Albert-Ludwigs-Universität Freiburg SS 2008
Institut für Informatik
Prof. Dr. Susanne Albers
Dr. Alexander Souza

Combinatorial Optimization

Exercise 1 (Flow Decomposition)

Prove the following *flow decomposition* result.

Theorem 1. *Given a network $N = (G, c, s, t)$ and an $s-t$-flow f then there is a familiy $\mathcal{P}$ of simple paths, a famililiy $\mathcal{C}$ of simple cycles and positive numbers $h : \mathcal{P} \cup \mathcal{C} \to \mathbb{R}^+$ such that $\mathrm{val}(f) = \sum_{T \in \mathcal{P} \cup \mathcal{C}} h(T)$ and $|\mathcal{P}| + |\mathcal{C}| \leq |A|$.*

Hint. Proceed by induction on the number of edges with non-zero flow.

Exercise 2 (Unbalanced Assignments)

In the lecture we assumed that the bipartite graph $G = (L \cup R, E)$ of the assignment problem satisfies $|L| = |R|$. Show that this assumption can be made without loss of generality. That means, for every instance with a bipartite graph $G' = (L' \cup R', E')$ with $|L'| \leq |R'|$ and corresponding weight function construct an equivalent instance with a graph G as above.

Exercise 3 (Scheduling on Parallel Machines)

In the Scheduling on Parallel Machines problem we are given m machines and n jobs, where job j takes time p_{ij} if assigned to machine i. The jobs assigned to any machine are scheduled in a certain order. The completion time of job j is denoted c_j and refers to the following: If job j is assigned to machine i then the times p_{ik} of the jobs k also assigned to machine i but scheduled before job j contribute to the completion time of j, i.e., $c_j = \sum_{k \text{ on } i,\, k \leq j} p_{ik}$ (where "$k \leq j$" means that job k is scheduled before job j). The objective is to minimize the total completion time $\sum_j c_j$.
The goal of this exercise is to show that this problem can be formulated as an Assignment problem.

(a) Let there be one machine only and let the n jobs J be ordered such that $p_1 \leq p_2 \leq \ldots p_n$. Show that it is optimal for $\sum_{j \in J} c_j$ to schedule the jobs in this ordering. *Hint.* Use an exchange argument.

    Further prove that we can write $\sum_{j \in J} c_j = \sum_{j=1}^{n} (n - j + 1) p_j$.

(b) We have thus reduced the problem of finding an optimal schedule (i.e. an assignment of jobs to machines and their ordering) to that of finding an optimal assignment only in the following sense: Once the assignment of any job j to a machine i is fixed we schedule according to non-decreasing p_{ij}-values. Use the above properties to construct an appropriate network for an Assignment problem that solves the scheduling problem with $m \geq 1$ machines.