Approximation Algorithms (MA5517)

Dr. Jannik Matuschke | M.Sc. Marcus Kaiser

Problem Set 3

Relaxed Decision Procedures and a FPTAS

Problem 3.1 (Related Parallel Machine Scheduling)
In this problem, we consider a variant of the problem of scheduling on parallel machines so as to minimize the length of the schedule. Now each machine $i \in [m]$ has an associated speed $s_i > 0$, and it takes p_j/s_i units of time to process job $j \in [n]$ of length $p_j \geq 0$ on machine $i \in [m]$. We call these related machines. Assume that machines are ordered such that $s_1 \geq s_2 \geq \ldots \geq s_m$.

a) A ρ-relaxed decision procedure for a scheduling problem is an algorithm such that given an instance of the scheduling problem and a deadline D either produces a schedule of length at most ρD or correctly states that no schedule of length D is possible for the instance. Show that given a polynomial-time ρ-relaxed decision procedure for the problem of scheduling related machines, one can produce a ρ-approximation algorithm for the problem.

b) Consider the following variant of the list scheduling algorithm, now for related machines. First, the list of jobs is sorted with respect to their processing times in non-increasing order, i.e. $p_1 \geq p_2 \geq \ldots \geq p_n > 0$. Given a deadline D, anytime a machine $i \in [m]$ becomes idle before time D it processes the first job $j \in [n]$ in the list that has not yet been processed and satisfies $p_j/s_i \leq D$. If no such job is available or machine i becomes idle at a time D or later, it stops processing. In the case that not all jobs are processed by this procedure, the algorithm states that no schedule of length D exists. Prove that this algorithm is a polynomial-time 2-relaxed decision procedure.

Problem 3.2 (Unrelated Parallel Machine Scheduling)
Suppose we are given a set of n jobs to be assigned to m machines. Each job $j \in [n]$ is to be scheduled on exactly one machine. If job $j \in [n]$ is scheduled on machine $i \in [m]$, then it requires $p_{i,j} \geq 0$ units of processing time. The goal is to find a schedule of minimum length, which is equivalent to finding an assignment of jobs to machines that minimizes the maximum total processing time required by a machine. This problem is sometimes called scheduling unrelated parallel machines so as to minimize the makespan.

We show that deterministic rounding of a linear program can be used to develop a polynomial-time 2-relaxed decision procedure. Consider the following set of linear inequalities for a parameter D:

$$
\begin{aligned}
\sum_{i=1}^{m} x_{i,j} &= 1 && \forall j \in [n] \\
\sum_{j=1}^{n} p_{i,j} x_{i,j} &\leq D && \forall j \in [m] \\
x_{i,j} &\geq 0 && \forall i \in [n], j \in [m] : p_{i,j} \leq D \\
x_{i,j} &= 0 && \forall i \in [n], j \in [m] : p_{i,j} > D
\end{aligned}
$$

Please turn over.

If a feasible solution exists, let x be a basic feasible solution for this set of linear inequalities. Consider the bipartite graph G on machine nodes $\mathcal{M} = \{M_1, \ldots, M_m\}$ and job nodes $\mathcal{N} = \{N_1, \ldots, N_n\}$ with edges (M_i, N_j) for each variable $x_{i,j} > 0$.

a) Prove that the linear inequalities are a relaxation of the problem, in the sense that if the length of the optimal schedule is D, then there is a feasible solution to the linear inequalities.

b) Prove that each connected component of G of k nodes has at most k edges.

c) If $x_{i,j} = 1$, assign job j to machine i. Once all of these jobs are assigned, use the structure of the previous part to show that it is possible to assign at most one additional job to any machine. Argue that this results in a schedule of length at most $2D$.

d) Use the previous parts to give a polynomial-time 2-relaxed decision procedure, and conclude that there is a polynomial-time 2-approximation algorithm for scheduling unrelated parallel machines to minimize the makespan.

Problem 3.3 (Constrained Shortest Path Problem)
Suppose we are given a directed acyclic graph $G = (V, A)$ with specified source node $s \in V$ and sink node $t \in V$, and each arc $a \in A$ has an associated cost $c_a \geq 0$ and length $l_a \geq 0$. We are also given a length bound $L \geq 0$. Give a fully polynomial-time approximation scheme for the problem of finding a minimum-cost path from s to t of total length at most L.

This problem set will be discussed in the tutorial on December 6th, 2016.