First set of exam problems for the course 'Combinatorial Optimization' (DM867)

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

The problems are available from the course page as of March 16, 2022 The solutions must be uploaded via itselearning no later than Friday April 8, 2022 at 15.00.

Note the following important points:

- You may work in groups of up to three. Each group uploads one copy of their solutions with birthdate and name of all members of the group.
- **Only one of Problems 6A and 6B may be handed in** (choose your favorite). Hence there are 125 points to earn.
- The second set of exam problems will have 75 points and must be done 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 as 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).
- **You may work in groups of size up to three, but any exchange of results/ideas between different groups (including one-person groups) before 15.00 April 8, 2022 will be considered as exam fraud.**

PROBLEM 1 (25 point)

This problem is about edge-connectivity in graphs. Notice that we allow multiple edges!

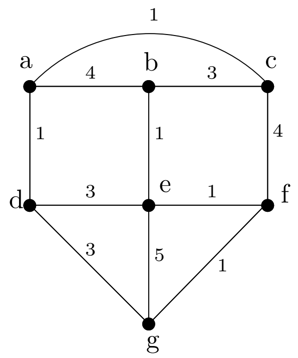

Figure 1: A graph with 7 vertices. The numbers on the edges indicate the number of parallel edges. Thus there are 4 edges between vertices a and b.

Question a:

Let $G = (V, E)$ be the graph in Figure 1. Show how to find the edge-connectivity $\lambda(G)$ of G via max-back orderings. You should start from vertex a in each iteration and give a max-back numbering of the current graph in each step.

Question b:

Give a small certificate $G' = (V, E')$ which shows that G is $\lambda(G)$-edge-connected. You must explain how you obtained the set of edges. The set E' may not have more than $\lambda(G)(|V| - 1)$ edges. Explain briefly (e.g. by referring to the appropriate result in the course material) why it is always possible to find such a certificate for every input graph $G = (V, E)$.

Question c:

Is G' above minimally $\lambda(G)$-connected?

Question d:

Find a Gomory Hu tree for the graph G. You should show all the steps that lead to the final tree. You may find the cuts by inspection in each step (and say which cut you consider next). You should be careful and check that your cut is actually a min cut, but you do not have to list a flow of the same value in your solution. Note that in all iterations you should follow the following rule:

- Always select that vertex of T which corresponds to a vertex set of size at least 2 in G and which contains the lexicographically smallest vertex of V (i.e. if a is in a non-trivial vertex X of T, you should select X to split).
- Always consider the lexicographically smallest pair of vertices when looking for a minimum cut between vertices in X (i.e. if $X = \{b, e, d, g\}$, then you should consider the pair b, d).

PROBLEM 2 (20point)

This problem is about the greedy algorithm for subset systems. Recall that $(S, \mathcal{F})$ is a subset system if $Y \in \mathcal{F}$ and $X \subseteq Y$ implies that $X \in \mathcal{F}$ (Definition 12.1 in Papadimitriou and Steiglitz).

Question a:

Start by describing the greedy algorithm on a subset system $(S, \mathcal{F})$ and a weight function $\omega : S \to \mathbf{R}$. Next give (without proof) a characterization of subset systems for which the greedy algorithm always works, no matter the choice of ω as long as ω is non-negative.

Question b:

For each of the problems below you should first show that one can associate a subset system to each of them and then either prove that the greedy algorithm always works, no matter the choice of (non-negative) weight function or show by an example or other correct method that the greedy algorithm does not always work for the problem.

(a) Given a set S, a natural number $k \leq |S|$ and non-negative weight function $\omega : S \to \mathbf{R}_+ \cup \{0\}$. Find a maximum weight subset $S' \subset S$ such that $|S'| \leq k$.

(b) Given an undirected graph $G = (V, E)$ and a non-negative weight function $\omega : E \to \mathbf{R}_+ \cup \{0\}$. Find a maximum weight subset E' of E such that the graph $H = (V, E - E')$ (the graph obtained by deleting the edges in E') is still connected.

