Exam Advanced Algorithms. 16-12-2014.

- It is not allowed to use any books, notes, calculator ... just pen and paper.
- Keep your answers short. Just give the main idea. Notation is not so important.
- There are many questions but most have a really short answer.
- The table shows the maximum number of points per sub question. Although all questions have the same weight, some are clearly easier than others.

1a	1b	1c	1d	2a	2b	2c	3a	3b	4a	4b	4c	4d	5a	5b	6a	6b	6c	Σ
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	18

In your answers, you may use the following facts.

- The Hamiltonian Cycle problem is NP-complete.
- In a complete graph with costs (weights) on the edges, a perfect matching of minimum cost can be found in polynomial time.
- Linear programs can be solved in polynomial time.
- A minimum spanning tree can be found in polynomial time.
- The shortest path between points s and t in a network (a graph with lengths on the edges) can be found in polynimial time.
- $\cos(2\pi/3) = -0.5$.

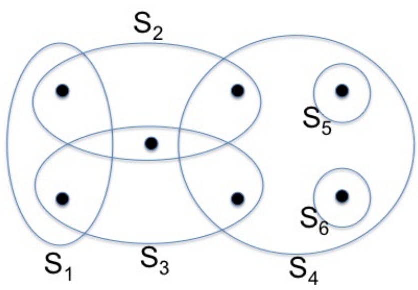
1. This question is about the unweighted set cover problem of chapter 1. The instance below has 7 elements and 6 sets.

(a) Give an optimal set cover for the instance below. (Which sets do you take?)

(b) Write down the ILP for set cover for the given example.

(c) For this example, give a solution to the LP-relaxation which has value strictly smaller than the value of the optimal set cover.

(d) In the given example, each element is in exactly two sets. It can therefore also be seen as a vertex cover problem. Draw the corresponding graph.



2. This question is about the *weighted* set cover problem. An instance is given by a set of elements (items) $E = \{e_1, \ldots, e_n\}$, subsets $S_1, \ldots, S_m \subseteq E$, and a weight w_j for all $j \in \{1, \ldots, m\}$.

(a) Give the LP dual for the weighted set cover problem. Denote the variables by y_i $(i = 1, \ldots, n)$.

Set cover algorithm:
Step 1: Solve the dual.
Step 2: For each j, take set S_j in the solution if the dual constraint for S_j is *tight* (meaning we have equality).

For questions (b) and (c) you may use the following notation. Denote the optimal dual solution by $y_1^*, \ldots, y_n^*$ and denote the optimal primal and optimal dual values by Z_{LP}^* and Z_D^*. Let I be the indices of the sets in the solution given by the algorithm: $I = \{j \mid S_j \text{ in solution}\}$.

(b) Argue that the algorithm above always returns a feasible set cover.

(c) Assume that each element appears in at most f sets, for some integer f. Argue that the algorithm is an f-approximation algorithm.

3.
TSP (SYMMETRIC):

Instance:	Complete graph with a cost c_{ij} for every pair of points i, j.
Solution:	A cycle that goes through each point exactly once.
Value:	The length (sum of the edge costs) of the cycle.
Goal:	Find a solution of minimum cost.

(a) Let $\alpha \geqslant 1$ be some constant. Prove that there is no polynomial time α-approximation algorithm for the non-metric TSP, unless P=NP.

Now, consider the metric-TSP, that means, $c_{ij} \leqslant c_{ik} + c_{kj}$ for all i, j, k.

(b) Describe a 3/2-approximation algorithm for metric TSP and prove the ratio $3/2$. (Just give the main idea/arguments.)

4. In the minimum s,t-cut problem we are given a graph $G=(V,E)$ and $s,t \in V$ and we need to find a smallest set of edges $W \subseteq E$ such that removing W separates s from t (that means s and t end up in different components). You probably know that the problem can be solved by a max flow algorithm. In this exercise we use a different approach to solve it.
Let $\mathscr{P}$ be the set all simple paths from s to t in the graph. (A path is simple if no vertex is visited more than once by the path.) For every edge $(u,v) \in E$ introduce a variable x_{uv}. The following ILP is an exact formulation of the minimum s,t-cut problem.

$$\begin{aligned}
\text{(ILP)}\min \quad & Z = \sum_{(u,v)\in E} x_{uv} \\
s.t. \quad & \sum_{(u,v)\in P} x_{uv} \geqslant 1 \qquad \text{for all } P \in \mathscr{P}. \\
& x_{uv} \in \{0,1\} \qquad \text{for all } (u,v) \in E.
\end{aligned}$$

(a) In the LP-relaxation, we take $x_{uv} \geqslant 0$ in stead of $x_{uv} \in \{0,1\}$. Explain why it is not immediately clear that the relaxation can be solved in polynomial time.

