Prof. Friedrich Eisenbrand — Due Date: June 03, 2010
Martin Niemeier — Discussions: May 20, May 27

Discrete Optimization

Spring 2010

Solutions 6

You can hand in written solutions for up to two of the exercises marked with (*) or (Δ) to obtain bonus points. The duedate for this is June 03, 2010, before the exercise session starts. Math students are restricted to exercises marked with (*). Non-math students can choose between (*) and (Δ) exercises.

Exercise 1

The Lucky Puck Company has a factory in Vancouver that manufactures hockey pucks, and it has a warehouse in Winnipeg that stocks them. Lucky Puck leases space on trucks from another company to ship the pucks from the factory to the warehouse. Because the trucks travel over specified routes between cities and have a limited capacity, Lucky Puck can ship at most $c(u,v)$ crates per day between each pair of cities u and v. Lucky Puck has no control over these routes and capacities and so cannot alter them. Their goal is to determine the largest number p of crates per day that can be shipped from the factory to the warehouse.

Show how to compute p by finding a maximum flow in a network.

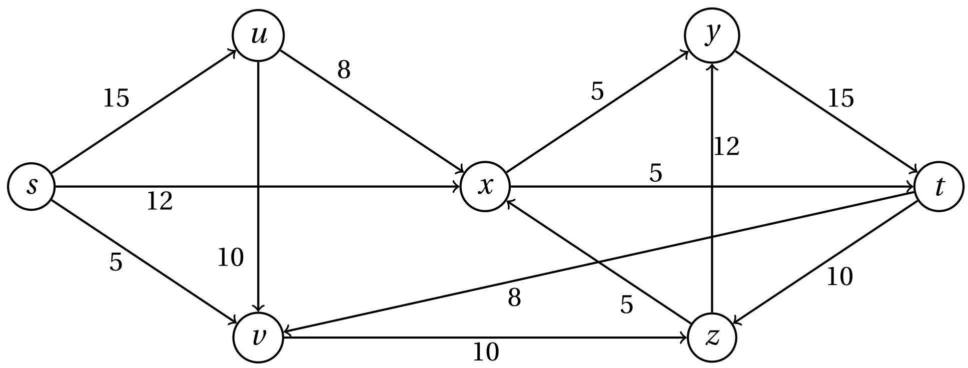
Solution

Let V be the set of cities that are possible starting and endpoints for the trucks. The set of arcs A is defined as follows. For each pair $u,v \in V$ of cities, we have $(u,v) \in A$ if and only if $c(u,v) > 0$. Let $D = (V,A)$ be our network (with capacities c).

Observe that every shipping strategy corresponds directly to a $Vancouver - Winnipeg$-flow in D and vice-versa. Hence, the number p is the value of a max $Vancouver - Winnipeg$-flow in V.

Exercise 2

Consider the following network:

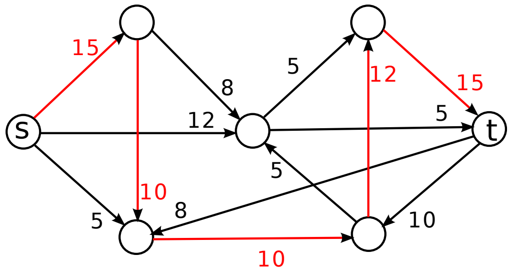


Run the Ford-Fulkerson algorithm to compute a max $s-t$-flow. For each iteration give the residual network and mark the path you choose for augmentation.

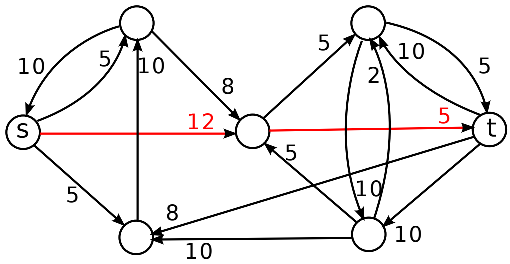
Further give a minimum $s-t$-cut in the network.

Solution

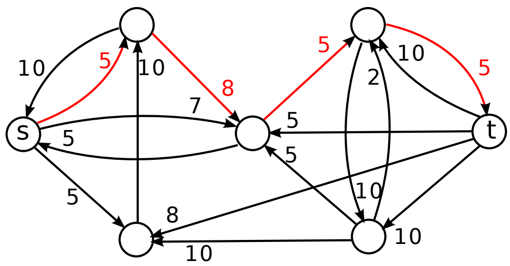
The first residual network is idential to the original network. The $s-t$-flow we choose to augment is marked red. We can augment 10 units of flow:



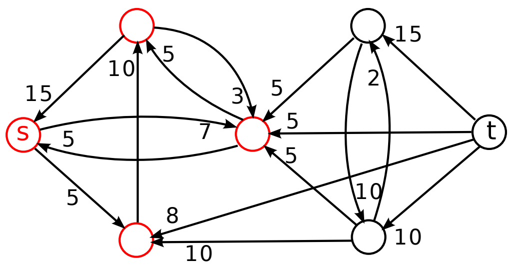
The residual network after augmentation looks as follows. The next augmenting path is marked red. We can augment 5 units of flow.



