Combinatorial Optimisation, Exam 21 March 2011

The examination lasts 2 hours and 45 minutes. Grading will be done before April 2, 2011. Students interested in checking their results can make an appointment by e-mail.

The examination consists of four exercises. The maximum number of points to be gained on the various parts are displayed in the following table:

1a	1b	2a	2b	2c	3a	3b	4
5	5	5	5	5	5	10	10

In Questions 1b and 3a the number of points will never be less than 0.

The result is obtained by dividing the total number of points by 5. This implies that 28 points are needed to pass.

During the examination only the book *Combinatorial Optimization* by Papadimtriou and Steiglitz without any additional leaflets is allowed to be on your desk and all electronic equipment should be switched off.

1a. Prove the following theorem (Theorem 1):

Theorem 1. Every spanning tree of a graph has exactly $n-1$ edges, where n is the number of vertices of the graph.

Hint: You may use Theorem 2 below in a proof by induction. But any other correct proof is also fine.

Theorem 2. Every spanning tree of a graph has at least one leaf (vertex with degree 1).

Answer 1a. *More proofs than the one below can be correct.*

Proof. The statement is clearly true for $n = 1$. Let it be true for any graph with n vertices, then we will prove it for a graph G with $n+1$ vertices. Take any spanning tree T. By Theorem 2 there must be a vertex, v say, in T that has degree 1, by the edge $\{v, w\}$ say. Delete v and $\{v, w\}$ from T. What remains is a spanning tree on the graph with v deleted from it, since deleting v and $\{v, w\}$ does not destroy any path in T between any two other vertices of G. By induction this spanning tree has exactly $n-1$ edges. Hence T has $n-1+1 = n$ edges. □

1b. Decide for each of the following statements if it is "true" or "false" without giving any explanation. A correct answer gives 1 point, a mistake gives -1 point.

(a) There exist graphs with all vertices having different degrees.

(b) A graph with all of its vertices having degree at least k contains a path of length k.

(c) If in a network none of the capacities of the arcs have integer values, then the maximum flow from s to t in the network cannot have integer value.

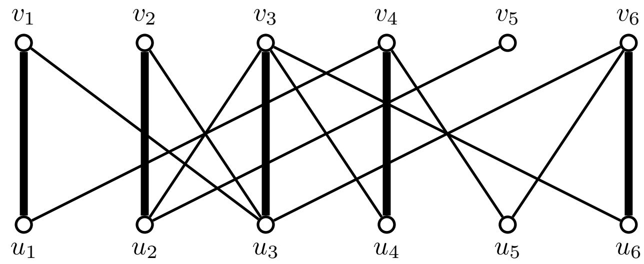

Figure 1: Graph instance of Question 2b. Matching edges are indicated in bold.

(d) If in a network none of the capacities of the arcs have integer values, then it can happen that the value of a maximum flow is not equal to that of a minimum cut.

(e) Given a weighted undirected graph, no two edges of which have equal weight. Any such a graph has a unique minimum spanning tree.

Answer 1b.

(a) False

(b) True

(c) False

(d) False

(e) True

2a. Consider any graph $G = (V, E)$. Make a *subdivision* of each of the edges: i.e., for every edge $e = \{u, v\} \in E$ create an extra vertex v_e and replace the edge $\{u, v\}$ by the two edges $\{u, v_e\}$ and $\{v_e, v\}$. (You may think of it as every edge of G getting an extra vertex in the middle.) Prove that the resulting graph is bipartite.

Answer 2a. *More proofs can be correct than the one below.*

Proof. Take any cycle of the new graph (call this graph G'). This is of the form $\{v_1, v_{\{v_1,v_2\}}, v_2, \ldots, v_k, v_{\{v_1,v_2\}}, v_1\}$ hence of even length. Since a bipartite graph is a graph without odd cycles G' must be bipartite. □

2b. Given is an *unweighted* bipartite graph $G = (V, E)$ displayed in Figure 1. And given is a matching displayed as boldface edges in Figure 1. Starting from this matching determine a maximum matching by using the augmenting path method or conclude that the current matching is optimal.

