Exam problems for the course 'Network Programming' (DM817)

Department of Mathematics and Computer Science
University of Southern Denmark

The problems are handed out Thursday May 14. 2009. The solutions must be returned by Wednesday June 17, 2009

Note that exactly one of problems 4A and 4B can be handed in. Thus there are 100 points to earn.

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 can understand it (but you do not have to give pseudo code). You should also give the complexity of the algorithm when relevant. Note also that illustrating an algorithm means that one has to follow the steps of the algorithm meticulously (slavisk).

It is strictly forbidden to work in groups and any exchange of results before June 17, 2009 will be considered as exam fraud.

PROBLEM 1 (16 point)

Question a:

Give a short description of the capacity scaling algorithm and illustrate it by applying it to find a maximum $(1, 8)$-flow in the network in Figure 1.

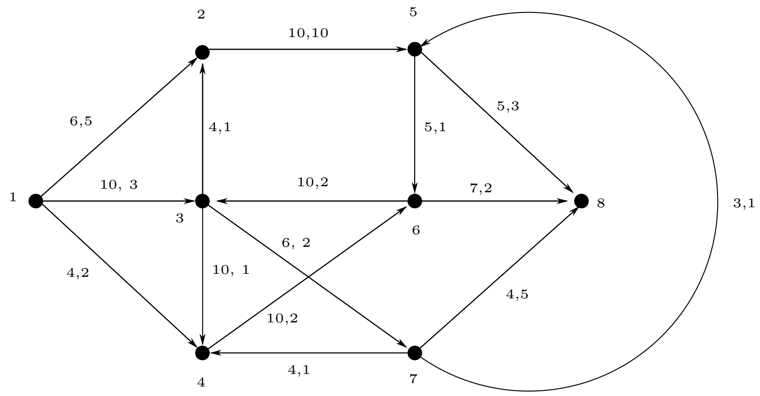

Figure 1: A network with capacities and costs on the arcs. The data is given as u_{ij}, c_{ij}.

Question b:

Show how to find a maximum $(1, 8)$-flow of minimum cost. You may choose any min cost flow algorithm you like.

PROBLEM 2 (24 point)

Let $D = (V, A)$ be a digraph. Recall that a cycle factor of D is a collection of vertex disjoint cycles which cover all vertices of D.

Question a:

Explain how we can check using flows whether D has a cycle factor. You may assume that you have an algorithm for finding a maximum (s, t)-flow in a network and you should show how to use this algorithm as a subroutine to decide the existence of a cycle factor in D.

Recall that D is a regular digraph if there exists a natural number k such that $d^+(x) = d^-(x) = k$ for all $x \in V(D)$.

Question b:

Prove that every regular digraph D has a cycle factor and give the fastest algorithm you can think of for finding such a cycle factor in a given regular digraph D.

Question c:

Describe a polynomial algorithm for deciding, for a given input digraph D and a natural number r, whether D has r arc-disjoint cycle factors. You must also show how to obtain the r arc-disjoint cycle factors. Hint: the union of the arcs of these will induce a regular subdigraph in D.

Question d:

Explain how one can obtain a polynomial algorithm for finding, in a digraph D which has a cycle factor $\mathcal{F}$, a second cycle factor $\mathcal{F}'$ which shares the minimum possible number (over all cycle factors of D) of arcs with $\mathcal{F}$.

Suppose now that the input digraph D does not have r arc-disjoint cycle factors and that we now allow that an arc is used in several cycle factors at a certain price.

Question e:

Suppose first that we wish to minimize the maximum number of times any arc is used in a collection of r distinct cycle factors of D. Explain how to solve this problem in polynomial time.

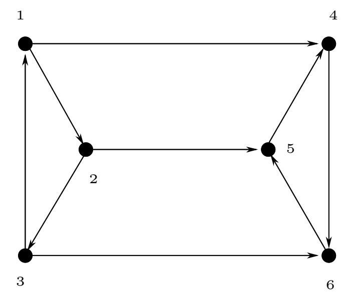

Figure 2: A digraph which is not 2-arc-strong.

Question f:

Now suppose that the price for using an arc ij p times is $(p-1)^2$ when $p \geq 1$ (that is, using it only once costs nothing). Explain how to use flows to find a collection of r cycle factors which minimize this measure (summed over all arcs).

PROBLEM 3 (16 point)

Question a:

Explain how to decide for a given input digraph D and a natural number k, whether D is k-arc-strong. Give the best complexity you can find for this problem using the algorithm you propose.

Assume below that D is not k-arc-strong.

Question b:

Describe the main steps in Frank's algorithm for augmenting the arc-strong connectivity of D to k and explain how this algorithm can be implemented using a maximum flow algorithm as a subroutine. You do not have to prove the correctness of Frank's algorithm.

Question c:

Illustrate the algorithm on the digraph in Figure 2 when $k = 2$. You do not have to give all details. It is sufficient to explain in words how you decide which arcs can be deleted (to and from the new vertex s) and which splittings are legal (e.g. based on values of $\lambda(i, j)$ for suitable pairs of vertices i, j).