(c) Given a digraph $D = (V, A)$ which has a cycle-factor, that is, a spanning collection of disjoint directed cycles and a non-negative cost function $\omega : A \to \mathbf{R}_+ \cup \{0\}$. Find a minimum cost spanning collection of vertex disjoint cycles, that is, a minimum cost subset of A of size $|V|$ which induces a collection of disjoint cycles in D.

(d) Let $x_1, x_2, \ldots, x_n$ and K be distinct non-negative integers. Find a maximum cardinality subset $I = \{i_1, i_2, \ldots, i_r\}$ of $\{1, 2, \ldots, n\}$ such that $\sum_{i \in I} x_i \leq K$.

(e) Given an undirected graph $G = (V, E)$ and a non-negative weight function $\omega : V \to \mathbf{R}_+ \cup \{0\}$. Find a maximum weight independent set of G (a set $X \subset V(G)$ is independent if no edge of G has both end-vertices in X).

(f) Given a bipartite graph $G = (X, Y, E)$, where X and Y are the two bipartition classes (all edges are between X and Y) and a non-negative weight function $\omega : X \to \mathbf{R}_+ \cup \{0\}$. Find a maximum weight subset $X' \subseteq X$ such that G has a matching M covering all vertices of X' (every vertex of X' is matched to a different vertex in Y by M).

PROBLEM 3 (15 point)

Question a:

Describe in words how one can obtain a polynomial algorithm for deciding whether a given graph $G = (V, E)$ contains a collection $C_1, C_2, \ldots, C_k$, $k \geq 1$ of vertex disjoint cycles such that $V = V(C_1) \cup \ldots \cup V(C_k)$. The algorithm should return a set of disjoint cycles covering V if one exists. You should explain the important steps in the algorithm and give the complexity.

Question b:

Suppose we are given a bipartite graph $B = (X, Y, E)$ and we want to know whether B has two edge-disjoint perfect matchings. Suggest an algorithm for solving this problem.

Question c:

Does the algorithm for Question b always work when B is not bipartite?

Question d:

Suppose that M is a matching in a bipartite graph $B = (X, Y, E)$ and that M is not a maximum matching. Describe a polynomial algorithm for finding a maximum matching which shares as many edges with M as possible.

PROBLEM 4 (20 point)

This problem is about increasing the edge-connectivity of a tree from 1 to 2 by adding new edges guided by a depth-first search and via the splitting off operation.

Let $T = (V, E)$ be the tree shown in Figure 2 with a DFS numbering of its 8 leaves.

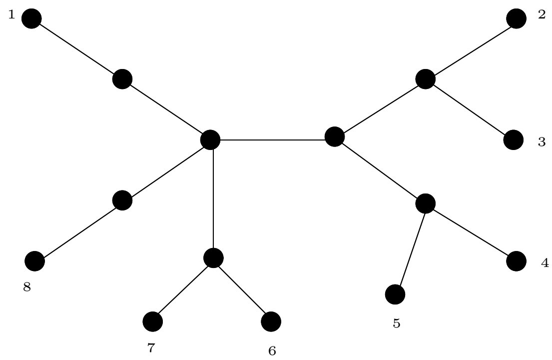

Figure 2: A tree T with a DFS numbering of its leaves

Question a:

Argue that we can obtain a 2-edge-connected graph from T by adding the 4 edges 15,26,37 and 48. Hint: follow the slides containing a solution to some of the problems on Weekly note 4.

Question b:

Will we also obtain a 2-edge-connected graph if we add the edges 18,27,36 and 45? ◇

Now we want to follow the approach of Frank (also on Weekly note 5) where we first add a new vertex s and two parallel edges from s to every vertex of the tree and the trim these by deleting edges that can be deleted without destroying the property that there are two edge-disjoint (x, y)-paths for every choice of $x, y \in V$.

