Handout 3: Dynamic Programming, Local Search

Uriel Feige

12 Dec 2018

Topics discussed in class on Dec 5 (Julia Chuzhoy):

Notions such as PTAS, FPTAS, pseudo-polynomial time algorithms, and strong NP-hardness.

Using dynamic programming so as to obtain:

1. An FPTAS for knapsack.
2. A PTAS for minimizing makespan on a constant number of parallel identical machines.
3. A PTAS for minimizing makespan on an arbitrary number of parallel identical machines.

More details appear in [WS11].

Topics that we intend to discuss on Dec 12.

Use of local search techniques:

1. Max cut in graphs, via greedy algorithms and via local search. Approximation ratio $\frac{1}{2}$. Major open question: can one find a locally optimal cut in weighted graphs in polynomial time.
2. Matching in r-uniform hypergraphs. Factor $\frac{1}{r}$ by greedy algorithm, and $\frac{2}{r+1}$ by local search. Open question: is there a constant c and an approximation algorithm that for every r has an approximation ratio no worse than $\frac{4}{r+c}$?
3. Legal edge-coloring of graphs with the minimum number of colors. Solvable in polynomial time in bipartite graphs, but NP-hard in general. Vizing's theorem gives an additive approximation of 1.

Note: the first two topics are not discussed in [WS11].

Homework – hand in by December 26. (Grader: Yael Hitron. Recall conventions for homework from Handout 1.)

1. (Problem 3.6 in [WS11].) Suppose we are given a directed acyclic graph with specified source node s and sink node t, and each arc e has an associated cost $c_e > 0$ and length $\ell_e > 0$. We are also given a length bound L. 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.

2. (Problem 3.9 in [WS11].) Suppose we have a strongly NP-hard minimization problem Π with the following two properties. First, any feasible solution has a nonnegative, integral objective function value. Second, there is a polynomial p, such that if it takes n bits to encode the input instance I in unary, the value of the optimum solution for I is at most $p(n)$. Prove that if there is a fully polynomial-time approximation scheme for Π, then there is a pseudo-polynomial algorithm for Π. Since there is no pseudo-polynomial algorithm for a strongly NP-hard problem unless P=NP, conclude that this would imply P=NP.

3. Recall that the cut function in graphs with nonnegative edge weights can be represented as a nonnegative nonmonotone submodular function of the set of vertices, and that max-cut can be approximated within a constant factor. Consider the following greedy algorithm for maximizing general nonnegative nonmonotone submodular functions on n items. Starting from the empty set, at each step, if there is an item with positive marginal value (when added to the existing set), add to the existing set the item of largest marginal value (breaking ties arbitrarily). Stop when the *if* condition no longer holds.

   (a) Prove that this greedy algorithm has an approximation ratio no worse than $\frac{1}{n-1}$.

   (b) For every $\epsilon > 0$ and $n > 2$, design a nonnegative submodular function (and prove that it is nonnegative and submodular) on which the approximation ratio of the above greedy algorithm is not better than $\frac{1+\epsilon}{n-1}$.