The LP-relaxation can be solved in polynomial time by using the ellipsoid method together with a separation oracle.

(b) Describe what a separation oracle does in general for an LP. Further, show that this LP-relaxation has a polynomial time separation oracle.

Let x^* be an optimal solution for the LP-relaxation and let Z^* be its value. Now, for every edge $(u,v) \in E$ define its *length* as x^*_{uv}. Using these lengths, let $L(v)$ be the distance from s to v in G, for every $v \in V$. In other words, $L(v)$ is the length of the shortest path from s to v in G. For any $\gamma \in [0,1[$, define the set of vertices $S_\gamma = \{v \in V \mid L(v) \leqslant \gamma\}$, that means, S_γ is the set of vertices at distance at most γ from s.

Min cut algorithm
Step 1: Solve the LP-relaxation $\rightarrow\ x^*, Z^*$.
Step 2: Take $\gamma \in [0,1[$ uniformly at random. $\rightarrow\ S_\gamma$.
Step 3: Return the set of edges that have exactly one endpoint in S_γ. Denote this set by W.

(c) Show that W has at most Z^* edges in expectation: $\mathbb{E}(|W|) \leqslant Z^*$.

(d) Explain how the algorithm can be derandomized.
(*This question can be answered even if you did not find an answer for (c).*)

One more page $\longrightarrow$

5.

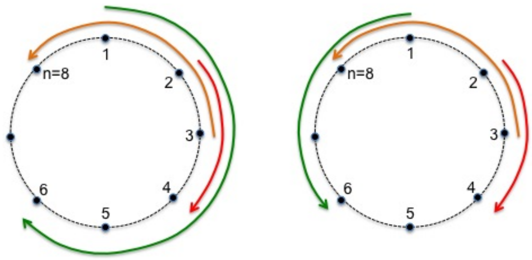


Figure 1: *(For question 5) The instance in this example is given by* $n = 8$, $C = \{c_1, c_2.c_3\}$ *with* $c_1 = (1,6)$, $c_2 = (2,4)$, *and* $c_3 = (3,8)$. *Left is a solution with value* 3. *Right is an optimal solution with value 2.*

The following problem arises in telecommunications networks, and is known as the SONET ring loading problem. The network consists of a cycle on n nodes, numbered 1 through n clockwise around the cycle. Some set C of calls is given; each call is a pair (i, j) originating at node i and destined to node j. The call can be routed either clockwise or counterclockwise around the ring. The load L_e on edge e of the cycle is the number of calls routed through edge e. The value of the solution is the maximum over all n loads: $\max_e L_e$. The objective is to route the calls so as to minimize the maximum load on the network.

(a) Give a (mixed) ILP for the SONET ring loading problem. Use the following notation: Take a binary decision variable x_i for each call $c_i \in C$, where $x_i = 1$ iff call c_i is routed clockwise. For each edge e, let C_e be the set of (indices of) the calls which route through e *if* routed clockwise. For example, $C_e = \{1,3\}$ for edge $e = (4,5)$ in the example above.

(b) Describe a 2-approximation algorithm and give a proof for this.

6.

(a) Give the vector program relaxation for the graph 3-coloring problem of Section 6.5.

(b) One can show that the optimal solution of the vector program has value at most -0.5 if the graph is 3-colorable. Give a 3-colorable graph with exactly 5 edges and sketch a solution to the vector program of value at most -0.5.

(c) Now suppose you apply one iteration of the algorithm of section 6.5 to your graph with your vector program solution given in (b). More precisely, take $t = 2$ hyperplanes at random. The two hyperplanes divide the solution space into 4 regions. (The probability that the two hyperplanes are the same is zero). Let each region correspond with a color. Hence, this gives a coloring of the vertices with (at most) 4 colors. Show that the probability that the coloring is feasible (i.e., no two adjacent vertices get the same color) is at least $4/9$.

– – – – – – – – – – – – – – –

Solutions

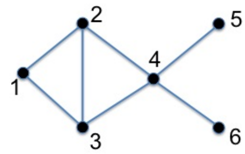
1a For example, $1,2,4$. (Other possibilities: $2,3,4$ and $1,3,4$). Optimal value is 3.

