Resit Combinatorial Optimization, 9 December 2020

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)

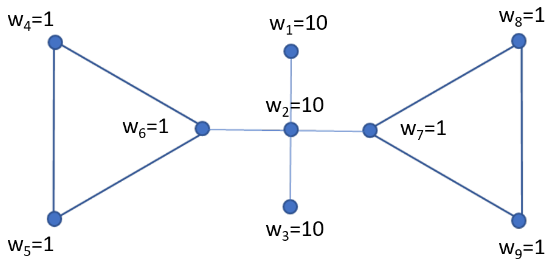
The graph shows an instance of the *weighted* vertex cover problem. The weights are $w_1 = w_2 = w_3 = 10$ and all other weights are 1.

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

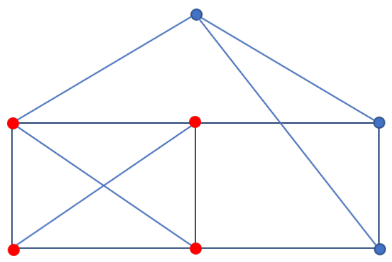
(b) (2pt.) Write down the Integer Linear Program (ILP) for this weighted vertex cover instance.

(c) (2pt.) For this instance, give a solution to the LP-relaxation which has a value strictly smaller than the optimal value (given in (a)).

(d) (2pt.) Suppose you apply the LP-rounding algorithm to the LP-solution that you gave in part (c). Then, what is the vertex cover returned by the algorithm and what is its value?



Question 2 (5 points total) A *clique* in a graph $G = (V,E)$ is a set $S \subseteq V$ such that for all $i, j \in S$, edge $(i, j) \in E$. That means, every pair of vertices in S is connected by an edge. In the maximum clique problem we need to find a clique of maximum size. For example, the graph shown has a clique of size 4.



(a) (2pt.) Consider the following greedy algorithm:
`Greedy: Select an arbitrary vertex and keep adding vertices to S as long as S is a clique.`
Show by an example that the approximation ratio of this algorithm can be arbitrarily bad. For example, give an instance (make a picture) for which the optimal value is 4 times the value of a solution of the algorithm and argue that your example can be generalized to any (integer) ratio $\alpha \geq 1$.

Now assume that the graph has a special structure: the vertices can be partitioned into 2 sets A and B such that both A and B are cliques. For example, in the figure, the red vertices form a clique but also the blue vertices form a clique.

(b) (1pt.) Show that given the partition A, B, a 0.5-approximate solution is easy to find.

(c) (2pt) Show that even when the partition A, B is not given (but it does exist), still a 0.5-approximate solution is easy to find.

Question 3 (4 points total)

First a recap of some theory:
For each clause C_j let P_j be the indices of the variables x_i that occur positively in the clause, and let N_j be the indices of the variables x_i that are negated in the clause. The following mixed ILP is an exact formulation of the Max Sat problem.

$$\begin{array}{lll} \text{(ILP) max} & Z = \sum_{j=1}^{m} w_j z_j & \\ s.t. & \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geqslant z_j & \text{for all } j = 1 \ldots m, \\ & y_i \in \{0,1\} & \text{for all } i = 1 \ldots n, \\ & z_j \in \{0,1\} & \text{for all } j = 1 \ldots m. \end{array}$$

For the relaxation, replace the last two constraints by $0 \leq y_i \leq 1$ and $0 \leq z_j \leq 1$.

Now we change the problem a bit. Assume that each clause C_j has an *even* number of literals l_j. Moreover, a clause C_j is said to be satisfied if at least $l_j/2$ of its literals are true (in stead of just 1). Again, the objective is to maximize the number of satisfied clauses.

(a) (2pt.) Give the ILP for this problem, based on the ILP above.

(b) (2pt.) Describe, for such instances in general, an LP-solution y, z for which the LP-value is at least the optimal value: $Z_{LP} \geq \text{OPT}$.

