Exam Mastermath / LNMB MSc course on Discrete Optimization

January 10, 2022, 13:30 – 16:30

- Allowed materials are 4 pages of "cheat sheet" (written on both sides).
- 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 is 70. You need 35 points to pass.
- Relevant problem definitions, e.g. for **NP**-hardness, appear at the end of the exam.

Exam Questions

Problem 1 (Spanning Trees) Let $G = (V, E)$ be an undirected, connected graph, and let $w : E \to \mathbb{R}_{\geq 0}$ be a non-negative weight function on the edges of G. Give a polynomial time algorithm that computes a spanning tree T of G that minimizes the maximum weight of any edge in T, that is, it should compute a spanning tree T^* to minimize the function

$$f(T) := \max\{w(e) \mid e \in T\}$$

among all spanning trees T of G. Prove the correctness of your algorithm, and briefly argue about its computation time. [**Hint**: A greedy algorithms works.]

Problem 2 (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 2?]

Problem 3 (Matchings & Matroids) Consider an undirected graph $G = (V, E)$. A subset of edges $M \subseteq E$ *covers* a subset of vertices $W \subseteq V$ if for all $w \in W$, there is at least one edge $e \in M$, so that $w \in e$. Also recall that a *perfect* matching $M \subseteq E$ is a matching of G that covers all vertices V of G.

(a) Graph $G = (V, E)$ is *k-regular* if each node $v \in V$ has degree $d(v) = k$. Show that if $k \geq 1$, any k-regular, and bipartite graph $G = (A \cup B, E)$ has a perfect matching. [**Hint:** You may use Hall's theorem, that says that a bipartite graph $G = (A \cup B, E)$ has a matching that covers A, if and only if $|X| \leq |N(X)|$ for all $X \subseteq A$.]

(b) Decide if the following is a matroid or not, by giving a proof or a counterexample.

$$\{W \subseteq V \mid \text{There exists a matching } M \text{ that covers } W\}$$

[**Hint:** Verify if the augmentation property is fulfilled.]

Problem 4 (Minimum Cost Flows) Consider a minimum cost flow problem on a directed network $G = (V, E)$ with edge costs $c : E \to \mathbb{Z}_{\geq 0}$ and edge capacities $w : E \to \mathbb{Z}_{\geq 0}$. Let $f^* : E \to \mathbb{R}_{\geq 0}$ be a minimum cost flow, and π be a set of corresponding node labels such that the reduced cost optimality condition is fulfilled for f^*. That is, for $e = (u, v)$, $c^\pi(e) = c(e) - \pi(u) + \pi(v) \geq 0$ for all edges e in the residual graph, $e \in G(f^*)$.
Let $G^o(f^*)$ be the subgraph consisting only of those edges of $G(f^*)$ with zero reduced cost, that is, $c^\pi(e) = 0$.

Show that the following statements are equivalent.

(a) Flow f^* is not the unique minimum cost flow.

(b) The graph $G^o(f^*)$ has a directed cycle.

Problem 5 (Approximation Algorithms) Recall the SET COVER problem: We are given a ground set $E = \{1, \dots, m\}$, a collection of subsets $S_j \subseteq E$, $j = 1, \dots, n$, such that $E = \bigcup_{j=1}^{n} S_j$. Subset S_j has a cost $c_j \geq 0$. The goal is to find a minimum cost cover for E. In other words, find a collection of subsets S_j, $j \in W \subseteq \{1, \dots, n\}$, so that $E = \bigcup_{j \in W} S_j$, with minimal total cost $\sum_{j \in W} c_j$.
Letting parameter $a_{ij} = 1$ if item i is contained in subset S_j (and 0 otherwise), recall the following integer linear programming formulation for SET COVER, where variables $x_j \in \{0, 1\}$ denote selecting or not selecting subset S_j.

$$\begin{aligned} \min & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \geq 1 & \forall\, i = 1, \dots, m \\ & x_j \in \{0, 1\} & \forall\, j = 1, \dots, n \end{aligned}$$

The corresponding linear programming relaxation (LP) is obtained by relaxing the $\{0, 1\}$ constraints:

$$\begin{aligned} \min & \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \geq 1 & \forall\, i = 1, \dots, m \\ & x_j \geq 0 & \forall\, j = 1, \dots, n \end{aligned}$$

Let f_e be the number of times that an element e appears in the subsets S_j, so $f_e := |\{S_j \mid e \in S_j\}|$. Also, let $f := \max\{f_e \mid e \in E\}$. Assume we we have algorithm LP-SOLVER that can compute an optimal solution x^{LP} of linear programming relaxation (LP), in polynomial time.

