Exam Combinatorial Optimization, 22 October 2020 (version 1)
Time: 18.45-21.30 (2 hours, 45 min.)

- It is not allowed to use any books, notes, or calculator. Just pen and paper.
- You may also obtain points for partial answers.
- Do explain each computation that you make.
- This exam has 7 questions, one on each page. The total number of points is 40

Question 1 (8 points total) The table shows an instance of the weighted Set Cover problem. A value '1' means that element e_i is included in set S_j. For example, $S_1 = \{1,2\}$. The second row gives the weights of the sets.

	S_1	S_2	S_3	S_4	S_5	S_6
	100	50	50	30	100	100
e_1	1	0	0	0	1	0
e_2	1	0	0	0	0	1
e_3	0	1	1	0	1	0
e_4	0	1	1	0	0	1
e_5	0	1	0	1	1	0
e_6	0	0	1	1	0	1

(a) (2pt.) Give an optimal solution and its value for this set cover instance.

(b) (2pt.) Give the ILP-formulation for <u>**this**</u> set cover instance.

(c) (2pt.) Give the the dual of the LP-relaxation of the formulation given in (b). Use a variable y_i for each element e_i.

(d) (2pt.) Apply the primal dual algorithm to this instance. Increase the dual values step by step in the order $y_1, y_2, \ldots$. Give the final y-values and the sets chosen by the algorithm.

Question 2 (6 points total)

$w_1 = 4, \quad C_1 = x_1 \vee x_2$
$w_2 = 4, \quad C_2 = x_2 \vee \bar{x}_3$
$w_3 = 8, \quad C_3 = x_3 \vee \bar{x}_4 \vee x_5$
$w_4 = 16, \quad C_4 = x_3 \vee \bar{x}_5 \vee x_6 \vee \bar{x}_7$
$w_5 = 16, \quad C_5 = \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_7$

(a) (1pt.) Shown is an instance of the weighted maximum satisfiability problem. What is the expected total weight of satisfied clauses when you assign each variable independently to true with probability 0.5? Show your computation.

(b) (1pt.) Assume that you are given an instance in which each clause has at least one positive x-variable. Argue that an optimal solution is easily found. (For example, for the given instance, only C_5 has no positive variables.)

(c) (2pt.) Argue that the following algorithm is a 0.5-approximation for the weighted maximum satisfiability problem,.

```
Consider the following two solutions and take the best:
(1) Set all variables to True.
(2) Set all variables to False.
```

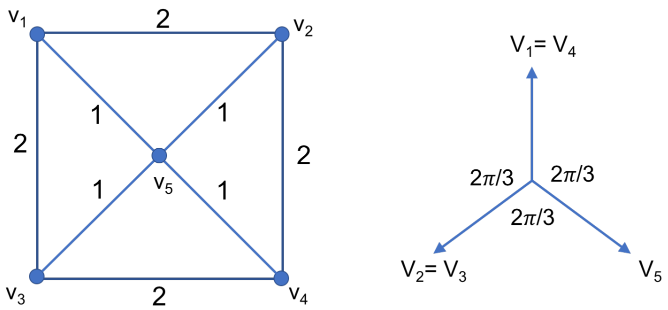
(d) (2pt.) It is known that the algorithm of question (a) is a $3/4$-approximation when all clauses have at least two variables. Give an example with 2 clauses and 2 variables in each clause which shows that the approximation ratio of the algorithm of (c) is not better 0.5.

Question 3 (6 points total)

Shown below are the Quadratic Program and the Vector Program relaxation for the maximum weighted cut problem, where an instance is an edge-weighted graph $G = (V,E)$ on n vertices.

$$\text{(QP)} \quad \max \quad \frac{1}{2} \sum_{(i,j)\in E} (1 - y_i y_j) w_{ij}$$
$$s.t. \quad y_i \in \{-1, 1\} \qquad i = 1, \ldots, n.$$

$$\text{(VP)} \quad \max \quad \frac{1}{2} \sum_{(i,j)\in E} (1 - v_i \cdot v_j) w_{ij}$$
$$s.t. \quad v_i \cdot v_i = 1, \; v_i \in \mathbb{R}^n \qquad i = 1, \ldots, n.$$



(a) (1pt.) What is the value of the maximum weighted cut in the graph above (left)? Explain your answer.