Question c:

Argue that after the trimming (deletion phase) we will have $d(s) = 8$ so that s has precisely one edge to each of the leaves and no edges to any other vertex. ◇

Denote the graph we have right now by G_0, so G_0 is T plus the 8 edges si, $i = 1, 2, \ldots, 8$. The next step in Frank's algorithm is to perform $d(s)/2$ feasible splittings from s (replacing a pair of edges su, sv by the edge uv).

Question d:

Explain why the splitting $(s2, s3)$ is not feasible and then argue that the splitting $(s2, s4)$ is feasible.

Let G_1 be the graph after the splitting $(s2, s4)$.

Question e:

Argue that in G_1 the splitting $(s3, s5)$ is not feasible and that $(s1, s3)$ is a feasible splitting.

Let G_2 be the graph we obtain from G_1 after the splitting $(s1, s3)$.

Question f:

Argue that the splittings $(s5, s6)$ and $(s7, s8)$ are both feasible and show the resulting 2-edge-connected graph.

PROBLEM 5 (15 point)

Let $k \geq 2$ be an integer and let $G = (V + s, E)$ be a graph with a special vertex s so that

$$\lambda(x, y) \geq k \text{ for every choice of } x, y \in V \tag{1}$$

We say that a splitting (su, sv) is **feasible** if (1) holds for all $x, y \in V$, where G' is the graph we obtain by applying the splitting operation to the pair (su, sv). Recall that (1) is equivalent to

$$d(X) \geq k \text{ for every non-trivial subset } X \text{ of } V \tag{2}$$

The purpose of this problem is to prove the following extension of Lovász's splitting theorem from Weekly note 5:

Theorem 1 *Let $k \geq 2$ be an integer and let $G = (V + s, E)$ be a graph with a special vertex s such that $d(s)$ is positive and even and (1) holds for G. Then for every edge $e = su$ incident with s there exists at least $d(s)/2 - 1$ other neighbours v of s so that the splitting (su, sv) is feasible.*

We know from Weekly note 5 that there is at least one neighbour v of s so that the splitting (su, sv) is feasible, but we want to show that there are many more (unless $d(s)$ is small). Recall that a splitting (su, sv) is feasible if and only if there is no set $X \subset V$ containing u, v such that $d(X) \leq k + 1$ (such a set is **dangerous**).

Question a:

Show, using the same type of argument as we used on Weekly note 5, that if we can cover u and all the neighbours v of s for which the splitting (su, sv) is not feasible by just one dangerous set X, then there are at least $d(s)/2$ neighbours v of s for which the splitting (su, sv) is feasible. Hint: remember that $d(s)$ is even.

Question b:

Show that if we cannot cover all the neighbours v of s for which the splitting (su, sv) is not feasible by just one dangerous set X, then we can cover all of them by just two dangerous sets. Hint: prove, using similar arguments as we did in the weekly note, that if there we needed three or more dangerous sets to cover all neighbours v of s for which the splitting (su, sv) is not feasible, then some set containing u would have degree just 1.

Question c:

Let X, Y be two dangerous sets whose union cover all the neighbours v of s for which the splitting (su, sv) is not feasible. Prove that the number of neighbours of s in $V - (X \cup Y)$ is at least $d(s)/2 - 1$ and explain why this proves the theorem.

PROBLEM 6A (30 point)

The purpose of this problem is to give a proof of the following result:

Theorem Let $G = (V, E)$ be an undirected graph and let $s \in V$ be fixed. Then G can be oriented as a digraph D which satisfies

$$d_D^-(X) \geq k \text{ for all } \emptyset \neq X \subseteq V - s \tag{3}$$

if and only if for every partition[1] $\mathcal{P} = \{V_1, V_2, \ldots, V_t\}$ of V we have

$$e_{\mathcal{P}} \geq k(t-1), \tag{4}$$

where $e_{\mathcal{P}}$ denotes the number of edges in G which connect different sets in $\mathcal{P}$.