1b

$$
\begin{array}{lll}
\min & x_1+x_2+x_3+x_4+x_5+x_6 & \\
s.t. & x_1+x_2 \geqslant 1 & \\
 & x_2+x_3 \geqslant 1 & \\
 & x_1+x_3 \geqslant 1 & \\
 & x_2+x_4 \geqslant 1 & \\
 & x_3+x_4 \geqslant 1 & \\
 & x_4+x_5 \geqslant 1 & \\
 & x_4+x_6 \geqslant 1 & \\
 & x_i \in \{0,1\} & \text{i=1,2,}\ldots\text{, 6.}
\end{array}
$$

1c For example, $(x_1,x_2,x_3,x_4,x_5,x_6) = (\frac{1}{2},\frac{1}{2},\frac{1}{2},1,0,0)$ with value 2.5.

1d



2a

$$
\begin{array}{llll}
\text{(D)} & \max & Z = \sum_{i=1}^{n} y_i & \\
 & s.t. & \sum_{i:e_i \in S_j} y_i \leqslant w_j & \text{for all } j = 1,\ldots,m \\
 & & y_i \geqslant 0 & \text{for all } i = 1,\ldots,n.
\end{array}
$$

2b Assume that e_i is not covered. Then, none of the constraints j with $e_i \in S_j$ is tight. But then we can increase y_i^* by a small positive value and obtain a feasible solution with higher value. This contradicts that y^* is optimal.

2c The value of the solution found is

$$\sum_{j \in I} w_j = \sum_{j \in I} \sum_{i:e_i \in S_j} y_i^* \leqslant f \sum_{i=1}^{n} y_i^* = f Z_D^* \leqslant f Z_{LP}^* \leqslant f \text{OPT}.$$

The first *equality* above follows since only tight sets S_j were picked. The first *inequality* follows from the fact that each of the y_i^*'s appears at most f times in the summation. The second inequality follows from weak duality. (Also, the algorithm runs in polynomial time and in (b) we already showed that any solution is feasible.)

3a Follows by a reduction from the Hamiltonian Cycle problem. Assume we have an α-approximation algorithm ALG. Given an instance $G = (V,E)$ define an instance of TSP by taking

$$c_{ij} = 1 \text{ if } (i,j) \in E \text{ and } c_{ij} = M \text{ if } (i,j) \notin E,$$

where M is a large number. Let OPT and ALG denote the optimal value and algorithm's value for the TSP instance. Then, the following implications hold.

$$
\begin{array}{lllll}
\text{G has a Hamiltonian cycle} & \Rightarrow & OPT = n & \Rightarrow & ALG \leqslant \alpha n. \\
\text{G has no Hamiltonian cycle} & \Rightarrow & OPT \geqslant n-1+M & \Rightarrow & ALG \geqslant n-1+M.
\end{array}
$$

Choose M such that $\alpha n < n-1+M$. For example, $M = \alpha n$.

3b This is done by Christofides' algorithm: (1) Construct a minimum spanning tree T. (2) Find a mincost perfect matching M of the odd-degree vertices of T. (3) Find an Euler tour in the graph with edges $T \cup M$. (4) Shortcut the tour.
Claim 1: length of $T \leqslant \text{OPT}$: If we delete an edge from the optimal tour then we get a path connecting all vertices. Since this is also a tree, the minimum spanning tree has length at most OPT.
Claim 2: length of $M \leqslant \text{OPT}/2$: Let O be the odd degree vertices in T. Shortcut the optimal tour on O. This tour consists of exactly two perfect matchings on O. Hence, the length (cost) of M is at most $\text{OPT}/2$.
Claim 1+2 $\Rightarrow \text{ALG} \leqslant \text{OPT} + \text{OPT}/2$.

4a The number of constraints is not polynomially bounded. There may be exponentially many simple s,t paths.

4b Given en LP-solution x, a separation oracle either states (correctly) that x is feasible or it gives us a violated constraint. For the given LP-relaxation, a separation oracle should tell wether or not there is a simple s,t path P for which $\sum_{(u,v)\in P} x_{uv} < 1$. This can be done by computing the shortest path from s to t using x for the distances of the edges. If the shortest path has length at least 1 then the solution is feasible and otherwise the shortest path P will be a violated constraint.

4c Consider an edge (u,v) and assume $L(u) \leqslant L(v)$. Then,

$$\Pr(\text{ edge } (u,v) \text{ in cut }) = \Pr(L(u) \leqslant \gamma < L(v)) \ \leqslant L(v) - L(u) \leqslant x^*_{uv}.$$

The last inequality follows since $L(v)$ is at most the length of the path from s to v via u: $L(v) \leqslant L(u) + x^*_{uv}$. Hence,

$$\mathbb{E}[|W|] = \sum_{(u,v)\in E} \Pr(\text{ edge } (u,v) \text{ in cut }) \leqslant \sum_{(u,v)\in E} x^*_{uv} = Z^*.$$

Although not asked for, you also get points if you showed that W is indeed a feasible cut. Since $L(s) = 0$ and $L(t) \geq 1$ it follows from the definition of S_γ that $s \in S_\gamma$ and $t \notin S_\gamma$ for any $\gamma \in [0,1[$. So W is a feasible cut.

