CS612 Assignment 10

Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

December 28, 2009

Notice:

1. Due Jan. 14, 2010.
2. Please send your answer to wangchao1987@ict.ac.cn, shaomingfu@gmail.com, yuanxiongying@ict.ac.cn
3. You can arbitrarily choose one problem from Problems 1-3 and one from Problems 4-7.

1 Tree Decomposition(10 marks)

Suppose we are given a directed graph $G = (V,\ E)$, with $V = \{v_1, v_2, ..., v_n\}$, and we want to decide whether G has a Hamiltonian path from v_1 to v_n. Show that the Hamiltonian Path Problem can in fact be solved in time $O(2^n p(n))$, where $p(n)$ is a polynomial function of n.

2 Tree Decomposition(10 marks)

The Node-Disjoint Paths Problem is given by an undirected graph G and k pairs of nodes (s_i, t_i) for $i = 1, 2, ..., k$. The problem is to decide whether there are node-disjoint paths P_i so that path P_i connects s_i to t_i. Give a polynomial-time algorithm for the Node-Disjoint Paths Problem for the special case in which G has tree-width 2, and we are also given a tree decomposition T of G with width 2.

3 Tree Decomposition(10 marks)

Consider the class of 3-SAT instances in which each of the n variables occurs, counting positive and negated appearances combined, in exactly three clauses. Show that any such instance of 3-SAT is in fact satisfiable, and that a satisfying assignment can be found in polynomial time.

4 Local Search(10 marks)

Given a sequence g_i, $i = 1, 2, ..., k$, and for $j = 1, 2, ..., k$,

$$G(j) = \Sigma_{i=1}^{j} g_i$$

if $G(k) > 0$, show that there exists a circle permutation $g_r, g_{r+1}, ..., g_k, g_1, ..., g_{r-1}$, $1 \leq r \leq k$ such that each $G(j)$ of the new sequence is positive.

5 Local Search(10 marks)

Neighborhood N is exact, iff a feasible solution F is global optimal if F is local optimal. In the TSP problem, define k-interchange neighbor:
$N_k(f) = g{:}g \in F$ and g delete two edges from f and insert two new ones instead of them.
(a)Show that N_{n-3} is not a exact neighbor for TSP problem.
(b)Show that N_{n-1} is a exact neighbor.

6 Local Search(10 marks)

Consider the following Gradient Ascent Algorithm for finding a matching in a bipartite graph.
As long as there is an edge whose endpoints are unmatched, add it to the current matching. When there is no longer such an edge, terminate with a locally optimal matching.
(a)Give an example of a bipartite graph G for which this gradient ascent algorithm does not return the maximum matching.
(b)Let M and M' be matchings in a bipartite graph G. Suppose that $|M'| > 2|M|$. Show that there is an edge $e' \in M'$ such that $M \cup \{e'\}$ is a matching in G.
(c)Use (b) to conclude that any locally optimal matching returned by the gradient ascent algorithm in a bipartite graph G is at least $half$ as large as a maximum matching in G

7 Local Search(10 marks)

Suppose you are consulting for a biotech company that runs experiments on two expensive high-throughput assay machines, each identical, which we'll label M_1 and M_2. Each day they have a number of jobs that they need to do, and each job has to be assigned to one of the two machines. The problem they need help on is how to assign the jobs to machines to keep the loads balanced each day. The problem is stated as follows. There are n jobs, and each job j has a required processing time t_j. They need to partition

the jobs into two groups A and B, where set A is assigned to M_1 and set B to M_2. The time needed to process all of the jobs on the two machines is $T_1 = \Sigma_{j \in A} t_j$ and $T_2 = \Sigma_{j \in B} t_j$. The problem is to have the two machines work roughly for the same amounts of time, that is, to minimize $|T_1 - T_2|$.

A previous consultant showed that the problem is NP-hard. Now they are looking for a good local search algorithm. They propose the following. Start by assigning jobs to the two machines arbitratily(say jobs $1, ..., \frac{n}{2}$ to M_1, the rest to M_2). The local moves are to move a single job from one machine to the other, and we only move jobs if the move decreases the absolute difference int the processing time. You are hired to answer some basic questions about the performance of this algorithm.

(a)How good is the solution obtained? Assume that there is no single job that dominates all the processing time, that is, that $t_j \leq \frac{1}{2}\Sigma_{i=1}^{n} t_i$ for all jobs j. Prove that for every locally optimal solution, the times the two machines operate are roughly balanced: $\frac{1}{2}T_1 \leq T_2 \leq 2T_1$.

(b)Next you worry about the running time of the algorithm: How often will jobs be moved back and forth between the two machines? You propose the following small modification in the algorithm. If, in a local move, many different jobs can move from one machine to the other, then the algorithm should always move the job j with maximum t_j. Prove that, under this variant, each job will move at most once.

(c)Finally, they wonder if they should work on better algorithms. Give an example in which the local search algorithm above all not lead to an optimal solution.