Second set of exam problems for the course 'Combinatorial Optimization' (DM867) spring 2022

Jørgen Bang-Jensen
Department of Mathematics and Computer Science
University of Southern Denmark

The problems are available from the course page as of April 4, 2022 The solutions must be uploaded via Itslearning no later than Monday May 2, 2022 at 9 a.m.

Note the following important points:

- There are 75 points to earn.
- You must work alone.
- It is important that you explain how you obtain your answers and argue why they are correct.
- If you are asked to describe an algorithm, then you must supply enough details so that a reader who does not already know the algorithm (but knows basic algorithms such a searching for a path from p to q, finding a maximum (s,t)-flow or a minimum (s,t)-cut, an alternating path with respect to a given matching in a bipartite graph, etc) can understand it (but you do should not give pseudo code).
- You should also give the complexity of the algorithm when relevant. In particular, when you are asked to describe a polynomial algorithm, you should argue why the algorithm you describe is in fact polynomial.
- If you are asked to show how to find a certain structure or number, then you must describe a polynomial algorithm for this.
- You may use all the algorithms described in the course as subroutines when this is appropriate. That is, if you are asked to do something where one of the algorithms from the course IS the algorithm (and not just one step in a larger one), then you must still explain the algorithm in detail.
- Note also that illustrating an algorithm means that one has to follow the steps of the algorithm meticulously (DK: slavisk).
- **Any exchange of results/ideas between different students before 9 a.m. May 2, 2022 will be considered as exam fraud.**

PROBLEM 1 (15 point)

Consider the instance $< K_6, w >$ of TSP shown in Figure 1.

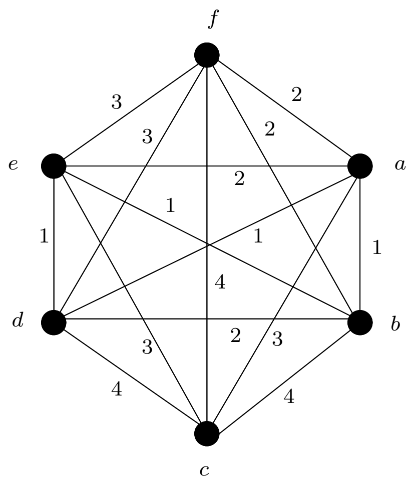

Figure 1: An instance of TSP. The weights are as follows: ab, ad, be, de all have weight 1, ae, af, bd, bf all have weight 2, ac, ce, df, ef all have weight 3 and bc, cd, cf have weight 4.

Question a:

Argue that the instance satisfies the triangle inequality.

Question b:

Argue that the tree $T = (V, E)$ with $V = \{a, b, c, d, e, f\}$ and $E = \{ab, ac, ad, af, de\}$ is a minimum spanning tree of the complete graph in Figure 1.

Question c:

Demonstrate the MST-based 2-approximation algorithm when you start from the minimum spanning tree T and give the resulting hamiltonian cycle and its weight.

Question d:

Demonstrate Christofides' algorithm when you start from the tree T and give the resulting hamiltonian cycle and its weight.

PROBLEM 2 (15 point)

This problem is about the Steiner tree problem.

Question a:

Give a short description of the 2-approximation algorithm for the Steiner tree problem that we have seen in the course.

Question b:

Illustrate the algorithm on the weighted graph in Figure 2 where the black vertices form the set of Steiner vertices (those that must be in the tree).

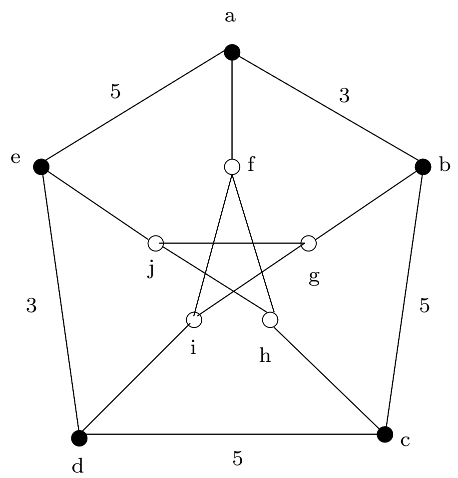

Figure 2: An instance of the Steiner tree problem. The black vertices are the Steiner vertices. All edges whose weight is not shown have weight 1.

Question c:

Is your solution unique?

Question d:

Is the solution optimal?

PROBLEM 3 (15 point)

This problem is about the 2-path problem in acyclic digraphs, that is, we are looking for a pair of vertex disjoint paths P_1, P_2 with prescribed end vertices s_1, t_1, respectively s_2, t_2.

Question a:

Explain in your own words (you do not need to prove correctness) how to reduce the 2-path problem for an acyclic digraph D to the problem of finding a directed path from a vertex s to a vertex t in another graph D'. You should give the running time of the algorithm that you obtain. Hint: see the proof of Theorem 9.2.14 in BJG.

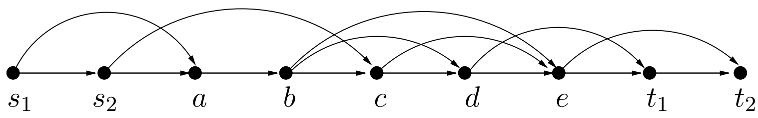

Figure 3: An acyclic digraph D

Question b:

Let D be the acyclic digraph in Figure 3. Which path in D' corresponds to the solution $P_1 = s_1 \to a \to b \to d \to t_1$ and $P_2 = s_2 \to c \to e \to t_2$?