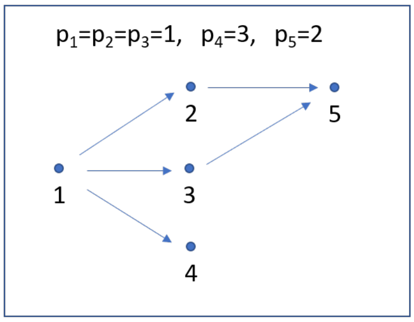
Question 4 (4 points)

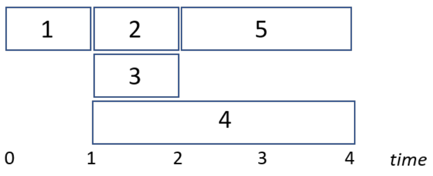
In the precedence constrained scheduling problem we are given a set of n jobs where each job j has a length p_j and also given is an acyclic directed graph $G = (V,A)$, describing the job precedence relations. If arc $(j,k) \in A$ then job j must be completed before job k can start. Preemption (splitting of jobs) is not allowed. Further, we are given a deadline D by which all jobs need to be completed. We need to find a feasible schedule such that all jobs complete before time D. The value of the solution is the number of machines used. An illustrating example is given below. If, for this example, we are given a deadline $D = 4$, then the minimum number of machines we need is 3. If $D = 5$, then 2 machines are enough.

PRECEDENCE CONSTRAINED SCHEDULING:

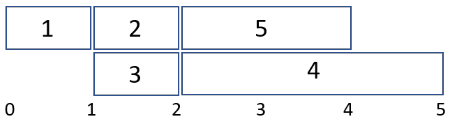
Instance:	n jobs with length p_j for $j = 1,\dots,n$, a directed graph $G = (V,A)$ with $\lvert V\rvert = n$, and a number D.
Solution:	A schedule on identical machines satisfying the precedence constraints and of length at most D.
Value:	The number of machines in the schedule.
Goal:	Minimize value.

Example

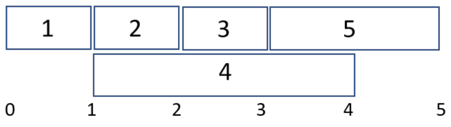




A schedule of length 4 on **3** machines.



A schedule of length 5 on **2** machines.



A schedule of length 5 on **2** machines.

It is known that deciding if there exists a schedule with at most 2 machines is an NP-complete problem. (That means, assuming P$\neq$ NP, there is no polynomial time algorithm that will always correctly answers the question "Is there a solution of value at most 2?" correctly.)

Show that there is no α-approximation algorithm for this scheduling problem for $\alpha < 3/2$, assuming that P$\neq$ NP.

Question 5 (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.

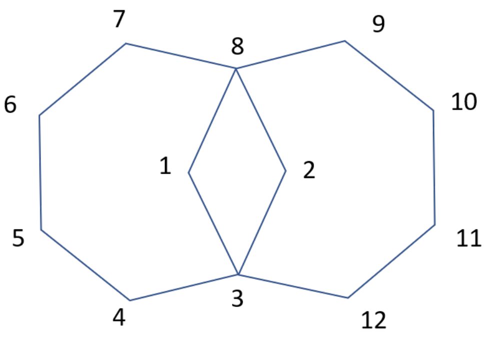
$$\begin{aligned}
\text{(QP)}\quad \max \quad & \tfrac{1}{2} \sum_{(i,j)\in E} (1 - y_i y_j) w_{ij} & \\
s.t. \quad & y_i \in \{-1, 1\} & i = 1, \ldots, n.
\end{aligned}$$

