Combinatorial Optimization 1 Winter term 2019/20

1. The SUBSET SUM problem is given as follows

   INPUT: $n \in \mathbb{N}$, n natural numbers $a_1, \ldots, a_n$, number $B \in \mathbb{N}$.
   QUESTION: Does there exist a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = B$ holds?

   The partition problem PARTITION is given as follows

   INPUT: $n \in \mathbb{N}$, n natural numbers $a_1, \ldots, a_n$.
   QUESTION: Does there exist a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$ holds?

   (a) Prove that SUBSET SUM is NP-complete. You can assume for your proof that it is known that PARTITION is NP-complete.

   (b) Prove that PARTITION is NP-complete. You can assume for your proof that SUBSET SUM is NP-complete.

2. Prove that the following problem is NP-hard:

   INPUT: $n \in \mathbb{N}$, $n \times n$ matrix $D = (d_{ij})$.
   QUESTION: A TSP tour with length at most $(1+\epsilon)$ times the length of a shortest TSP tour. (ϵ is a fixed constant > 0.)

   You can assume that it is known that the Hamiltonian cycle problem (HC) is NP-complete.

3. Prove that the maximum clique problem (CLIQUE) is NP-complete.

   INPUT: An undirected graph $G = (V, E)$, a number $k \in \mathbb{N}$.
   QUESTION: Does there exist a clique (complete subgraph of G) with cardinality (=number of vertices) $\geq k$?

   You can assume that it is known that (VERTEX COVER) is NP-complete:

   INPUT: An undirected graph $G = (V, E)$ and a number $k \in \mathbb{N}$.
   QUESTION: Does G contain a vertex cover $C \subseteq V$ with $\leq k$ vertices? (A subset V' of V is called vertex cover of G if every edge in E is incident with at least one vertex in V'.)

4. Prove that the maximum cut problem (MAXIMUM CUT) is NP-hard.

   INPUT: An undirected graph $G = (V, E)$ along with nonnegative edge weights w_{ij}.
   QUESTION: A cut $(X, \bar{X})$ in G ($X \cup \bar{X} = V$, $X \cap \bar{X} \neq \emptyset$) with maximum capacity $\sum\limits_{(i,j) \in E; i \in X, j \in \bar{X}} w_{ij}$.

   (Hint: Use the PARTITION in your reduction.)

5. (a) Use Kruskal's algorithm to determine a minimum spanning tree (MST) for the graph displayed in Figure 1.

   (b) As in (a), but now use Prim's algorithm. Start from vertex r.

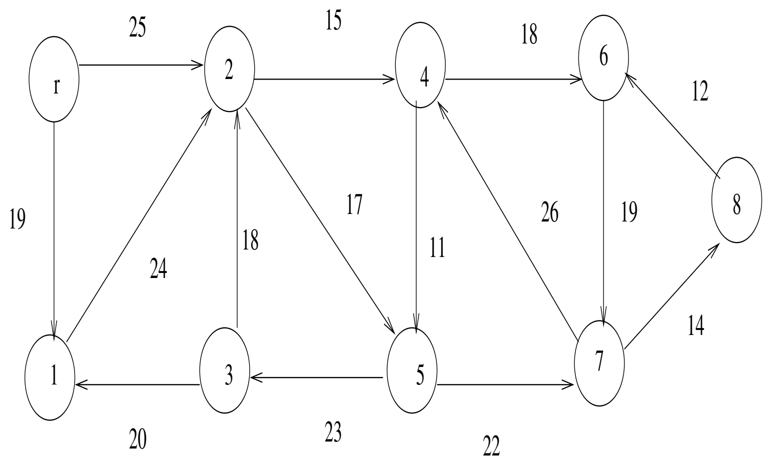

Figure 1

6. We are given an undirected graph $G = (V, E)$ with edge weights w_e, $e \in E$. Are the following statements true or false? (Provide a short proof for true ones and a counterexample for false ones.)

   (a) If $T^* = (V, E_{T^*})$ is a spanning tree which fulfills $\sum_{e \in E_{T^*}} w_e \geq \sum_{e \in E_T} w_e$ for all spanning trees $T = (V, E_T)$, it is implied that $\max_{e \in E_{T^*}} w_e \geq \max_{e \in E_T} w_e$ for all spanning trees $T = (V, E_T)$.

   (b) If $T^* = (V, E_{T^*})$ is a spanning tree with $\max_{(i,j) \in E_{T^*}} w_{ij} \leq \max_{(i,j) \in E_T} w_{ij}$ for all spanning trees $T = (V, E_T)$, it is implied that $\sum_{(i,j) \in E_{T^*}} w_{ij} \leq \sum_{(i,j) \in E_T} w_{ij}$ holds for all spanning trees $T = (V, E_T)$.