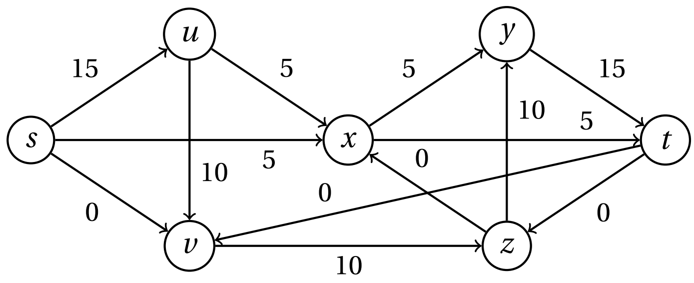
The residual network after augmentation looks as follows. The next augmenting path is marked red. We can augment 5 units of flow.



The residual network after augmentation looks as follows:



There is no more $s-t$-path in the residual network. Thus our flow is maximal and of value 20. The nodes marked red are reachable from s. Thus $\{s, u, v, x\}$ defines a minimal $s-t$-cut. The flow is as follows:



Exercise 3 (*)

Recall that an *undirected graph* $G = (V, E)$ is a set of *nodes* together with a set of *edges* $E \subseteq \{\{u, v\} : u, v \in V\}$. G is *connected* if for each pair of nodes $u, v \in V$ there is a path from u to v.

The *edge connectivity* of an undirected graph is the minimum number k of edges that must be removed such that the resulting graph is not connected anymore.

Show how the edge connectivity of an undirected graph $G = (V, E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ arcs.

Hint: Consider the bidirected graph $D = (V, A)$, where $A = \{(u, v) : \{u, v\} \in E\}$ (for each edge {u,v} we have the arcs (u, v) and (v, u)). Is there any relation between the edge connectivity of G and minimum size cuts in D?

Solution

For an undirected graph $G = (V, E)$ and $U \subseteq V$, let $\delta(U) = \{e \in E : e = \{u, v\}$ for some $u \in U, v \notin U\}$. Let k be the edge connectivity number of G.

We claim that

$$k = \min_{U \subset V} |\delta(U)|. \qquad (1)$$

To see that $k \leq \min_{\emptyset \subset U \subset V} |\delta(U)|$, observe that for each $\emptyset \subset U \subset V$, if we remove all edges in $\delta_G(U)$ from G, the graph becomes disconnected.

To see that $k \geq \min_{\emptyset \subset U \subset V} |\delta_G(U)|$, let $M \subseteq E$, $|M| = k$, be a set of edges such that $G' = (V, E \setminus M)$ is disconnected. Let $s \in V$ and let U be the set of nodes reachable from s in G'. Then $\delta_G(U) \subseteq M$ and thus $\min_{\emptyset \subset U \subset V} |\delta_G(U)| \leq k$.

This shows (1). Thus it is sufficient to find a minimum size cut $\delta_G(U)$ in G.

Now consider the bidirected graph $D = (V, A)$, where $A = \{(u, v) : \{u, v\} \in E\}$ with unit capacities $c : A \to \mathbb{Z}_{\geq 0}$, $c(a) = 1 \ \forall a \in A$. Observe that for each $U \subseteq V$ we have $|\delta_G(U)| = |\delta_D^{out}(U)| = |\delta_D^{in}(U)|$. Thus it is sufficient to find a minimum size cut in D.

Let $\emptyset \subset U \subset V$ such that $|\delta_G(U)| = |\delta_D^{out}(U)|$ is minimal. Fix an arbitrary node $s \in V$. If $s \in U$, then for all $t \in V \setminus U$, $\delta_D^{out}(U)$ is an $s-t$ cut. If $s \notin U$, then $\delta_D^{out}(V \setminus U)$ is an $s-t$ cut for all $t \in U$. Since $|\delta_D^{out}(V \setminus U)| = |\delta_D^{in}(U)| = |\delta_D^{out}(U)|$, $\delta_D^{out}(V \setminus U)$ is a minimum size cut as well.