(b) (2pt.) Shown is a solution to the VP for this instance (right). What is the VP-value of this solution?

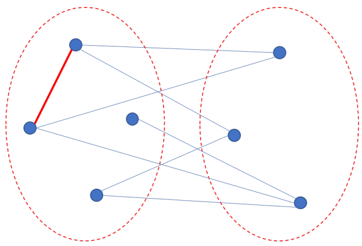
(c) (1pt.) Argue that this is not an optimal solution to the VP.

(d) (2pt.) Remember that the max-cut algorithm of Goemans and Williamson (of chapter 6) was defined as follows: Solve VP and apply randomized rounding. What is the expected weight of the cut when you apply this algorithm? Show your computation.

Question 4 (6 points total) Consider the following clustering problem. Here, we want to partition the vertices of a graph into k sets (clusters) so that the total number of edges within a cluster is minimized.

k-CLUSTERING:

Instance:	Graph $G = (V, E)$
Solution:	A partition of the vertices into k sets (clusters): $S_1, S_2, \ldots, S_k \subseteq V$
Cost:	The total number of edges that have both endpoints in the same cluster.
Goal:	Minimize cost.



k=2 and the cost is 1.

(a) (2pt.) Show, by a reduction from the maximum cut problem, that Clustering is NP-hard for $k = 2$.

(b) (2pt.) Now show by a reduction from the 3-coloring problem (vertex coloring) that Clustering is NP-hard for $k = 3$.

(c) (2pt.) Now show by a reduction from the 3-coloring problem that for $k = 3$ there is no α-approximation algorithm for k-Clustering for any $\alpha \geq 1$.

Question 5 (3 points total) Consider the k-center problem. Remember, an instance is given by a set of points in a metric space and a number k. For a given set of points in a metric space, let OPT_k be the optimal value for the k-center problem when k centers can be chosen and let OPT_{k+1} be the optimal value when $k+1$ centers can be chosen.

(a) (2pt.) Give (draw) an example (a set of points and distances) for which $\text{OPT}_1 > 3\text{OPT}_2$.

(b) (1pt.) Argue that for any k and $\alpha > 1$ there exists an instance for which $\text{OPT}_k/\text{OPT}_{k+1} > \alpha$.

Question 6 (4 points total) Below is the ILP for the Uncapacitated Facility Location (UFL) problem.

$$
\begin{aligned}
\text{(ILP)}\quad \min\quad & Z = \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\
s.t.\quad & \sum_{i \in F} x_{ij} = 1 & \text{for all } j \in D, \\
& x_{ij} \leqslant y_i & \text{for all } i \in F, j \in D, \\
& x_{ij} \in \{0,1\} & \text{for all } i \in F, j \in D, \\
& y_i \in \{0,1\} & \text{for all } i \in F.
\end{aligned}
$$

Here, F is the set of facilities and D is the set of clients,

We make the following change to the UFL: *In stead of connecting all clients we only need to connect* 3 *clients and we need to find those 3 for which the total cost (connection plus opening) is minimized.*

(a) (3pt.) Give an ILP for this problem based on the ILP above.

(b) (1pt.) Argue that an optimal solution can be found in polynomial time.

Question 7 (7 points total)

For this question, we first recap the theory of the Prize-collecting Steiner Tree problem (as written in the lecture notes).