2c. Theorem 3. In a bipartite graph, the size of a maximum cardinality matching is equal to the size of a minimum cardinality vertex cover.

Prove this theorem.

Answer 2c. *More proofs can be correct than the one below.*

Proof. Let μ^* be the size of a maximum cardinality matching and τ^* the size of a minimum vertex cover. Given any matching at least one vertex of each of its edges must be in the vertex cover. Hence $\tau^* \geq \mu^*$.

To prove that $\tau^* \leq \mu^*$, take the maximum perfect matching M. Start the search tree for augmenting paths from all exposed vertices of V, w.r.t. M. Clearly, this does not reach any exposed vertex of U w.r.t. M. Select all the U-vertices in this search tree, which are all incident to edges of M. Similarly, select all the V-vertices in the search tree from all exposed vertices of U w.r.t. M. In this way we have selected $|M| = \mu^*$ vertices. We claim that the selected vertices form a vertex cover. Suppose not, then there exists an edge $\{u, v\}$ such that neither u nor v is in the cover. None of the two can be an exposed node, otherwise the other one would be in the cover. But then either u is reachable by an alternating path from an exposed node of V or v is reachable from an exposed node of U, hence one of the two must be in the cover. □

3a. Decide for each of the following statements if it is "true" or "false" without giving any explanation. A correct answer gives 1 point, a mistake gives −1 point.

(a) If $\Pi_1 \in NP$ and for every $\Pi_2 \in NP$, $\Pi_2 \preceq \Pi_1$, then Π_1 is *NP*-complete.

(b) If $\Pi_1 \in P$ and for every $\Pi_2 \in NP$, $\Pi_2 \preceq \Pi_1$, then $P = NP$.

(c) If $\Pi_1 \preceq \Pi_2$ and $\Pi_2 \preceq \Pi_3$, then $\Pi_1 \preceq \Pi_3$.

(d) If $\Pi_1 \in NP$, Π_2 is *NP*-complete, $\Pi_1 \preceq \Pi_2$, then Π_1 is *NP*-complete.

(e) If Π_1, Π_2 are *NP*-complete, then $\Pi_1 \preceq \Pi_2$ and $\Pi_2 \preceq \Pi_1$.

Answer 3a.

(a) True

(b) True

(c) True

(d) False

(e) True

3b. Prove that the decision version of the VEHICLE ROUTING PROBLEM is *NP*-complete.

VEHICLE ROUTING-DECISION:

Instance: A set of points X and one central depot c, together with the distance $d(x, y)$ between each pair $x \in X$ and $y \in X \cup \{c\}$. (The distances are symmetric.) The points in X have to be supplied by vehicles from the depot. Each vehicle has a capacity of supplying at most q points. There are sufficiently many vehicles to supply all points. Also given is a constant K.

Goal: Determine whether there exist a set of routes, each starting and ending at the depot c, of total length at most K that supplies all points in X.

Hint: Use the fact that the decision version of the Traveling Salesman Problem *is NP-complete.*

Answer 3b. *More proofs can be correct than the one below.*

Proof. We first argue that the Vehicle Routing Problem is in *NP*.

We have to show that every yes-instance of the problem admits a certificate that can be verified in polynomial time. A certificate for a yes-instance is a set of routes $R = \{r_1, \ldots, r_m\}$, where each route $r_i \in R$ is represented by a sequence of points in $X \cup \{c\}$. We need to check whether

(a) each route $r_i \in R$ starts and ends in the depot c,

(b) each route $r_i \in R$ consists of at most q points in X,

(c) every point in X is supplied by at least one route $r_i \in R$,

(d) the total length of all routes in R is at most K.

The time needed to check (a) is $O(1)$ per route; thus $O(m)$ in total. The time needed to check (b) is $O(q)$ per route; thus $O(qm)$ in total. The time needed to check (c) is $O(qm+|X|)$. The time needed to calculate the total length of all routes in R is at most $O(qm)$. We can assume without loss of generality that $m \leq |X|$ and thus the above verification can be done in polynomial time.

Next we show that the Traveling Salesman Problem is polynomial-time reducible to the Vehicle Routing Problem. Because the Traveling Salesman Problem is *NP*-complete, this establishes *NP*-completeness of the Vehicle Routing Problem.