Question a:

Prove that the condition (4) is necessary for the existence of an orientation satisfying (3). Hint: every set not containing s must have in-degree at least k. ◇

In the rest of this problem we will prove the other direction and derive a polynomial algorithm for achieving such an orientation, provided that (4) holds for G. Below we shall always assume that (4) holds for G.

By adding a set F of sufficiently many new edges between s and $V - s$ we can obtain a supergraph $G' = (V, E \cup F)$ such that G' has an orientation D' satisfying

$$d_{D'}^-(X) \geq k \text{ for all } \emptyset \neq X \subseteq V - s \tag{5}$$

In particular, adding k parallel edges between s and v for each $v \in V - s$ suffices, since then we just orient all these **new edges** out of s and all original edges arbitrarily and clearly (5) is satisfied. The goal is now to show how to get rid of all the extra edges that we added one by one by reversing along directed paths in D' until we reach an orientation D of G which satisfies (3). Below we will always denote by D' the oriented graph which we have currently.

Question b:

Prove that if in the currently existing orientation D' there is any new edge $sv \in F$ which is oriented from v to s, then we may delete the edge sv from G' (and the arc vs from D') without violating (5). ◇

Similarly, we may assume that in D' all original edges (an edge e is **original** if $e \in E$) incident with s are oriented out of s. Hence we may assume below that all edges incident to s are oriented out of s in D' and, since we are not done yet, that there is still at least one new edge sv incident with s. Let T be the set of those vertices that can be reached from v via a directed path in D'. By the remark above $s \notin T$ and

$$d_{D'}^-(W) = 0, \tag{6}$$

where $W = V - T$.

Let us call a set $X \subseteq V - s$ **tight** if $d_{D'}^-(X) = k$.

Question c:

Prove that if $X$and Y are tight sets with $X \cap Y \neq \emptyset$ then $X \cap Y$ and $X \cup Y$ are also tight. Then argue that if X and Y are maximal tight sets (with respect to inclusion) then X and Y are disjoint. ◇

[1] Recall that a partition of a set S is a collection of disjoint subsets of S whose union is S.

Question d:

Prove that for every tight set X we have $X \subseteq T$ or $X \cap T = \emptyset$. Hint: Suppose $X \cap T \neq \emptyset$ and $X \cap W \neq \emptyset$. Use (6) and Proposition 7.1.1 in "Digraphs" (Bang-Jensen and Gutin) to show that $v \in X \cap T$ must hold and then use the arc sv to derive a contradiction. $\diamond$

Suppose that every $u \in T$ is contained in a tight set and let $V_1, V_2, \ldots, V_{t-1}$ be a partition of T into maximal tight sets such that $v \in V_1$. Let $V_t = W$ and let $\mathcal{P}$ be the partition $\mathcal{P} = \{V_1, V_2, \ldots, V_t\}$ of V. Using $e'_{\mathcal{P}}$ to denote the number of edges in G' which connect different sets in $\mathcal{P}$ we have (using that each V_i is tight for $i = 1, 2, \ldots, t-1$):

$$\begin{aligned} k(t-1) &= \sum_{i=1}^{t} d^-_{D'}(V_i) \\ &= e'_{\mathcal{P}} \\ &> e_{\mathcal{P}}, \end{aligned}$$

since the arc sv contributes to $d^-_{D'}(V_1)$ and $e'_{\mathcal{P}}$ but not to $e_{\mathcal{P}}$. This contradicts that (4) holds for G and hence there must exist a vertex $u \in T$ such that $d^-_{D'}(X) \geq k+1$ for every $X \subseteq V - s$ containing u. Now let P be any path from v to u in D'.

Question e:

Argue that reversing the orientation of all arcs on P and deleting the arc sv we obtain a new orientation (which we also call D') such that (5) still holds. Hint: examine how the reversal affects to the in-degree of a set. $\diamond$

Repeating the process above until all edges from F have been deleted, we obtain the desired orientation D of G.

