
- (b) Formalize this scheduling optimization problem as an IP.
- (c) Suppose $v_i = c_i \cdot v$ for some fixed v and $c_i \in \{1, \dots, 10\}$. (Hint: 10 could be any small number.) Use dynamic programming DP to derive a polynomial time algorithm which will determine whether or not there exists a feasible schedule having value V and if so will find such a schedule. Assuming you state a DP recursively, state informally why the definition is correct.
Hint: Consider the DP used in the makespan problem.

5. Consider the restricted machines model makespan problem when all processing times are unit (i.e. $p_j = 1$ for all jobs). Show how to use a flow algorithm to optimally solve this problem.