   (c) If T is an MST with respect to the weights w_e, T is also an MST with respect to the weights $w'_e = w_e + k$ with $k \in \mathbb{R}$.

   (d) If T is an MST with respect to the weights w_e, T is also an MST with respect to the weights $w'_e = k w_e$ with $k \in \mathbb{R}$.

   (e) Let $E = \{e_1, \ldots, e_m\}$ be a numbering of the edges such that $w_{e_1} \leq w_{e_2} \leq \ldots \leq w_{e_m}$ and set $w'_{e_i} = i^2$, $i = 1, \ldots, m$. If $T^* = (V, E_{T^*})$ is a spanning tree for which $\max_{e \in E_{T^*}} w_e \leq \max_{e \in E_T} w_e$ holds for all spanning trees $T = (V, E_T)$, then it also holds that $\max_{e \in E_{T^*}} w'_e \leq \max_{e \in E_T} w'_e$ for all spanning trees $T = (V, E_T)$.

   (f) For each minimum spanning tree problem P in a connected graph with arbitrary real edge weights w_e there exists an equivalent minimum spanning tree problem $\widetilde{P}$ on the same graph but with nonnegative edges weights $\widetilde{w}_e$. (P and $\widetilde{P}$ are said to be equivalent if they have the same set of optimal solutions.)

7. We are given an undirected graph $G = (V, E)$ with edge weights $w_e \in \mathbb{R}$ for all $e \in E$. Provide an algorithm which determines a spanning forest $F = (V, E_F)$ (spans all vertices; each component is a tree) with minimum weight $w(F) = \sum_{e \in E_F} w_e$.

8. We are given an undirected graph $G = (V, E)$ with edge weights $w_e \in \mathbb{R}$ for all $e \in E$. Provide an algorithm which determines a spanning connected subgraph $H = (V, E_H)$ of G with minimum weight $w(H) = \sum_{e \in E_H} w_e$.

9. Consider the minimum spanning tree problem with the *lexicographic objective function* which can be described as follows:

   We are given an undirected graph $G = (V, E)$ with edge weights $w_e \in \mathbb{R}$ for all $e \in E$. Let $m = |E|$ and let $T = (V, E_T)$ be a spanning tree. The lexicographic weight $w(T)$ of T is defined to be the m

dimensional vector which results by sorting the edge weights of the edges in E_T in non-decreasing order. (For example for a tree T with edge weights 3, 4, 4 and 7 we have $w(T) = (3, 4, 4, 7)$.)

We want to find a spanning tree T with minimum lexicographic weight. (The order used here is the classical lexicographic order for vectors.)

Provide an algorithm that finds such a tree (along with a correctness argument). Try to aim for a fast algorithm.

10. Let $G = (V, E)$ be an undirected graph with edge weights w_e, $e \in E$. For each $e \in E$ we define the set $M_e := \{f : f \in E, w_f \leq w_e\}$. Let $e_0 \in E$ be an edge which fulfills the following property:

$$e_0 \in \operatorname{argmin}\{w_e : e \in E, M_e \text{ contains the edge set of a spanning tree in } G\}\,.$$

Prove that w_{e_0} is the optimal objective function value of the minimum spanning tree problem with bottleneck tree weight $w(T) = \max_{e \in E_T} w_e$.

11. **The cut optimality condition.** Let $G = (V, E)$ be an undirected, connected graph with edge weights w_e, $e \in E$ and let $T = (V, E_T)$ be a spanning tree in G. Let $e \in E_T$ be a tree edge. If e is deleted from T, T decomposes into 2 components; denote their vertex sets by V_e and $V \setminus V_e$. Let D_e be the cut $\delta(V_e) = \{e = u, v \in E : u \in V_e, v \in V \setminus V_e\}$.

Prove the following statement: T ist an MST in G iff for each edge $e \in E_T$ we have that $w_e \leq w_p$ holds for all edges $p \in D_e$.

12. **The path optimality conditions.** We use the same notation as in problem 11.

Let $e = u, v \in E \setminus E_T$ be an edge not in the tree T. We denote by P_e the unique path in T that joins u and v. Prove the following statement: T is an MST in G iff for each edge $e \in E \setminus E_T$ we have that $w_e \geq w_p$ holds for all edges $p \in P_e$.

13. Devise a criterion with the help of which one can decide whether a given instance of the minimum spanning tree problem has more than one optimal solution (MST). This criterion needs to work also for the case when not all edge weights are pairwise disjoint.