Question f:

Based on the steps above describe a polynomial algorithm which given a graph $G = (V, E)$, a vertex $s \in V$ and natural number k either finds an orientation D of G satisfying (3) or produces a partition $\mathcal{P}$ of V which violates (4). Remember to state the complexity of your algorithm (does it depend on k?). Hint: If the process above does not produce the desired orientation, then every vertex in T must lie in a tight set with respect to the current orientation. Now use this fact to produce a partition that violates (4). $\diamond$

PROBLEM 6B (30 point)

This problem is about orientations of graphs. By an **orientation** of a graph $G = (V, E)$ we mean a directed graph $D = (V, A)$ that we can obtain from G by assigning precisely one of the orientations $u \to v$ and $v \to u$ to every edge $uv \in E$. At the lectures I proved the following theorem (see the notes on Weekly note 6):

Theorem 2 (Nash-Williams) *Let $k \geq 1$ be an integer and let G be a $2k$-edge-connected graph. Then G has a k-arc-strong orientation D.*

We say that an orientation $D = (V, A)$ of a graph $G = (V, E)$ is **balanced** if $|d^+(v) - d^-(v)| \leq 1$ holds for every vertex $v \in V$. That is, the in-degree and the out-degree of each vertex are at most one apart (and they are equal if $d_G(v)$ is even).

The purpose of this exercise is the show the following small extension of Nash-Williams' theorem and develop a polynomial algorithm for constructing such an orientation of a given $2k$-edge-connected graph.

Theorem 3 *Let $k \geq 1$ be an integer and let G be a $2k$-edge-connected graph. Then G has a k-arc-strong balanced orientation D.*

Question a:

Show that if $D = (V, A)$ is k-arc-strong then the following holds

- If C is a directed cycle in D, then the digraph $D(\overleftarrow{C})$ that we obtain from D by reversing all arcs of C is also k-arc-strong.
- If P is a directed path from s to t in D, then the digraph $D(\overleftarrow{P})$ that we obtain from D by reversing all arcs of P is also k-arc-strong, unless there is a set X with $s \in X$, $t \notin X$ and $d^+_D(X) = k$

Question b:

Use the observation above to prove that if D is a strongly connected balanced orientation of a graph G and we add a new edge st to G, then we can orient the new edge so that the new oriented graph D' is also balanced or can be made balanced by reversing a path between s and t in D.

Question c:

Repeat the proof of Theorem 2 that I gave on Weekly note 6 and explain how you can can modify some of the steps so that the resulting orientation becomes not only k-arc-strong but also balanced. You should give enough detail so that it is clear what you do in each step.

Question d:

Explain how to do each of the steps algorithmically and give the complexity of the final algorithm for constructing a balanced k-arc-strong orientation of a given $2k$-edge-connected graph G. $\diamond$

Recall that a graph is **eulerian** if it is connected and the degree of every vertex is even. Recall that every eulerian graph has an **Euler tour** W, that is, a closed tour that starts in some vertex v, uses each edge of G precisely once and returns to v.

Question e:

Prove that every eulerian graph $G = (V, E)$ has a balanced orientation D for which we even have $\lambda_D(x, y) = \lambda_D(y, x) = \lambda_G(x, y)/2$ for every pair of vertices $x, y \in V$. Hint: use an Euler tour W to construct a good orientation.

Question f:

Now let D and D' be two different k-arc-strong orientations of the same graph $G = (V, E)$ so that $d^+_D(v) = d^+_{D'}(v)$ for all vertices $v \in V$. Prove that there exist a collection of arc-disjoint cycles $C_1, C_2, \ldots, C_r$, $r \geq 1$ in D so that we can obtain D' from D by reversing all the arcs in the cycles $C_1, C_2, \ldots, C_r$. Hint: consider D' as coming from a flow on D and use flow decomposition.