PRIZE-COLLECTING STEINER TREE:

Instance:	$G=(V,E)$ and a cost c_e for every edge $e \in E$ and a penalty π_i for every vertex $i \in V$. Also given is a root $r \in V$.
Solution:	Tree T containing r. Let $V(T)$ be the vertices in T
Value:	$\sum_{e \in T} c_e + \sum_{i \in V-V(T)} \pi_i$.
Goal:	Find a solution of minimum cost.

In words, the goal is to find a tree containing r that minimizes the cost of all edges in the tree plus the penalty cost for the vertices that are not connected by T. An ILP for the problem is as follows.

$$\begin{aligned}
\text{(ILP)} \quad \min \quad & Z = \sum_{e \in E} c_e x_e + \sum_{i \in V} \pi_i (1 - y_i) & \\
s.t. \quad & \sum_{e \in \delta(S)} x_e \geqslant y_i, & \text{for all } i, S \text{ with } i \in S \subseteq V - r, \\
& y_r = 1, & \\
& x_e \in \{0,1\}, & \text{for all } e \in E, \\
& y_i \in \{0,1\}, & \text{for all } i \in V.
\end{aligned}$$

Algorithm : $(0 < \alpha < 1)$
Step 1: Solve the LP-relaxation of (ILP) $\rightarrow \; x^*, y^*, Z^*_{LP}$.
Step 2: Let $U = \{i \mid y^*_i \geqslant \alpha\}$. Construct a Steiner tree T on U.

Lemma 1 *The connection cost for the Steiner tree T is*

$$\sum_{e \in T} c_e \leqslant \frac{2}{\alpha} \sum_{e \in E} c_e x^*_e.$$

Lemma 2 *The total penalty cost is*

$$\sum_{i \in V-V(T)} \pi_i \leqslant \frac{1}{1-\alpha} \sum_{i \in V} (1 - y^*_i) \pi_i.$$

Theorem 1 *The algorithm above with $\alpha = 2/3$ is a 3-approximation algorithm for the Prize-collecting Steiner Tree problem.*

Proof: The sum of connection and penalty cost is at most

$$\frac{2}{\alpha}\sum_{e\in E} c_e x_e^* + \frac{1}{1-\alpha}\sum_{i\in V}(1-y_i^*)\pi_i \leqslant \max\left\{\frac{2}{\alpha}, \frac{1}{1-\alpha}\right\} Z_{LP}^*.$$

The value $\max\left\{\frac{2}{\alpha}, \frac{1}{1-\alpha}\right\}$ is 3 for $\alpha = 2/3$. Hence, the total cost of the solution of the algorithm is at most

$$3Z_{LP}^* \leqslant 3\mathrm{OPT}.$$

(a) (2pt.) If we would apply this algorithm with $\alpha = 1/2$, then what approximation ratio do you get? Show your computation. You may refer to the theory above.

(b) (1pt.) In the *unrooted* version of the problem, no root r is given. That means, any tree is allowed. Show that if you have a β-approximation algorithm for the rooted version (for example with $\beta = 3$), then you can easily get a β-approximation for the unrooted version.

Now, we make the following adjustment to the Prize-collecting Steiner Tree problem:
In stead of one root r we are given two roots, r_1 and r_2. The problem is to find two trees T_1, T_2 with $r_1 \in T_1$ and $r_2 \in T_2$. The goal is to minimizes the cost of all edges in the two trees plus the penalty cost for the vertices that are not in any tree.

(c) (2pt.) For some given instance, let $\mathrm{OPT}(r_1, r_2)$ be the optimal value for the Prize-collecting Steiner Tree problem with roots r_1 and r_2 and let $\mathrm{OPT}(r_1)$ be the optimal value for the standard Prize-collecting Steiner Tree problem with one root r_1. Argue that $\mathrm{OPT}(r_1, r_2) \leq \mathrm{OPT}(r_1)$.

(c) (2pt.) Write down an ILP for this problem based on the ILP above. Use variables x_e^1 and x_e^2 and y_i^1 and y_i^2.