Handout 2: Greedy algorithms

Uriel Feige

28 Nov 2018

Next class, on December 5, will be given by Julia Chuzhoy, a visiting Professor from TTIC. (She will also give a seminar of Dec 3.)

Topics discussed in class on Nov 21 and 28:

Metric TSP. The double tree algorithm, the algorithm of Christofides.

k**-center**. The greedy algorithm, and hardness of approximation within ratio better than 2.

Scheduling jobs on identical parallel machines. Every greedy algorithm has ratio no worse than 2, but sorting first gives ratio no worse than $\frac{4}{3}$.

Submodular functions. A $1 - \frac{1}{e}$ greedy approximation for selecting k items of maximum value. (NP-hard to do better.)

Analysis of the greedy algorithm for **weighted set cover** via dual fitting.

Homework – hand in by December 12. (Grader: Yael Hitron. Recall conventions for homework from Handout 1.)

1. (Problem 2.1 in [WS11].) The k-facility problem is related to the k-center problem. The input is a positive integer k, a set F of facilities, a set D of costumers, and distances $d_{ij} > 0$ between any i and j in $F \cup D$. Distances are symmetric ($d_{ij} = d_{ji}$) and obey the triangle inequality. One needs to select a set S of k facilities, and then each client pays a connection cost that equals its distance to the nearest facility in S. The goal is to select S that minimizes the maximum connection cost over all clients, namely, $\text{opt} = \min_{S \subset F; |S|=k} \max_{j \in D} \min_{i \in S} d_{ij}$.

   (a) Give a polynomial time approximation algorithm for the k-facility problem, and prove that its approximation ratio is no worse than 3.

   (b) Prove that there is no polynomial time algorithm with an approximation ratio better than 3, unless P=NP.

2. (Problem 2.3 in [WS11].) There are n jobs and m identical machines (that can run in parallel). Each job i has an integer processing time $p_i > 0$. In addition there are arbitrarily many precedence constraints of the form $i \prec j$, meaning that job j cannot start processing before job i ends. The precedence constraints are consistent (they induce a partial order over the jobs). A *schedule* is an assignment of jobs to machines together with the processing intervals for each job: each job i receives an interval of

p_i consecutive time steps on one of the m machines, such that no two intervals on the same machine overlap, and the precedence constraints are satisfied. All machines start at time step 1, and the *makespan* of the schedule is the last time step in which a machine still processes a job. The goal is to find a schedule of smallest makespan. A *greedy* schedule is one that schedules each job at the first time step available to it (a time step t is available for a job j if all jobs preceding j in the precedence constraints have finished before time t, and some machine does not process any other job at time t), breaking ties arbitrarily. Prove that the makespan of every greedy schedule is at most twice that of the optimal schedule.

3. (Based on Problem 2.5 in [WS11].) In the *minimum-cost Steiner tree* problem the input is an undirected graph $G(V, E)$, nonnegative costs $c_{ij} > 0$ for all edges $(i, j) \in E$, and a subset $T \subset V$ of vertices designated as terminals. The goal is to find a tree of minimum cost (the cost of the tree is the sum of costs of edges in the tree) that contains all terminals. (The non-terminal vertices in the tree are referred to as *Steiner vertices.*) Denote this cost by $\text{opt}(G)$.

   Consider the following algorithm for approximating $\text{opt}(G)$ within a ratio of 2. Construct an auxiliary graph $G'(T, E')$ whose set of vertices is T, and for every two terminals $i, j \in T$ there is an edge $(i, j) \in E'$ of cost equal to the cost of the least costly path from i to j in G. (These edge costs can be computed in polynomial time.) Denote the cost of the minimum cost spanning tree in G' by $MST(G')$.

   (a) Explain why $MST(G') \geq \text{opt}(G)$, and give an example in which strict inequality holds.
   (b) Prove that $MST(G') \leq 2\text{opt}(G)$.

4. (Based on Problem 2.14 in [WS11].) In the *edge-disjoint paths* (EDP) problem in directed graphs the input is a directed graph $G(V, A)$ and a set $D = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ of source-sink pairs $s_i, t_i \in V$. The goal is to select a subset $S \subset D$ of maximum cardinality such that for every $(s_i, t_i) \in S$ we can select a directed path P_i from s_i to t_i, and no two selected paths share a directed edge.

   Consider the following greedy algorithm for EDP. It uses the well know fact that shortest paths can be computed in polynomial time. Among the source-sink pairs not yet in S, find the pair (s_i, t_i) with the currently shortest directed connecting path (if there is such a pair), add this pair to S, remove the arcs of the path from G, and repeat.

   (a) Show that the approximation ratio of this algorithm is at least $\frac{1}{1+\sqrt{m}}$, where $m = |A|$ is the number of arcs in $G(V, A)$.
   (b) Show that for every $k \geq 3$ there are instances of EDP with $|D| = k$ and $m \leq k^2$, in which the optimal solution is $S = D$, whereas the greedy algorithm described above selects only one pair from D.