14. Analyse the time complexity of the algorithms for checking the minimality of a spanning tree by means of (a) the optimality criterion from Problem 12 and (b) the optimality criterion from Problem 11?

Is one of the two criteria outperforming the other in all cases? If so, which one and why? If no, what influences which of the two criteria is better suitable?

15. Consider the following algorithm for determining a spanning tree in a given undirected connected graph G: Sort the edges with respect to non-increasing weight and process the edges in this order. An edge is rejected (deleted from G) if it is not a bridge (the deletion of a bridge creates two components). Otherwise the edge is left in G. Continue with this procedure until all edges have been processed.

Does this algorithm deliver a MST as resulting graph? In the affirmative case provide a proof. Otherwise provide a counterexample.

16. Proof that every spanning tree which minimizes the sum weight $\sum_{e \in E_T} w_e$ also minimizes the bottleneck weight $\max_{e \in E_T} w_e$.

17. Do the optimality criteria from Problems 11 and 12 remain valid for the minimum spanning tree problem with bottleneck weight $w(T) = \max_{e \in E_T} w_e$?

Provide a counterexample or a proof respectively for each of the two criteria.

18. Let $G = (V, E)$ be a connected undirected graph where the edge set is partitioned into red and blue edges.

    (a) Provide an algorithm that finds a spanning tree with a maximum number of red edges.

    (b) Let T be a spanning tree with k red edges and let T' be a spanning tree with k' ($k' > k$) red edges. Prove that for each k'', $k < k'' < k'$, there exists a spanning tree with k'' red edges.

19. Let $T = (V, E_T)$ be an undirected graph and $1 \in V$ be a special vertex. T is called a *1-tree* if E_T contains exactly one cycle C and if the vertex 1 is contained in C and has degree $d(1) = 2$. Provide a greedy algorithm for the following problem:

    Given an undirected graph $G = (V, E)$ with edge weights w_e, find a spanning 1-tree $T = (V, E_T)$, $E_T \subseteq E$, with minimal weight $w(T) = \sum_{e \in E_T} w(e)$.

20. Let $T^* = (V, E_{T^*})$ be an MST. A spanning tree $\widetilde{T} = (V, E_{\widetilde{T}})$ is called *second-best* spanning tree if no other spanning tree with the exception of T^* has smaller weight. Devise an algorithm for finding a second-best spanning tree (try to aim for a fast algorithm).

    (Additional task: Generalize your approach to finding a k-best spanning tree for $k \in \mathbb{N}$.)

21. For a finite set V of points in the Euclidean plane, the Voronoi diagram consists of the regions

    $$P_v := \{\, x \in \mathbb{R}^2 : ||x - v||_2 = \min_{w \in V} ||x - w||_2 \,\}$$

    for $v \in V$. The Delauney triangulation of V is the graph G_D with vertex set v and edge set $\{\{v, w\} \subseteq V, v \neq w, |P_v \cap P_w| > 1\}$.

    Let G be the complete graph with vertex set V. The weight w_e of edge $e = \{v, w\}$ is given by $||v - w||_2$.

    Prove that every minimum spanning tree of G is a subgraph of the Delauney triangulation G_D.

22. The club TOP SECRET has n members who all live in different cities. A secret message needs to be distributed to all members by e-mail. The used e-mail program allows only to send the message from one member to one other member (not to several members at the same time). Of course a member can call the program more than once. For internal reasons, not every pair of members can communicate directly. We are given a list L of all (unordered) pairs $\{i, j\}$ of members where i and j can communicate directly with each other (it is assumed that if i can contact j, j can contact i, too). For each such pair $\{i, j\}$ in L we are given the probability p_{ij} of the event that an e-mail sent between i and j ends up in the wrong hands (due to some network attack).

    The goal is to determine a communication plan which minimizes the overall probability that the message gets into the wrong hands at some point during the communication process. (We assume that the events that a pair communication gets attacked are independent from each other).

    (a) Model this problem as combinatorial optimization problem (ground set, objective function) and describe an efficient algorithm for its solution.

    (b) How does this problem considered here relate to the problems we have dealt with in the lecture?

23. Extra problem (Task: Attack it without using literature): Either prove that the following problem is NP-hard or provide a polynomial time algorithm for its solution:

    We are given an undirected graph $G = (V, E)$ with edge weights $w_e \in \mathbb{R}$ for $e \in E$ and a set $L \subset V$. The goal is to find a spanning tree with minimum weight among all trees for which all leaves are contained in the set L (there can be additional leaves in the tree not contained in L).