Question c:

Which solution in D corresponds to the following path in D'?

$$\binom{s_1}{s_2} \to \binom{a}{s_2} \to \binom{a}{c} \to \binom{b}{c} \to \binom{e}{c} \to \binom{e}{d} \to \binom{e}{t_2} \to \binom{t_1}{t_2}$$

Question d:

A **feedback vertex set** of a digraph $D = (V, A)$ is a set of vertices $X \subseteq V$ such that the digraph $D' = D - X$ obtained by deleting all vertices in X is acyclic. So for acyclic digraphs the size of a minimum feedback vertex set is zero.

Prove that the 2-path problem can be solved in polynomial time on digraphs that have a feedback vertex set of size one. Hint: use that the k-path problem is polynomial for acyclic digraphs when k is a constant.

Question e:

Generalize your solution above to digraphs with a feedback vertex set of size at most 2.

PROBLEM 4 (10 point)

Let G be the weighted graph in Figure 4.

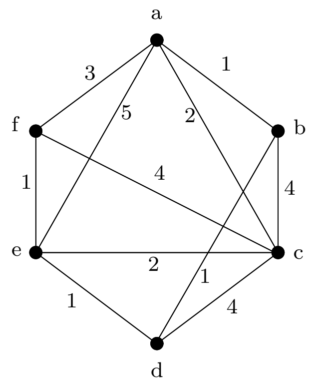

Figure 4: An instance of the Chinese postman problem. The numbers indicate the lengths of the edges.

Question a:

Let G, w be the instance of the Chinese postman problem shown in Figure 4. Show how to find a minimum length set of edges to double (use twice) in G so that one can make an Euler tour in the extended graph G'. Show G' and use that to describe a minimum length postman tour in G.

Question b:

Suppose now that the length of all the edges is increased by one, so, e.g. the length of ab is now 2. Will the same postman tour still be optimal?

PROBLEM 5 (20 point)

Let $B = (X_1, X_2, E)$ denote a bipartite graph with vertex set $X_1 \cup X_2$ and edges E, where each edge in E has one end in X_1 and the other in X_2. Define two subset systems $\mathcal{I}_i$, $i = 1, 2$ on E by letting $\mathcal{I}_i$ consist of those subsets E' of E where no two edges in E' share a vertex in X_i for $i = 1, 2$.

Question a:

Show that for each $i = 1, 2$ the set system $M_i = (E, \mathcal{I}_i)$ is a matroid whose rank function r_i satisfies that $r_i(E')$ is equal to the number of vertices in X_i which are incident with an edge from E'.

Question b:

Show how the problem of finding a maximum matching in a bipartite graph can be formulated as a matroid intersection problem.

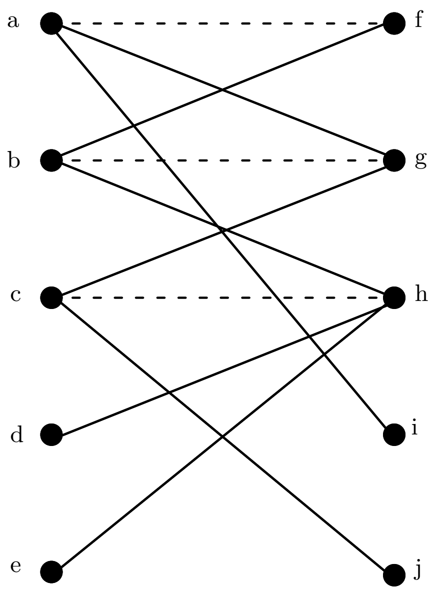

Figure 5: A bipartite graph with bipartition $X_1 = \{a, b, c, d, e\}$, $X_2 = \{f, g, h, i, j\}$. The three dotted edges indicate the current matching.

Question c :

Give a detailed explanation of the matroid intersection algorithm specialized to the problem above and indicate how the algorithm works by showing how the algorithm finds a maximum matching in the graph in Figure 5 when it is started from the matching shown. Does it matter whether we take a shortest (S_X, T_X)-path or an arbitrary (S_X, T_X)-path for this specialization of the algorithm?

Question d :

Give a proof of König's theorem (the size of a maximum matching in a bipartite graph equals the size of a minimum vertex cover) based on Edmonds' min-max formula for the maximum size of a common independent set of two matroids on the same ground set (Theorem 13.31 in Korte and Vygen). Hint: use your reduction above and the property of the rank functions or M_1, M_2 that you proved in question a.

Question e :

Give a proof of Hall's Theorem (A bipartite graph $B = (U, V, E)$ has a perfect matching if and only if $|U| = |V|$ and for all $X \subset U$ the number of neighbours of X is at least the size of X) again based on Theorem 13.31 in Korte and Vygen. Note that you must do this directly. It is not sufficient to show how Hall's theorem follows from König's theorem.

Question f :

What is wrong with the following "algorithm" to find a maximum matching in a general graph $G = (V, E)$: Let $\mathcal{F} = \{E' \subset E :$ no vertex in V is incident to more than one edge in $E'\}$ and find a maximum cardinality set in $\mathcal{F}$ using the greedy algorithm?

Question g :

Describe briefly in words how to use matroid intersection to decide whether a given bipartite graph has a spanning (i.e. every vertex is covered) collection of vertex-disjoint cycles.