Exam problems for the course 'Flows in Networks' (DM33)

Department of Mathematics and Computer Science
University of Southern Denmark

The problems are handed out Monday May 22. 2000. The solutions must be returned by Friday August 4. 2000

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. 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 August 5 will be considered as exam fraud.

Note that only one of the problems 5A and 5B may be handed in. Thus there are 100 points to earn and a complete solution corresponds to 100 points.

PROBLEM 1 (20 point)

Question a:

Give a short description of the capacity scaling algorithm for finding a maximum (s, t)-flow in a network and illustrate the algorithm by applying it to the network in Figure 1

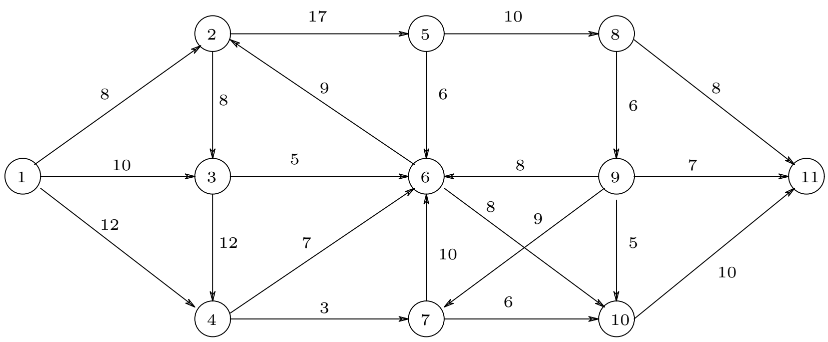

Figure 1: A network with capacities shown and all lower bounds zero. The source is the vertex 1 and the sink is the vertex 11.

Question b:

Give a short description of Dinic's algorithm for finding a maximum (s,t)-flow in a network and illustrate the algorithm by applying it to the network in Figure 1

Question c:

Which of the algorithms above has the best complexity for unit capacity networks?

PROBLEM 2 (10 point)

Let x be a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let y be a feasible flow in $\mathcal{N}(x)$. Show that $\mathcal{N}(x \oplus y) = \mathcal{N}(x)(y)$, where $\mathcal{N}(x)(y)$ denotes the residual network of $\mathcal{N}(x)$ with respect to y. That is, show that the two networks contain the same arcs and with the same residual capacities.

PROBLEM 3 (15 point)

Tree solution to a flow problem. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with n vertices for which there exists a feasible flow and let $D = (V, A)$ be the digraph corresponding to $\mathcal{N}$ (all data removed). Prove that there exists a feasible flow x in $\mathcal{N}$ such that the number of arcs on which $0 < x_{ij} < u_{ij}$ is at most $n-1$. Hint: Consider cycles in $UG(D)$ where $0 < x_{ij} < u_{ij}$ for every arc on the cycle. Here $UG(D)$ denotes the underlying undirected graph of D, that is the graph which has an edge ij whenever D has at least one of the arcs $i \rightarrow j$, $j \rightarrow i$.

PROBLEM 4 (30 point)

A primal-dual algorithm for minimum cost flows. We say that a feasible flow x is an optimal flow in a network $\mathcal{N} = (V, A, l \equiv 0, u, c)$ if x has minimum cost among all flows with the same balance vector as x. Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be a network with source s and sink t for which the value of a maximum (s,t)-flow is $K > 0$. Let x be an optimal feasible (s,t)-flow of value $k < K$ in $\mathcal{N}$ and let $\pi : V \rightarrow \mathcal{R}$ be chosen such that $c^{\pi}_{ij} \geq 0$ for every arc ij in $\mathcal{N}(x)$. Define A_0 as those arcs ij of $\mathcal{N}(x)$ for which we have $c^{\pi}_{ij} = 0$ and let $\mathcal{N}_0$ be the subnetwork of $\mathcal{N}(x)$ induced by the arcs of A_0.

Question a:

Show that if y is a feasible (s,t)-flow in $\mathcal{N}_0$ of value p then $x' = x \oplus y$ is an optimal (s,t)-flow of value $k + p$ in $\mathcal{N}$.

Question b:

Suppose y is a maximum (s,t)-flow in $\mathcal{N}_0$, but $x' = x \oplus y$ has value less than K. Let S denote the set of vertices which are reachable from s in $\mathcal{N}_0(y)$. Let $\epsilon, \epsilon_1, \epsilon_2$ be defined as follows. Here we let $\epsilon_i = \infty$ if there are no arcs in the corresponding set, $i = 1, 2$.

$$
\begin{aligned}
\epsilon_1 &= \min\{c^{\pi}_{ij} | i \in S, j \in \overline{S}, c^{\pi}_{ij} > 0 \text{ and } x_{ij} < u_{ij}, \} \\
\epsilon_1 &= \min\{-c^{\pi}_{ij} | i \in \overline{S}, j \in S, c^{\pi}_{ij} < 0 \text{ and } x_{ij} > 0\}, \\
\epsilon &= \min\{\epsilon_1, \epsilon_2\}.
\end{aligned}
$$

Argue that $\epsilon < \infty$.

Question c:

Now define π' as follows: $\pi'(v) := \pi(v) + \epsilon$ if $v \in S$ and $\pi'(v) := \pi(v)$ if $v \in \overline{S}$. Let $\mathcal{N}'_0$ contain those arcs of $\mathcal{N}(x')$ for which we have $c^{\pi'}_{ij} = 0$ and let S' denote the set of vertices which are reachable from s in $\mathcal{N}'_0$. Show that S is a proper subset of S' and that $c^{\pi'}_{ij} \geq 0$ holds for all arcs ij of $\mathcal{N}(x')$. Hint: use the result of Problem 2.

