Exam Mastermath / LNMB MSc course on Discrete Optimization

January 11, 2020, 14:00 – 17:30

- Open book exam: Allowed materials are lecture slides, lecture notes, exercise solutions.
- The exam consists of seven problems. You have a bit less than 1/2 h per problem.
- Please start a **new page for every problem**.
- Each question is worth 10 points. The total number of points is 70. 35 Points to pass.
- Relevant problem definitions, e.g. for **NP**-hardness, appear at the end of the exam.

Important practical rules for online exam

- The exam **must be hand-written**. Using a tablet computer with a pen for hand-writing is also allowed. If you write by hand and on paper, please make sure to convert your scans or pictures into **one single pdf-file** before uploading to the ELO platform. (Example apps are: adobe scan, CamScanner, Genius Scan, Tiny Scanner, ...)
- You must write the following **declaration of integrity** on top of the first page of your solutions: **"This exam was be solely undertaken by myself, without any assistance from others, and without use of sources other than those allowed." followed by your name, date, and your signature**.
- Please write your **name and student ID** on each single page (top right).
- We reserve $\frac{1}{2}$**h extra time** for any technical work such as taking pictures, file conversion, and upload. That means you can upload you exam until 17:30, w/o being marked late. Anything uploaded **later than 17:30, cannot be accepted** (unless in special circumstances such as, e.g., extra-time).
- There is a **Zoom meeting** open during the whole exam. In case of questions, you may use that. This is the link to the Zoom meeting (Meeting ID: 871 0923 3234, Passcode: 158050). With a random sample of students, I am conducting an ID & paper check using the same Zoom meeting. This will be between 17:00 and 17:30. Those students will be informed by email.

Exam Questions

Problem 1 (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 2 (Matroids) Given is an undirected graph $G = (V, E)$. We say that a subset of edges $C \subseteq E$ *covers* a subset of vertices $W \subseteq V$ if for all $w \in W$, there is at least one $e \in C$ with $w \in e$. Which of the following independence systems is a Matroid? Prove, or give a counterexample.

(a)
$$\{W \subseteq V \mid \forall\, v, w \in W : \{v, w\} \notin E\}$$

(b)
$$\{W \subseteq V \mid \exists \text{ matching } M \text{ that covers } W\}$$

Problem 3 (Spanning Trees and Shortest Paths) Let $G = (V, E)$ be an undirected graph and $w : E \to \mathbb{R}_{>0}$ a strictly positive weight function on the edges of G. Consider an arbitrary vertex $s \in V$, and for each $v \in V$, let $P(s, v)$ be the edge set of *some* shortest path from s to v. Define $P := \bigcup_{v \in V} P(s, v)$. Show that, if T is *any* minimum spanning tree for G, we must have $P \cap T \neq \emptyset$. Show that $P \cap T = \emptyset$ *is* possible, if $w(e) = 0$ is allowed, too.

Problem 4 (Matchings) Consider an undirected graph $G = (V, E)$ with $|V| = n$ vertices and $|E| = m$ edges. Recall that a subset of edges $C \subseteq E$ *covers* a subset of vertices $W \subseteq V$ if for all $w \in W$, there exists at least one $e \in C$ with $w \in e$.

(a) Let $C \subseteq E$ be any *minimum cardinality* set of edges that covers all vertices V of G. Also, let M be a matching of *maximum cardinality* for G. Prove that $|M| + |C| = n$. [**Hint:** Show "$\leq$" and "$\geq$" separate from each other.]

(b) Recall that a "perfect" matching $M \subseteq E$ is a matching that covers all vertices V of G. Prove the following: If G is a tree, then G has at most *one* perfect matching. [**Hint:** A proof by contradiction works.]

Problem 5 (Designing Approximation Algorithms) Consider the following packing optimization problem, called Pack. Given n integer nonnegative numbers $x_1, \ldots, x_n$, and a threshold $k \in \mathbb{Z}_{\geq 0}$, find a subset $W \subseteq \{1, \ldots, n\}$ with $x(W) := \sum_{i \in W} x_i \leq k$, maximizing $x(W)$. This problem is **NP**-hard by a reduction from Partition.

(a) Give a (polynomial time) 2-approximation algorithm for Pack. Prove the performance guarantee.

(b) Give an algorithm to compute the optimal solution value for Pack, with pseudo-polynomial computation time.

Problem 6 (Hardness of Approximation) Given an undirected, connected graph $G = (V, E)$ with $|V| \geq 2$, the "lean spanning tree" (LST) problem is to find a spanning tree T of G that minimizes the maximal degree of the nodes in T. To be precise, we want a spanning tree $T = (V, E_T)$, $E_T \subseteq E$, minimizing $\max_{v \in V} d_T(v)$, where $d_T(v)$ is the degree of node v in T. Assuming **P**$\neq$**NP**, show that there cannot be an α-approximation algorithm for the LST problem with $\alpha < \frac{3}{2}$. [**Hint:** What is an LST with objective value $k = 2$?]

Problem 7 (True / False Questions) Which of the following claims are true or false? Explain your answers briefly, but precisely. That is, give a **short** proof or a counterexample.

(a) Consider the class **NP**, which is the class of decision problems that can be solved by a nondeterministic polynomial time algorithm. **Claim**: If there is a polynomial time algorithm to solve just one problem in **NP**, then there is a polynomial time algorithm to solve any problem in **NP**.

(b) Consider an undirected graph $G = (V, E)$ with $|V| = n$. For a given integer vector $d \in \mathbb{N}^n$, a d-matching M is a subset of edges M of $E(G)$ such that in (V, M), the degree of vertices equals exactly $d_1, \ldots, d_n$ (for some permutation of the vertices). Consider the decision problem d-M that asks if a given graph G does have a d-matching. **Claim**: There exists a polynomial time reduction from d-M to the VERTEX COVER problem.

(c) Consider the PARTITION problem. **Claim**: As the PARTITION problem has a pseudo polynomial time algorithm, but the SATISFIABILITY problem is strongly NP-hard, there cannot be a polynomial time reduction from SATISFIABILITY to PARTITION.

(d) Consider the MAXIMUM CUT optimization problem, which asks to compute a subset C of the nodes of an undirected graph $G = (V, E)$ maximizing the number of edges $|\delta(C)|$ of the cut $\delta(C)$. **Claim**: If there is an FPTAS (fully polynomial time approximation scheme) for the MAXIMUM CUT optimization problem, then there is a polynomial time algorithm to solve SATISFIABILITY.

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 $f : E \to \mathbb{R}_{\geq 0}$ with maximum value. The decision version asks if a flow f with value at least 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 $f : E \to \mathbb{R}_{\geq 0}$ that minimizes total costs. The decision version asks if a flow f with cost at most k exists for given k. There exist polynomial time algorithms for MINIMUM COST FLOW.

HAMILTONIAN PATH / CYCLE Given an undirected (or directed) graph $G = (V, E)$, does there exist a simple (directed) path / cycle that visits each of the vertices exactly once? All four 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 $\sum_{j \in W} a_j = \sum_{j \notin W} a_j$=B. 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 $|\delta(C)| \geq k$. This problem is strongly **NP**-complete.