Homework 5

Due: July 21st, 2009

(Total 30)

1. (6 marks)
   State if the following statements are true or false. If true, prove it. If false, give a counter-example. Answers without reasons will be marked 0.

   - **(3)** If the number of jobs $n < 2m$, then LPT returns an optimal solution to $(P||C_{max})$.
   - **(3)** If the number of jobs $n > km$, then LPT returns a $\frac{k+1}{k}$ factor approximation algorithm. (Be careful about the assumption we made in our analysis of LPT about the last job being the minimum processing time jobs. Go over why we made the assumption and wonder if it can be assumed even in this case.)

2. (8 marks)
   Consider the following generalization of list-scheduling for minimizing makespan in unrelated machines $(R||C_{max})$.

   Let $J = \{1, 2, \ldots, n\}$ be an arbitrary ordering of the jobs and process the jobs in this order. When job j is being processed, schedule it on the machine on which it will finish the first, that is, the one which minimizes C_j. Thus, if $\text{load}_j(i)$ is the sum of processing times of all jobs from $\{1, \ldots, j\}$ assigned to machine i, then schedule job j on the machine which minimizes

   $$\text{load}_j(i) + p_{ij}$$

   Note that $C_{max} = \max_i \text{load}_n(i)$.

   - **(2)** Run the above algorithm on the following instance of $(R||C_{max})$. The entry in the ith row and jth column is p_{ij}. Assume the order is $(1, 2, 3, 4)$.

	Job 1	Job 2	Job 3	Job 4
M/c 1	1	2	4	4
M/c 2	3	1	3	4
M/c 3	2	3	4	3

   (2) Does this return an optimal schedule? Is there any order such that running the above algorithm on that order will lead to an optimal schedule?

   - **(4)** Show that for any constant $B > 1$, there is an example of an instance and order of jobs such that the above algorithm returns a schedule with makespan $C_{max} \geq B \cdot OPT$. That is, the above algorithm is not a constant factor algorithm. *(Hint: Try $B = 3$. Try to generalize.)*

   Bonus (4) Come up with an example such that $C_{max} \geq B \cdot OPT$ no matter what the order is.

3. (10 marks)
   In this question, we will develop a $(2 - \frac{1}{m})$-algorithm for makespan minimization in uniformly related machines $(Q||C_{max})$. Recall, in this problem, jobs have processing time $(p_1, \ldots, p_n)$ and machines have speeds $(s_1, \ldots, s_m)$, and the time taken by machine i to process job j is $\frac{p_j}{s_i}$. The goal is to minimize the completion time of the last job.

   Suppose we order the machines in decreasing order of speeds: $s_1 \geq s_2 \geq \cdots \geq s_m$. Consider the following extension of the longest processing time rule:

Order the jobs in decreasing processing time order. $p_1 \geq p_2 \geq \cdots \geq p_n = p_{min}$. In this order, schedule job j on the machine in which it will finish the first. Thus, if $\text{load}_j(i)$ is the sum of processing times of all jobs from $\{1, \ldots, j-1\}$ assigned to machine i, then schedule job j on the machine which minimizes

$$\text{load}_j(i) + \frac{p_j}{s_i}$$

(a) **(1)** Suppose the optimum schedule doesn't assign any job to machine i. Argue that we can assume that the optimum schedule doesn't assign any jobs to machines $i+1, i+2, \ldots, m$, either.

(b) **(1+2)** *(Lower Bounds)*
Following part (a), suppose the optimum schedule assigns jobs to machines $(1, 2, \ldots, k)$. Show that

- $OPT \geq \frac{p_{min}}{s_k}$
- $OPT \geq \frac{\sum_{j=1}^{n} p_j}{\sum_{i=1}^{k} s_i}$

Note that these generalize the lower bounds which we had for $(P||C_{max})$.

Hint: Suppose the set of jobs scheduled by the optimum schedule on machine i is J_i, for $i = 1 \ldots k$. Define $p(J_i) := \sum_{j \in J_i} p_j$ to be the total processing times of jobs in J_i. What is OPT in terms of $(p(J_i)/s_i)$'s ? The following fact might be handy:

$$\text{If } x \geq \tfrac{a}{b} \text{ and } x \geq \tfrac{c}{d}, \text{ then } x \geq \tfrac{a+c}{b+d}.$$

(c) **(1)** As done for the analysis of LPT, argue that we can assume the last job to finish is n, the job with the minimum processing time.

(d) **(2)** Let ℓ be the machine which processes job n. Let t_n be the time when job n was started on machine ℓ. Let $\text{load}(i)$ be the total time taken by jobs in $\{1, \ldots, n-1\}$ on machine i. Show, using the definition of the algorithm,

$$\text{load}(i) \geq t_n + \frac{p_n}{s_\ell} - \frac{p_n}{s_i}$$

(e) **(3)** Show, using the upper bounds of part (b) and using part (d), that the completion time of job n,

$$C_n = t_n + \frac{p_n}{s_\ell} \leq (2 - \frac{1}{m})OPT$$

Hint: Use the fact that the sum of processing times of jobs assigned to machine i equals $s_i \times \text{load}(i)$. Then look at the sum of processing times of jobs assigned to machine 1 to k and lower bound this sum using part (d).

4. (6 marks)
Consider the following *graph balancing* problem. We are given a undirected graph $G = (V, E)$ with weights $w(e)$ on the edges. *Orienting* an edge (u, v) of the graph is to make the edge directed, either as $(u \to v)$ or $(v \to u)$. A complete *orientation* of the edges in E leads to a digraph $D = (V, A)$ where A are the oriented arcs.

The objective is to find an orientation which minimizes the maximum *weighted in-degree* of a vertex. That is, given a vertex v, let E_v be the set of edges incident on v which have been oriented towards v. Then, the goal is to minimize

$$\max_{v \in V} \sum_{e \in E_v} w(e)$$

Cast this problem as a problem of minimizing makespan in unrelated machines $(R||C_{max})$, and therefore argue that there is a 2-approximation for the problem.

BONUS (6) If all the weights $w(e)$ are 1, give a polynomial time algorithm for the problem.