4d Since Z^* is the optimal value of the relaxation we have $Z^* \leqslant \text{OPT}$. With question **4c** this implies $\mathbb{E}[|W|] \leqslant \text{OPT}$. Since W is always a feasible cut we have $|W| \geq \text{OPT}$ for any choice of γ. Together with $\mathbb{E}[|W|] \leqslant \text{OPT}$ this implies that $W = \text{OPT}$ for any choice of γ. Therefore, the derandomized algorithm can fix any value of γ. For example, $\gamma = 0$.

N.B. *It is also fine if you answered here that the derandomized algorithm simply tries many different values of γ and then takes the best solution. But do note here that it is enough to try only the values $L(v)$ for all $v \in V$, which are at most n different values. It is not OK if you answered 'by the method of conditional expectations' without any further explanation.*

5a

$$\begin{array}{lll}
\min & Z & \\
s.t. & \sum_{i\in C_e} x_i + \sum_{i\notin C_e} (1-x_i) \leqslant Z & \text{for all edges } e \\
 & x_i \in \{0,1\} & \text{for all calls } c_i \\
 & Z \geqslant 0 \text{ (not really needed)} &
\end{array}$$

5b Algorithm:
(1) Solve the LP-relaxation in which $x_i \in \{0,1\}$ is replaced by $0 \leqslant x_i \leqslant 1$.
(2) Route c_i clockwise if $x_i^* \geqslant 1/2$ and route it counter clockwise otherwise.
For the proof it is convenient to define the value $y_i = 1$ if $x_i^* \geqslant 1/2$ and $y_i = 0$ other wise. Then, $y_i \leqslant 2x_i^*$ and $1 - y_i \leqslant 2(1 - x_i^*)$. The load on an edge e is

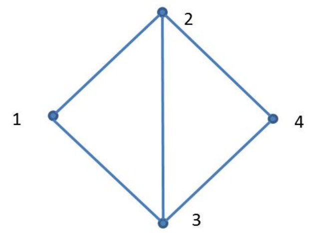
$$\sum_{i \in C_e} y_i + \sum_{i \notin C_e} (1 - y_i) \leqslant \sum_{i \in C_e} 2x_i^* + \sum_{i \notin C_e} 2(1 - x_i^*) = 2Z^* \leqslant 2\text{OPT}.$$

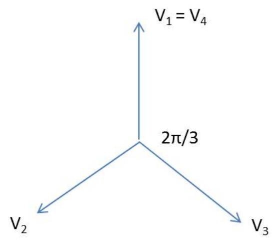
Other algorithms are possible. For example, always choosing the shortest of the two directions is also a 2-approximation.

6a For graph $G = (V,E)$ with $|V| = n$, the relaxation is

$$\begin{array}{lll} \min & \lambda & \\ s.t. & v_i \cdot v_j \leqslant \lambda & \text{for all } (i,j) \in E \\ & v_i \cdot v_i = 1 & \text{for all } i \in V \\ & v_i \in \mathbb{R}^n & \text{for all } i \in V. \end{array}$$

6b For example the graph left. (Actually, any graph with 5 edges is OK here.) The solution (right) has value -0.5. Another example (C_5) is given in the lecture notes.





6c For any edge (i,j) and <u>one</u> random hyperplane:

$$\Pr(v_i \text{ and } v_j \text{ are not separated }) \leqslant \frac{\pi/3}{\pi} = \tfrac{1}{3}.$$

Thus,
$$\begin{aligned} & \Pr(i \text{ and } j \text{ get the same color}) \\ = \; & \Pr(v_i \text{ and } v_j \text{ not separated be either hyperplane }) \leqslant \tfrac{1}{3}\tfrac{1}{3} = \tfrac{1}{9}. \end{aligned}$$

$$\Rightarrow \Pr(\text{endpoints of } \underline{\text{some}} \text{ edge get the same color }) \leqslant 5 \cdot \tfrac{1}{9} = \tfrac{5}{9}.$$

$$\Rightarrow \Pr(\text{coloring is feasible }) \geqslant 1 - \tfrac{5}{9} = \tfrac{4}{9}.$$