Question d:

If $t \notin S'$, then we can change π' as above (based on the set S' instead of S). Conclude that after at most $n - 1$ such updates of the vector π', the current network $\mathcal{N}'_0$ contains an (s, t)-path.

Question e:

Use the observations above to design an algorithm that starts with $x \equiv 0$ and then finds a minimum cost (s, t)-flow of value K in $\mathcal{N}$ by solving a sequence of maximum flow problems.

Question f:

What is the complexity of the algorithm in Question e?

Question g:

Argue that all arcs ij considered when ϵ_1 is calculated have $x_{ij} = 0$ and that all arcs ij considered when ϵ_2 is calculated have $x_{ij} = u_{ij}$.

PROBLEM 5A (25 point)

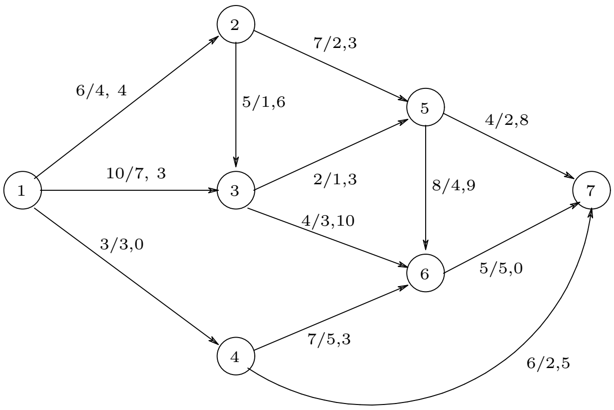

Figure 2: The graph corresponding to a project with 12 jobs. Each arc corresponds to a job with data given as normal time/minimum time, cost of shorting the execution time of the job by one time unit.

In Figure 2 a project plan involving 12 jobs is given

Question a:

Find the completion time of the project when all jobs are executed at their normal time and when all jobs are executed at their minimum time.

Question b:

Give all steps in the calculation of the project shortening cost curve for the project in Figure 2, when you use the method from Murty Section 7.2. In each step of the calculation (finding the next slope of the curve) you must show the admissible arcs and explain how you calculate the next slope and the interval in which it applies. Explain also how one detects (from the flow algorithm) that the calculation of the curve is completed. Finally, draw the curve you have found.

Question c:

Find the execution times for all jobs when the project is completed at minimum time. Use these execution times to verify that your curve was calculated correctly.

Question d:

Find the execution times given by the method when the project is to be completed after exactly 20 time units. Find the critical jobs when the project is completed after exactly 20 time units.

Question e:

How much can we shorten the project time if we have 40 units of money available?

PROBLEM 5B (25 point)

In this problem we study arc-connectivity of directed graphs (recall Section 6.6. in Ahuja, Orlin and Magnanti). For an arc $a = s \to t$ of a digraph we call the vertex s the tail of a and the vertex t the head of a. If $D = (V, A)$ is a digraph and $X \subset V$, we denote by $d_D^+(X)$ the number of arcs with tail in X and head in $V - X$. Similarly, $d_D^-(X)$ is the number of arcs with head in X and tail in $V - X$. A spanning subdigraph of a digraph $D = (V, A)$ is a digraph $D' = (V, A')$ such that $A' \subseteq A$. That is, D' has the same vertices as D and is obtained from D by deleting zero or more arcs. (In particular, the digraph $D_\emptyset = (V, \emptyset)$ where we have deleted all arcs is a spanning subdigraph of D. Note also that if D' is a strong spanning subdigraph of $D = (V, A)$, then every vertex of V has at least one arc entering and at least one arc leaving in D'.)

We say that a digraph D is k-arc-strong if $d_D^+(X) \geq k$ holds for every subset X of V such that $X \neq \emptyset$ and $X \neq V$. Let $\lambda(D)$ be the maximum integer k such that D is k-arc-strong. Below we always assume that $k = \lambda(D)$. An arc $a = s \to t$ is critical (for k-arc-strong connectivity) if $\lambda(D - a) < k$. Here $D - a$ is the digraph obtained from D by deleting the arc a.

A digraph $D = (V, A)$ is minimally k-arc-strong if $\lambda(D) = k$ but $\lambda(D - a) < k$ for every $a \in A$. That is, every arc in A is critical.

Question a:

Show that $a = s \to t$ is critical if and only if there exists a set X with $s \in X$ and $t \in V - X$ and $d_D^+(X) = k$.

Question b:

Show how to decide in time $O(k|A|)$ whether a given arc a is critical in a digraph $D = (V, A)$.

Question c:

Describe an $O(k|A|^2)$ algorithm to find a minimally k-arc strong spanning subdigraph of a given input digraph $D = (V, A)$.

Question d:

Does every digraph have a unique minimally k-arc strong spanning subdigraph?

Question e:

Suppose $\lambda(D) = k$ and that $a = s \to t$ is a critical arc of D. Prove that in $D - a$ there exists a unique minimal set (with respect to inclusion) X such that $s \in X$, $t \in V - X$ and $d_{D-a}^+(X) = k-1$ and a unique minimal set Y such that $t \in Y$, $s \in V - Y$ and $d_{D-a}^-(X) = k-1$.