Give an LP-rounding algorithm A that is an f-approximation algorithm for the SET COVER problem[1]. Prove that your algorithm is indeed an f-approximation algorithm.

Problem 6 (Algorithm Design) Give a dynamic programming algorithm to solve the following variant of the PARTITION problem: Given $\{a_1, \dots, a_n\}$ with integer $a_i \geq 0$ for $i = 1, \dots, n$, find a subset $S \subseteq \{1, \dots, n\}$ that minimizes $|\sum_{i \in S} a_i - \sum_{i \notin S} a_i|$. It suffices if you compute the optimal value (not S itself.) Also give a pseudo-polynomial upper bound on the computation time of your algorithm.

[1] Your algorithm **must** use the optimal LP solution x^{LP} and should "round" this fractional solution x^{LP} to a feasible solution $x^A \in \{0, 1\}^n$ of the SET COVER problem.

Problem 7 (True / False Questions) Which of the following claims are true, and which are false? Explain your answers briefly, but precisely. That is, give a **short** proof or a counterexample. For all questions, 1-3 sentences are enough.

(a) 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.

(b) 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 for any problem in **NP** there is a polynomial time algorithm that solves it.

(c) Consider the MINIMUM (s,t)-CUT problem, where we are given an undirected graph $G = (V, E)$, nodes $s, t \in V$, and an integer number k. It asks if there exists a subset C of the nodes with $s \in C$ but $t \notin C$, such that the cut $\delta(C)$ induced by C has at most k edges, $|\delta(C)| \le k$. **Claim:** There exists a polynomial time reduction from MINIMUM CUT to MAXIMUM CUT.

(d) 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.

(e) PRIMES is the decision problem "is n a prime?" **Claim:** PRIMES is in co-**NP**.

Collection of Decision Problems

MAXIMUM FLOW Given is a directed graph $G = (V, E)$, with integer edge capacities $w : E \to \mathbb{Z}_{\ge 0}$, and two designated nodes $s, t \in V$, and an integer number k. The problem asks if there exists a feasible (s,t)-flow $f : E \to \mathbb{R}_{\ge 0}$ with value $\text{val}(f) \ge k$. There exist polynomial time algorithms for MAXIMUM FLOW.

MINIMUM COST FLOW Given is a directed graph $G = (V, E)$, with integer edge capacities $w : E \to \mathbb{Z}_{\ge 0}$, integer edge costs $c : E \to \mathbb{Z}_{\ge 0}$, node balances $b : V \to \mathbb{Z}$, and an integer number k. The problem is to decide if a feasible flow $f : E \to \mathbb{R}_{\ge 0}$ exists such that its total cost fulfills $\sum_{e \in E} f(e)c(e) \le k$. There exist polynomial time algorithms for MINIMUM COST FLOW.

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

HAMILTONIAN PATH / CYCLE Given is an undirected (or directed) graph $G = (V, E)$. The problem is to decide if there exist a simple (directed) path / cycle that visits each of the vertices exactly once. All four problems are **NP**-complete.

KNAPSACK Given is a knapsack of weight capacity $W \in \mathbb{N}$, and n items with integer weights w_i and integer profits p_i, all nonnegative, and an integer number k. The decision problem asks if there exists $S \subseteq \{1, \ldots, n\}$ with $\sum_{i \in S} w_i \le W$ and $\sum_{i \in S} p_i \ge k$? 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.

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 problem asks if there exists a truth assignment $x \in \{\text{false,true}\}^n$ such that F = true? This problem is **NP**-complete.

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

MAXIMUM CUT Given is an undirected graph $G = (V, E)$, and an integer number k. The question is to decide if a subset $C \subseteq V$ of the nodes of G exists, such that the cut $\delta(C)$ induced by C, has at least k edges, $|\delta(C)| \geq k$. This problem is **NP**-complete.