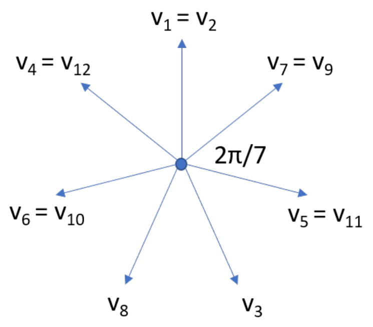
$$\begin{aligned}
\text{(VP)}\quad \max \quad & \tfrac{1}{2} \sum_{(i,j)\in E} (1 - v_i \cdot v_j) w_{ij} & \\
s.t. \quad & v_i \cdot v_i = 1,\ v_i \in \mathbb{R}^n & i = 1, \ldots, n.
\end{aligned}$$

(a) (2pt.) Give the Quadratic Program for the maximum cut problem for the graph shown below (left). (That means, do not give the general formulation but write the formulation for this instance explicitly.)

(b) (2pt.) For the graph below, give a subset S of the vertices such that the number of edges with exactly one endpoint in S is maximum. (In other words, give a maximum cut.) Argue why it is a maximum cut.

(c) (2pt.) Remember that the the randomized rounding algorithm first computes an optimal solution to the VP and then rounds that VP-solution by taking a hyperplane uniformly at random. Assume that, in stead of taking an optimal VP-solution, we apply this algorithm with the VP-solution shown (on the right). Then what is the expected number of edges in the cut? (Show your computation.)





Question 6 (5 points total) Consider a scheduling instance with $m = 3$ machines and 7 jobs with processing times $p_1 = p_2 = \cdots = p_6 = 1$, and $p_7 = 3$.

(a) (1pt.) Suppose that you apply the list scheduling algorithm to this instance where jobs are listed in the order $1, 2, \ldots, 7$. What is the length ($C_{\max}$) of the schedule returned by the algorithm?

(b) (1pt.) Suppose that you apply the LPT-algorithm to this instance. Then what is the length of the schedule returned?

(c) (3pt.) Show that for any number of machines m, there is an instance such that the approximation ratio of the list scheduling algorithm is **at least** $2 - 1/m$.

Question 7 (8 points total) Below is the ILP for the Uncapacitated Facility Location (UFL) problem.

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

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

We make the following change to the UFL (for questions (a) and (b) only):

- There are 2 types of facilities, F_1 and F_2, and each client must be connected to one facility of each type. ($F = F_1 \cup F_2$ and $F_1 \cap F_2 = \emptyset$.)

(a) (2pt.) Give an ILP for this problem based on the ILP above.

(b) (2pt.) For the standard UFL as given by the ILP above, a 4-approximation algorithm is known. Argue that a 4-approximation algorithm can be obtained for this problem as well.

We go back to the standard problem (UFL) and now make the following change:

- There is one more layer to the problem: In addition to clients and facilities there also is a set W of warehouses. The cost for opening a warehouse $k \in W$ costs w_k. We must choose which of the warehouses to open. Each open facility must be connected to an open warehouse. The cost for connecting facility i with warehouse k is b_{ik}.

Hence, all clients must be connected to one open facility and all open facilities must be connected to an open warehouse.

(c) (2pt.) For the standard UFL as given by the ILP above, a 4-approximation algorithm is known. Argue that if all opening costs for the warehouses are zero ($w_k = 0$) then a 4-approximation algorithm can be obtained for this problem as well. (Hint: explain how to reduce this problem to the standard problem.)

(d) (2pt.) Now assume again arbitrary opening costs w_k. Argue that if we restrict to instances with $|W| \leq 10$, then a 4-approximation algorithm can be obtained for this problem as well.

Appendix

The following problems can be solved in polynomial time:

- Linear programs and Vector programs.
- Finding a minimum cut in a graph.
- Finding the shortest path between two points in a graph.
- Finding a minimum spanning tree in a graph.
- Finding a maximum matching in a graph.

The following problems are NP-hard/NP-complete:

- Integer Linear programming
- Hamiltonian Cycle
- Hamiltonian Path
- Maximum cut
- Vertex coloring with 3 colors.
- Vertex Cover
- Set Cover