Due date: February 19, 2009

Note: in each problem set there will be somewhat harder problems marked by (*)'s; while these are optional, it is anticipated that you will attempt a few throughout the semester.

4. Consider the problem 1||f_{\text{max}} , when each \( f_j \) is of the form
   \[ \max\{w_j L_j, w_j E_j\} , \]
   where the earliness \( E_j = d_j - C_j \). Show that the decision version of this problem is NP-complete. Can you extend your proof to show that it is strongly NP-complete?

5. Consider the on-line model in which, for each time \( t > 0 \), the schedule up until time \( t \) must be constructed without knowledge of those jobs that are released at time \( t \) or after. (When a job is released, we do know its processing time and its delivery time/due date.)
   (a) Show that no such on-line algorithm can be guaranteed to find an optimal solution to 1\mid r_j, p_j = p \mid L_{\text{max}} .
   (b) (*) Prove that no on-line \( \rho \)-approximation algorithm exists for as large a value of \( \rho \) as you can.

6. A fully polynomial approximation scheme is a polynomial approximation scheme with the property that not only is \( A_k \) a \( (1 + 1/k) \)-approximation algorithm, it runs in time polynomial in the size of the input and \( 1/k \). Using only the strong NP-completeness of 1\mid r_j, d_j < 0 \mid L_{\text{max}} , prove that the existence of a fully polynomial approximation scheme for this problem implies that \( P = NP \).

7. For 1\mid r_j, d_j < 0 \mid L_{\text{max}} , consider the approximation algorithm that whenever a job completes, if jobs are available, chooses the one for which \( p_j + q_j \) is largest, or if not, waits until the next job is released, and then performs the same selection rule. Show that the difference between the objective function value found and the optimum is at most \( p_a + p_b - p_c \) for some set of jobs \( \{a, b, c\} \).

9. (Research question) Give an LP-based proof of the min-max theorem for $1|r_j,pmtn|L_{\text{max}}$. 