Since the size of a min $s-t$ cut equals the value of a maximum $s-t$ flow (Max-Flow-Min-Cut theorem), this shows that if we fix some node $s \in V$ and solve the max $s-t$ flow problem for each $t \in V$, $t \neq s$, the minimal cut found in all these iterations gives the capacity of a min cut and therefore the connectivity number k of G.

Exercise 4 (Δ)

A *matching* in an undirected graph $G = (V, E)$ is a subset $M \subseteq E$ of the edges such that no two edges in M share a common node of V.

The *matching-problem* is to find a matching M of maximum cardinality.

Explain how to solve the matching-problem on a *bipartite graph* computing a maximum flow in an auxiliary network. What is the running time of your algorithm?

Solution

Let A and B be a partition of the nodes of G into two stable sets. Make the graph directed by directing the edges in such a way that all tails are in A.

Augment the graph with two supernodes s and t and introduce arcs (s, v) for all $v \in A$ and (v, t) for all $v \in B$. Put a capacity of 1 on each arc of the network. Call the graph D.

Observe that each matching of cardinality k in G induces an $s-t$ flow in D of value k and vice versa.

Running time: Since the value of a max flow is bounded by n, there are at most n augmentations. As seen in the lecture, each augmentation has a running time of $O(m)$. Hence the running time is bounded by $O(nm)$.

Exercise 5 ($*$)

Consider a directed graph $G = (V, A)$, a node $s \in V$, a cost function $c\colon\ A \to \mathbb{N}_0$ and a benefit function $b\colon\ V \to \mathbb{N}_0$.

If you destroy a set $M \subseteq A$ of arcs, you have to pay a cost of $c(D)$. Let S_M be the set of nodes *not* reachable from s after removing the arcs of M from the graph. You receive a benefit of $b(S_M)$ for destroying the edges D.

Solve the problem of finding an arc set M such that $b(S_M) - c(M)$ is maximized using a maximum flow (min cut) algorithm. Prove that your construction is correct.

Solution

Construct a network as follows: Let $D := (V', A')$, where $V' := V + t$ and $A' := A \cup \{(v, t)\ :\ v \in V\}$. Define a capacity function $u\colon\ A' \to \mathbb{N}_0$ as:

$$u(a) := \begin{cases} c(a), & \text{if } a \in A \\ b(v), & \text{if } a = (v,t),\ v \in V. \end{cases}$$

Now consider a solution to the problem, i.e. let $M \subseteq A$ be an arc set, and S_M be the nodes not reachable after removing M from G. Let $S := V \setminus S_M$, i.e. the set of nodes reachable from s. Observe that by definition of S and M,

$$\delta_G^{out}(S) \subseteq M$$

holds. Hence we have

$$\begin{aligned}
b(S_M) - c(M) &\le b(S_M) - c(\delta_G^{out}(S)) \\
&= b(V) - (b(S) + c(\delta_G^{out}(S))) \\
&= b(V) - (\sum_{a \in A', a=(v,t), v \in S} u(a) + u(\delta_G^{out}(S))) \\
&= b(V) - u(\delta_D^{out}(S))),
\end{aligned}$$

the last equality holds by construction of the network D. This shows that for each solution of value k, we can find an $s-t$-cut in D of capacity at most $b(V) - k$.

Conversely, if we have an $s-t$-cut $\delta_D^{out}(S)$ in D, it gives rise to a solution of the problem of value at least $b(V) - u(\delta_D^{out}(S)$, because setting $M := \delta_G^{out}(S)$ we have

$$b(V) - u(\delta_D^{out}(S)) = b(V \setminus S) - c(M) \le b(S_M) - c(M),$$

the last inequality being due to the fact that S_M is a superset of $V \setminus S$, as every node not contained in S is not reachable from s by definition of $s-t$ cuts.

Putting both observations together, this shows that a min $s-t$-cut in D gives rise to an optimal solution to the problem: Let S be such that $\delta_D^{out}(S)$ is a min capacity $s-t$ cut in D, and assume that $M := \delta_G^{out}(S)$ is *not* an optimal solution. Hence there is a better solution M'. Let k be its value. With the second observation, we get

$$k > b(V) - u(\delta_D^{out}(S))$$

Using the first observation, we know that there is $S' \subseteq V$ such that $\delta_D^{out}(S')$ an $s-t$-cut of value at most k, i.e.

$$\delta_D^{out}(S') \le b(V) - k.$$

The two inequalities imply

$$u(\delta_D^{out}(S)) > \delta_D^{out}(S'),$$

in contradiction to the fact that $\delta_D^{out}(S)$ is a min $s-t$-cut in D.