Exam Mastermath / LNMB MSc course on Discrete Optimization

January 8, 2018, 10:00 – 13:00

- Use of calculators, mobile phones, and other electronic devices not allowed.
- The exam consists of seven problems. You have approximately 25 minutes per problem.
- Please start a **new page for every problem.**
- Each question is worth 10 points. The total number of points is 70. 39 Points to pass.
- Relevant problem definitions appear at the end of the exam.

Problem 1 (Spanning Trees) Let $G = (V, E)$ be a graph and $c : E \to \mathbb{Z}_{\geq 0}$ a non-negative cost function on the edges. Design a polynomial time algorithm that computes a spanning tree T of G that minimizes the maximum weight of any edge in T. Prove the correctness of your algorithm.

Problem 2 (Matroids) Let $M = (E, \mathcal{I})$ be an independence system. That is, $\emptyset \in \mathcal{I}$ and for any $J \in \mathcal{I}$ and any $I \subseteq J$, also $I \in \mathcal{I}$. Show that the following two conditions are equivalent:

1. For any $U \subseteq E$, every basis of U has the same cardinality.
2. For every $I, J \in \mathcal{I}$ with $|I| < |J|$ there exists an element $x \in J \setminus I$ such that $I \cup \{x\} \in \mathcal{I}$.

Problem 3 (Matchings) Given an undirected connected graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges, an *edge cover* is a subset $C \subseteq E$ of the edges of the graph such that each node $v \in V$ is incident with at least one edge $e \in C$ (i.e., a set of edges that "cover" all the nodes of the graph). Denote by $\alpha(G)$ the size of a *minimum* cardinality edge cover of G, and by $\mu(G)$ the size of a *maximum* cardinality matching of G. Show that $\mu(G) + \alpha(G) = n$.
(Hint: From any maximum cardinality matching M, construct an edge cover to show $\mu(G) + \alpha(G) \leq n$. From any minimum cardinality edge cover C, construct a matching to show $\mu(G) + \alpha(G) \geq n$.)

Problem 4 (Minimum Cost Flows) Let $G = (V, E)$ be a directed graph with edge capacities $w : E \to \mathbb{Z}_{\geq 0}$ and edge costs $c : E \to \mathbb{Z}_{\geq 0}$ and balances $b : V \to \mathbb{Z}$. Prove that the following statements are equivalent for all feasible flows f:

1. The flow f is the unique minimum cost flow.
2. For every directed cycle C in the residual graph G_f, we have $c(C) > 0$.

Problem 5 (Hardness of Approximation) Given an undirected, connected graph $G = (V, E)$ with $|V| \geq 2$, the min-max degree spanning tree problem is to find a spanning tree T of the graph such that the maximal degree of the nodes in T is minimized. In other words, find a spanning tree $T = (V, E_T)$, $E_T \subseteq E$, that minimizes $\max_{v \in V} d_T(v)$, where $d_T(v)$ is the degree of node v in T. For convenience, let us call this optimization problem MDST. Assuming **P**$\neq$**NP**, show that there cannot be an α-approximation algorithm for the MDST problem with $\alpha < \frac{3}{2}$.

(**Hint:** Consider the problem to decide if an MDST exists with objective value $k = 2$.)

Problem 6 (Approximation Algorithms) Given is a graph $G = (V, E)$ consider the problem to find a subset of nodes $C \subseteq V$ that maximises the size of the cut induced by C, $|\delta(C)|$. This problem is known as the maximum cut problem. Design a 2-approximation algorithm for this problem. That is, your algorithm needs to compute, in polynomial time, a set C^* with $|\delta(C^*)| \geq \frac{1}{2} \max_{C \subseteq V} |\delta(C)|$. Prove that your algorithm is indeed a 2-approximation.

(**Hint:** One possibility is to first consider a randomized algorithm.)

Problem 7 (True / False Questions) Which of the following is true or false, assuming **P**$\neq$**NP**. Please explain your answer briefly, but precisely. That is, give a short proof, or a counterexample.

(a) There is a polynomial time reduction from SATISFIABILITY to any problem in **NP**.

(b) If there is a strongly polynomial time algorithm to solve the PARTITION problem, then there is a polynomial time algorithm to solve SATISFIABILITY.

(c) There is a polynomial time reduction from MATCHING to VERTEX COVER.

(d) All problems in **NP** can be reduced to each other.

Collection of Problems

MAXIMUM FLOW Given is a directed graph $G = (V, E)$ with edge capacities $w : E \to \mathbb{Z}_{\geq 0}$, and two designated nodes $s, t \in V$, the source and the target. The problem asks to compute a feasible (s, t)-flow with maximum value. The decision version asks if a flow with value $\geq k$ exists for given k. There exist polynomial time algorithms for MAXIMUM FLOW.

MINIMUM COST FLOW Given is a directed graph $G = (V, E)$ with edge capacities $w : E \to \mathbb{Z}_{\geq 0}$, edge costs $c : E \to \mathbb{Z}_{\geq 0}$ and node balances $b : V \to \mathbb{Z}$. The problem is to find a feasible flow that minimizes total costs. The decision version asks if a flow with cost $\leq k$ exists for given k. There exist polynomial time algorithms for MINIMUM COST FLOW.

HAMILTONIAN PATH (CYCLE) Given an undirected graph $G = (V, E)$, does there exist a simple path (cycle) that visits each of the vertices exactly once? Both problems are strongly **NP**-complete.

MATCHING Given an undirected graph $G = (V, E)$, a *matching* $M \subseteq E$ is a set of non-incident edges. The *maximum matching* problem is to find a matching M of G with maximum cardinality $|M|$. The decision problem asks if, for a given k, a matching of size $\geq k$ exists in G. Edmonds' blossom shrinking algorithm solves the maximum matching problem in polynomial time.

PARTITION Given are n integral, non-negative numbers $a_1, \dots, a_n$ with $\sum_{j=1}^{n} a_j = 2B$. The decision problem is to decide if there is a subset $W \subseteq \{1, \dots, n\}$ such that $\sum_{j \in W} a_j = \sum_{j \notin W} a_j$. This decision problem is **NP**-complete but (as a special case of the KNAPSACK problem) has a pseudo-polynomial time algorithm.

SATISFIABILITY Given n Boolean variables $x_1, \dots, x_n$, and a formula F that consists of the conjunction of m clauses C_i, $F = \bigwedge_{i=1}^{m} C_i$. Each clause consists of the disjunction of some of the variables x_j (or their negation $\bar{x}_j$), for example $C_5 = (x_1 \vee x_4 \vee \bar{x}_7)$. The decision problem is: Does there exist a truth assignment $x \in \{\text{false}, \text{true}\}^n$ such that $F =$true? This decision problem is strongly **NP**-complete.

VERTEX COVER Given is an undirected graph $G = (V, E)$. A *vertex cover* is a subset C of the nodes of V such that for any edge $e = \{u, v\} \in E$, at least one of the nodes u or v is in C. The decision problem asks if a vertex cover C exists with $|C| \leq k$. This decision problem is known to be strongly **NP**-complete.

MAXIMUM CUT Given is an undirected graph $G = (V, E)$. The question is to find a subset $C \subseteq V$ of the nodes of G that maximizes the number of edges in the cut induced by C, that is, a cut that maximizes $|\delta(C)|$, where $\delta(C) := \{\{u, v\} \in E \mid u \in C, v \notin C\}$. The decision problem is to decide, for given k, if $C \subseteq V$ exists with $|\delta(C)| \geq k$.