Suppose we are given an instance (G, d, K) of the Traveling Salesman Problem with graph $G = (V, E)$, distance function $d : E \to \mathbb{Z}^+$ and parameter K. Fix an arbitrary vertex of G as the depot c and let $X = V \setminus \{c\}$. Identify the distance function d and the parameter K of the Vehicle Routing Problem with the distance function d and the parameter K of the Traveling Salesman Problem, respectively. Let there be one vehicle with capacity $q = n - 1$. This transformation can be done in polynomial time. Note that (G, d, K) is a yes-instance of the Traveling Salesman Problem if and only if the corresponding instance of the Vehicle Routing Problem is a yes-instance. This concludes the proof. □

4. Consider the Metric Steiner Tree Problem:

Metric Steiner Tree Problem:

Instance: An undirected complete graph $G = (V, E)$ with non-negative edge costs $(c_e)_{e \in E}$ satisfying the *triangle inequality*, i.e., for every $u, v, w \in V$, $c_{uw} \leq c_{uv} + c_{vw}$, and a set of terminal vertices $R \subseteq V$.

Goal: Compute a minimum cost tree of G that connects all terminals in R.

The vertices in $V \setminus R$ are called *Steiner* vertices. The Metric Steiner Tree problem thus asks for the computation of a minimum cost tree, also called *Steiner tree*, spanning all terminals in R and possibly some Steiner vertices; see Figure 2 for an example.

Develop a 2-approximation algorithm for this problem.

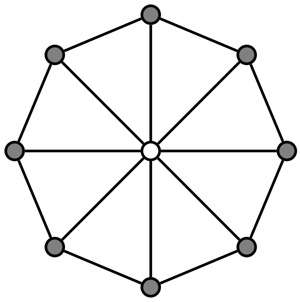

Figure 2: Example instance of the METRIC STEINER TREE PROBLEM in Question 4. There are 8 terminal vertices (indicated in gray) and one Steiner vertex. All edges to the Steiner vertex have cost 1 and all other edges have cost 2 (note that not all edges of the complete graph are shown). Observe that the cost of an optimal Steiner tree is 8, while the cost of a minimum spanning tree on the terminals is 14.

Hint: Show that a minimum spanning tree T on the terminal set R of G satisfies $\mathsf{OPT} \geq \frac{1}{2}c(T)$ and use this to derive an approximation algorithm.

Answer 4. *More proofs can be correct than the one below.*

Lemma 1. *Let T be a minimum spanning tree on the terminal set R of G. Then $\mathsf{OPT} \geq \frac{1}{2}c(T)$.*

Proof. Consider an optimal Steiner tree of cost OPT. By doubling the edges of this tree, we obtain a Eulerian graph of cost $2\mathsf{OPT}$ that connects all terminals in R and a (possibly empty) subset of Steiner vertices. Find a Eulerian tour C' in this graph, e.g., by traversing vertices in their depth-first search order. We obtain a Hamiltonian cycle C on R by traversing C' and short-cutting Steiner vertices and previously visited terminals. Because of the triangle inequality, this short-cutting will not increase the cost, and the cost of C is thus at most $c(C') = 2\mathsf{OPT}$. Delete an arbitrary edge of C to obtain a spanning tree on R of cost at most $2\mathsf{OPT}$. The cost of a minimum spanning tree T on R is less than or equal to the cost of this spanning tree, which is at most $2\mathsf{OPT}$. □

Lemma 1 gives rise to the following approximation algorithm.

Algorithm 1: Approximation algorithm for METRIC STEINER TREE.

Input: Complete graph $G = (V, E)$ with non-negative edge costs $(c_e)_{e \in E}$ satisfying the triangle inequality and a set of terminal vertices $R \subseteq V$.

Output: Steiner tree T on R.

1 Compute a minimum spanning tree T on terminal set R.
2 Output T.

Theorem 4. *Algorithm 1 is a 2-approximation algorithm for* METRIC STEINER TREE.

Proof. Certainly, the algorithm has polynomial running time and outputs a feasible solution. The approximation ratio of 2 follows directly from Lemma 1. □
