Exam Mastermath / LNMB MSc course on Discrete Optimization

January 7, 2019, 09:30 – 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 $w : E \to \mathbb{Z}_{\geq 0}$ a non-negative weight 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) Recall that a matroid is a family $\mathcal{I} \subseteq 2^E$ of subsets of some finite set E so that $\emptyset \in \mathcal{I}$, subsets of any $I \in \mathcal{I}$ also belong to $\mathcal{I}$, and for any two $I, J \in \mathcal{I}$ with $|I| < |J|$, there exists $e \in J$ such that $I \cup \{e\} \in I$. Prove that the following set system is a matroid. Given an undirected graph $G = (V, E)$, let

$$\mathcal{I} = \{S \subseteq V \mid \text{ there exists a matching } M \text{ of } G \text{ that covers } S\}.$$

Here, matching $M \subseteq E$ covers a set of nodes S if for all $v \in S$ there exists $e \in M$ so that $v \in e$. (Hint: Consider alternating paths in the symmetric difference of two matchings.)

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 maximum cardinality matching M, construct an edge cover to show $\mu(G) + \alpha(G) \leq n$. From 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:

(a) The flow f is the unique minimum cost flow.

(b) For every directed cycle C in the residual graph G_f, we have $c(C) > 0$.

Problem 4 (NP-hardness) The EXACT SPANNING TREE problem, denoted EST, is the following decision problem. Given is an undirected graph $G = (V, E)$, edge weights $w : E \to \mathbb{N}$, and a number $k \in \mathbb{N}$. Is there a spanning tree T of G such that $w(T) := \sum_{e \in T} w(e) = k$? Prove that EST is **NP**-complete. (Hint: To show **NP**-hardness, you can use a reduction from SUBSET SUM).

Problem 5 (Hardness of Approximation) Consider the BIN PACKING problem as defined in the problem collection. Show that the BIN PACKING problem cannot have an α-approximation algorithm when $\alpha < 3/2$.

Problem 6 (Designing 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 claims 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) Consider a directed network $G = (V, E)$ with $s, t \in V$ and nonnegative, integer edge capacities $w : E \to \mathbb{N}$ so that the maximum (s, t)-flow f has value > 0. **Claim:** When $w(e) \neq w(e')$ for any two $e, e' \in E$, there exists a unique minimum (s, t)-cut in G.

(b) Consider the KNAPSACK problem. and assume $P = \max_{i=1,\dots,n} p_i$ is the maximal profit value. **Claim:** If for some fixed ℓ there is an $O(n^\ell P)$-time algorithm to solve the KNAPSACK problem, then there is a polynomial time algorithm to solve SATISFIABILITY.

(c) Let an undirected graph $G = (V, E)$ be given, and $w : E \to \mathbb{N}$ be non-negative, integer edge weights. **Claim:** For any two $s, t \in V$, there exists a minimum spanning tree T that contains all the edges of a shortest (s, t)-path.

(d) A perfect matching M in an undirected graph $G = (V, E)$ is a subset of non-incident edges with $M = |V|/2$. Consider the decision problem PM that asks if a given graph G does have a perfect matching. **Claim:** There exists a polynomial time reduction from PM to the SATISFIABILITY problem.

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.

Knapsack Given is a knapsack of weight capacity $W \in \mathbb{N}$, and n items with integral weights w_i and integral profits p_i, all nonnegative. The decision problem is: Given an integer k, does there exist a subset of the items that fits into the knapsack and has value at least k? In other words, does there exists a set $S \subseteq \{1, \ldots, n\}$ with $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i \geq k$? This problem is **NP**-complete.

Partition Given are n integral, non-negative numbers $a_1, \ldots, a_n$ with $\sum_{j=1}^{n} a_j = 2B$. The decision problem is to decide if there is a subset $W \subseteq \{1, \ldots, n\}$ such that $\sum_{j \in W} a_j = \sum_{j \notin W} a_j$. This problem is **NP**-complete.

Subset Sum Given are n integral, non-negative numbers $a_1, \ldots, a_n$, and $k \in \mathbb{N}$. The decision problem is to decide if there is a subset $W \subseteq \{1, \ldots, n\}$ such that $\sum_{j \in W} a_j = k$. This problem is **NP**-complete.

Satisfiability Given n Boolean variables $x_1, \ldots, x_n$, 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 $x \in \{\text{false,true}\}^n$ such that F =true? This 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 problem is (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 $|\delta(C)| \geq k$, which is (strongly) **NP**-complete.

Bin Packing Given is a set $N = \{1, ..., n\}$ of items with sizes $s_i \in (0, 1]$, $s_i \in \mathbb{Q}$ for every $i \in N$. The goal is to pack all items into unit-size bins such that the number of used bins is minimized. Formally: find a partition of all items into k subsets $N_1, \ldots N_k$, such that $\sum_{i \in N_\ell} s_i \leq 1$ for all $\ell = 1, \ldots, k$, with k as small as possible. The decision problem asks if all items can be packed into at most k bins, for a given $k \in \mathbb{N}$.