TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TU Delft

Exam IN4301 Advanced Algorithms

January 28, 2013

- Use a separate sheet for each question.
- This is a closed book examination with 4 questions worth of 100 points in total.
- Your mark for this exam will be the number of points divided by 10.
- If you have at least a 5.0 for this exam, the average of the two programming exercises, as well as the average of the homework exercises, your final mark for this course is the average of these three marks, rounded to the nearest half of a whole number. That is, 9.7 is rounded to 9.5, and 5.8 is rounded to 6.
- Use of book, readers, notes, and slides is not allowed.
- Use of (graphical) calculators is not permitted.
- Specify your name, student number and degree program, and indicate the total number of submitted pages on the first page.
- Write clearly, use correct English, and avoid verbose explanations. Giving irrelevant information may lead to a reduction in your score.
  Notice that almost all questions can be answered in a few lines!
- This exam covers Chapters 10 and 11 of Kleinberg, J. and Tardos, E. (2005), *Algorithm Design*, all information on the slides of the course, and the set of papers as described in the study guide.
- The total number of pages of this exam is 3 (excluding this front page).

1. (a) (7 points) Consider the following problem. Given is an undirected graph $G = (V, E)$ and a nonnegative integer $k < |V|$. Can we transform G, by deleting or adding at most k edges, into a graph (V, E') that consists of a disjoint union of cliques?

Consider the following algorithm for this problem, which uses the fact that the answer to the problem is "yes" if and only if we can transform G into a graph (V, E') where there is no path $u, v, w \in V$ with $\{u, v\}, \{v, w\} \in E'$ and $\{u, w\} \notin E'$.

Edit (E, k):
if $k < 0$ **then**
  return *false*
if *there exist* $\{u, v\}, \{v, w\} \in E$ *for which* $\{u, w\} \notin E$ **then**
  return
    Edit($E \cup \{u, w\}, k - 1$) *or* Edit($E - \{u, v\}, k - 1$) *or* Edit($E - \{v, w\}, k - 1$)
else
  return *true*

- Derive a tight upper bound on the run time of this algorithm (explain your answer).
- Is this problem kernelizable? Explain why (not).

(b) (9 points) Given is one machine, a set S of jobs, each $j \in S$ with a length p_j and a weight w_j, and a directed acyclic graph of jobs representing precedence constraints. The problem is to find a schedule with completion time C_j for each job j obeying the precedence constraints, and with a minimum sum of weighted completion times, i.e., minimize $\sum_{j \in S} w_j C_j$.

The following function defines the value of the optimal solution for this scheduling problem: $\text{OPT}(\emptyset) = 0$ and otherwise $\text{OPT}(S) = \min_{j \in \text{LAST}(S)} \{\text{OPT}(S - \{j\}) + w_j p(S)\}$, where $\text{LAST}(S)$ is set of jobs in S that do not need to precede others, and $p(S) = \sum_{i \in S} p_i$.

- Give an analysis of a tight upper bound on the run time of a recursive implementation of this algorithm without memoization.
- Give an analysis of a tight upper bound on the run time of an iterative implementation of this algorithm using dynamic programming.
- Explain in which order the values should be computed in the iterative implementation.

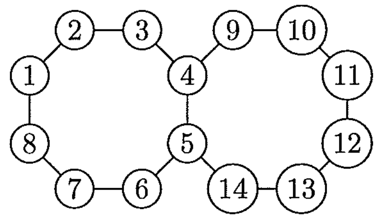

Figure 1: Graph G

(c) (9 points) Consider the graph in Figure 1. A student claims the following sets form a tree decomposition: $\{4, 5, 6\}$, $\{4, 6, 7\}$, $\{4, 7, 8\}$, $\{1, 4, 8\}$, $\{1, 2, 4\}$, $\{2, 3, 4\}$, $\{4, 5\}$, $\{4, 9, 10\}$, $\{4, 10, 11\}$, $\{4, 11, 12\}$, $\{4, 12, 13\}$, $\{4, 13, 14\}$, and $\{4, 5, 14\}$.

- Explain why this is (not) a tree decomposition: mention all relevant conditions.
- Repair if necessary and draw the resulting tree.
- Explain how to obtain the width of the resulting tree decomposition. Is this equal to the tree width? Why?

2. (a) (6 points) Suppose that we have a maximization problem Π. There is an algorithm A for Π such that A applied to an instance x of A never returns an answer that is more than three times as small as the the optimal value $OPT(x)$ of such an instance x. Moreover, there is at least one instance y for which A returns an answer that is exactly 3 times as small as $OPT(y)$. What can be concluded with respect to the performance ratio of A for a given instance x of Π, the approximation ratio for A, and the approximation threshold for Π?