PROBLEM 4A (16 point)

This problem deals with the primal-dual algorithm for the transportation problem.

Question a:

Explain in words how the algorithm works and how one can verify optimality once the algorithm terminates.

						a_i
	5	9	12	3	4	6
	8	5	12	14	5	7
	12	7	6	6	6	9
	5	4	3	2	6	3
b_j	8	6	5	3	3	

Figure 3: An instance of the transportation problem

Question b:

Illustrate the algorithm by applying it to the transportation problem shown in Figure 3

Question c:

Suppose now that one of the a_i's is decreased by 1 and that one of the b_j's is also decreased by 1. Explain how to re-optimize the optimal solution (for the general transportation problem) faster than starting a new calculation from scratch. Illustrate this method starting from the optimal solution you found for the example in Figure 3 when we let $a_3 = 8$ and $b_2 = 5$.

PROBLEM 4B (16 point)

This problem also deals with cycle factors in digraphs. Suppose we are given a digraph $D = (V, A)$ and that for each vertex $v \in V$ we are also given a (possibly empty) list of pairs $(u_1v, vw_1), ..., (u_{r_v}v, vw_{r_v})$ of arcs from A (with repetition of arcs allowed) which we call **forbidden**. The problem is to decide whether D has a cycle factor $\mathcal{F}$ such that none of the forbidden pairs are used in $\mathcal{F}$. That is, if u_iv, vw_i is a forbidden pair, then no cycle in $\mathcal{F}$ may use both of the arcs u_iv, vw_i.

Describe a polynomial algorithm which given a digraph D and forbidden pairs as above decides whether D has a cycle factor which avoids all these forbidden pairs. Give the best complexity you can find for the algorithm. Hint: use a variant of vertex splitting.

PROBLEM 5 (16 point)

Let $\mathcal{N}' = (V', A', \ell \equiv 0, u \equiv \infty, b)$ be a network where all lower bounds are zero, all capacities are infinite and the sum over all balance constraints is zero ($\sum_{i \in V'} b(i) = 0$).

Question a:

Prove that there exists a feasible flow x in $\mathcal{N}'$ (i.e. $b_x \equiv b$ and $0 \leq x_{ij} \leq u_{ij}$ for all $ij \in A'$) if and only if there is no subset S of V' such that $\sum_{i \in S} b(i) > 0$ and there is no arc of A' that goes out of S (that is, the out-degree of S is zero in the underlying digraph of the network $\mathcal{N}'$). Show how to find a feasible flow or a violating set S using one max-flow calculation in a suitable network. Hint: compare with Ahuja Section 19.2.

Consider now a network $\mathcal{N} = (V, A, \ell \equiv 0, u, b)$ which has a feasible flow $\hat{x}$. The goal is to find such a feasible flow which uses as few arcs as possible, that is, we want to minimize the number of arcs ij where $x_{ij} > 0$.

Question b:

Explain why the following is a correct network design model for the problem.

$$\begin{aligned}
&\textit{minimize} && z = \textstyle\sum_{ij \in A} y_{ij} && && (1)\\
&\text{subject to} && b_x(i) = b(i) && \text{for all } i \in V && (2)\\
& && 0 \le x_{ij} \le u_{ij} y_{ij} && \text{for all } ij \in A && (3)\\
& && y_{ij} \in \{0, 1\} && \text{for all } ij \in A && (4)
\end{aligned}$$

Question c:

Apply the Lagrangian relaxation technique to the problem above by relaxing the constraint (3) using a Lagrangian multiplier μ_{ij} for each $ij \in A$. Write up the resulting optimization problem $z(\mu)$ and explain for which values of the Lagrangian multipliers you get a valid lower bound for the minimum value of z in (1).

Question d:

Explain how to solve the Lagrangian problem $z(\mu)$ for a fixed vector μ of Lagrangian multipliers. Hint: the problem splits into two independent problems.

PROBLEM 6 (14 point)

Suppose we are given a network $\mathcal{N} = (V, A, \ell, u, b)$

Question a:

Show how to check whether $\mathcal{N}$ has a feasible flow x. Your algorithm must not be slower than the fastest max-flow algorithm we have covered in the course.

Suppose now that $\mathcal{N}$ does not have a feasible flow.

Question b:

Give a short argument that if we forget the balance constraints b, then there is always a feasible flow with respect to the lower bounds.

Suppose also that without the lower bound constraints there is a feasible flow which meets the balances and respects the capacities. Now we are interested in finding the minimum total amount we have to decrease the lower bounds in order obtain a network $\mathcal{N}' = (V, A, \ell', u, b)$ which has a feasible flow. That is, we want to minimize $\sum_{ij \in A}(\ell_{ij} - \ell'_{ij})$

Question c:

Formulate this problem as a convex cost flow problem. Hint: compare with Ahuja Exercise 14.4.