   (b) (7 points) Consider the following problem Π:

   Input: a labeled directed graph $G = (V, A, w)$, where $w : A \to \mathbb{N}$ is a function assigning a positive weight $w(a)$ to edge $a \in A$, and two nodes v and w in V,

   Question: find a path $\pi_{v,w}$ from v to w containing every node in V, such that the sum of the weights $w(a)$ of all edges a on the path $\pi_{v,w}$ is minimal.

   Argue whether or not there exists a constant $c < \infty$ and an approximation algorithm A for Π such that the approximation ratio of A is c. (Hint: consider TSP-opt.)

   (c) (4 points) Present the 2-approximation algorithm for MinCover as discussed in class.

   (d) (8 points) Consider the following problem:

   Input: An undirected graph $G = (V, E)$ and a positive integer r.

   Question: Find a set of r vertices $S \subseteq V$, $|S| = r$, that cover the largest possible number of edges.

   Here, the set of edges covered by a set of vertices S is $\{(u, v) \in E \mid S \cap \{u, v\} \neq \emptyset\}$.

   Prove or disprove the following claim: the MinCover 2-approximation algorithm is also a 2-approximation algorithm for the problem stated above.

3. (a) (5 points) Given is an undirected graph $G = (V, E)$. Let a vertex cover of G be defined as a subset of the vertices $C \subseteq V$ such that all edges $\{u, v\} \in E$ satisfy $\{u, v\} \cap C \neq \emptyset$. Each vertex $v \in V$ has a "cost" c_v. Formulate the problem of determining the vertex cover that minimizes the sum of the costs of the vertices in the cover as an integer linear optimization problem.

   (b) (5 points) Formulate the dual problem corresponding to the linear relaxation (LP-relaxation) of the vertex cost cover problem given in the previous question. If you are unable to formulate the problem in the previous question, then you may formulate the dual of the following knapsack problem for a maximum of 3 points:

$$\begin{aligned} \max \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} a_j x_j &\leq b, \\ x_j &\geq 0 \qquad j = 1, \ldots, n. \end{aligned}$$

   (c) (5 points) Consider an integer optimization problem P and its LP-relaxation $LP(P)$. Problem P is a minimization problem. Suppose we have a specific instance of P, and that for this instance the optimal solution to the integer problem is equal to 3 and the optimal solution to the linear relaxation is equal to $3/2$. Given this information, give a bound on the performance guarantee of any approximation algorithm that is based on rounding a solution to the linear relaxation. Do not forget to motivate your answer.

(d) (10 points) Given an undirected graph $G = (V, E)$ with nonnegative weights w_{ij} on the edges $\{i, j\} \in E$, the maximum cut problem is the problem of determining a cut (a partition of the vertices $(W, V \setminus W)$) in G such that the sum of the weights of the edges $\{i, j\}$, such that $i \in W$ and $j \in V \setminus W$ is maximized. This problem can be formulated as a quadratic integer optimization problem (QIP). Goemans and Williamson formulated a relaxation (RP) of this QIP:

$$(RP): \quad \max \frac{1}{2} \sum_{i<j}^{n} w_{ij}(1 - v_i v_j)$$
$$\text{s.t. } v_i \in R^n, \ \|v_i\| = 1, \qquad i \in V.$$

Formulate a semidefinite optimization problem that is quivalent to (RP). Explain why the semidefinite problem is equivalent to (RP) and how the solution to (RP) can be retrieved from the solution to the semidefinite problem.

4. In the last three lectures, (optimal mixing) evolutionary algorithms and evolutionary local search as well as the (re)design of these algorithms for multi-objective optimization were discussed.
   (a) (6 points) What is a family of subsets (FOS) and how can it be used in a model-based evolutionary algorithm to perform black-box optimization?
   (b) (6 points) Describe the main differences between the Genetic Algorithm (GA), the Estimation-of-Distribution Algorithm (EDA) and the Optimal Mixing Evolutionary Algorithm (OMEA).
   (c) (6 points) Give definitions for the following concepts:
      - Pareto dominance (for solutions),
      - Pareto optimal (for solutions),
      - Pareto optimal set, and
      - Pareto optimal front.
   (d) (7 points) Within the context of multi-objective optimization, explain the concepts of Utopian solution and its weighted Tchebycheff distance to another solution and how this can (in theory) be used to solve a multi-objective optimization problem.

End of test