

Sample Solution to Problem Set 4

1. (10 points) The escape problem

Problem 26-1 of text.

Answer:

- (a) Vertex capacities can be handled as follows. For each vertex v , we add two vertices v_{in} and v_{out} with an edge from v_{in} to v_{out} with capacity equal to that of v in the original network. Any edge (u, v) in the original network is now replaced by (u_{out}, v_{in}) with capacity ∞ . It is easily seen that any flow through the original network satisfying vertex capacities corresponds to a flow with the same value in the new network, satisfying edge capacities. And vice versa. This reduction is computable in linear time.
- (b) The escape problem is identical to a network flow problem with unit capacity on each vertex. We add a source that has an edge to every starting point, and a sink that has an edge from every point on the boundary; each edge from the source and each edge to the sink has capacity 1. The escape problem is equivalent to asking whether there is a maximum flow of value at least m in this network. By part (a), we can reduce this problem to vertex capacities to a traditional flow problem with edge capacities in linear time. Computing the traditional flow problem, and transforming the flow back to the grid yields the vertex disjoint paths for the escape problem.

2. (7 + 8 = 15 points) Updating the maximum flow

Problem 26-4 of text.

Answer: Both the parts can be solved by using the concepts of augmenting paths and the residual network.

- (a) We first note that the increase in capacity of one edge can increase the max flow can increase by at most 1. This follows directly from the max-flow min-cut theorem and the fact that the min cut capacity can increase by at most 1. Therefore, either the max flow increases by 1 or remains the same.

Consider the residual network obtained from the original maximum flow f . Let G_f denote the residual network. Suppose we are given that the capacity of the edge (u, v) has increased by 1. We increase the capacity of the edge (u, v) in the residual network by 1. If (u, v) was not in the residual network, then we now add it and set its residual capacity to 1. Let G'_f denote the residual network thus obtained. We now run one iteration of the Ford-Fulkerson algorithm on G'_f . If we find an augmenting path from s to t in G'_f , then the flow increases by 1; otherwise, it remains the same.

The running time of this algorithm is the same as the running time of one iteration of the Ford-Fulkerson algorithm, which is $O(V + E)$.

- (b) We first note that the decrease in capacity of one edge can decrease the max flow can increase by at most 1. This follows directly from the max-flow min-cut theorem and the fact that the min cut capacity can decrease by at most 1. Therefore, either the max flow decreases by 1 or remains the same.

Consider the residual network obtained from the original maximum flow f . Let G_f denote the residual network. Suppose we are given that the capacity of the edge (u, v) has decreased by 1. We consider two cases.

The first case is when (u, v) is in G_f . This means that the residual capacity of (u, v) was at least 1 before the decrease. In this case, (u, v) can still carry its original flow even if its capacity is decreased by 1. Therefore, the max flow value remains the same. To maintain the residual network, we reduce the residual capacity of (u, v) by 1; if its residual capacity becomes 0, we remove it from the graph.

The second case is when (u, v) is not in G_f . This implies that $f(u, v)$ equals $c(u, v)$. The capacity of (u, v) has decreased by 1, however. Therefore, we have to reduce the flow on (u, v) by 1. In order to maintain the flow conservation law, we have to reduce the flow on each edge along some path from s to t containing (u, v) by 1. We find such a path by traversing the residual network. We find a path from u to s and from t to v in the residual network. This together with the edge (v, u) in the residual network gives a path p from t to s in the residual network. For each edge (x, y) in the path p , we make the following changes: we increase $f(x, y)$ by 1 and decrease $f(y, x)$ by 1. Note that one of the effects of these changes is that $f(u, v)$ decreases by 1, thus satisfying the capacity constraint of (u, v) . Furthermore, the net flow value has also decreased by 1. We now consider the new residual network G' . We run one more iteration of the Ford-Fulkerson algorithm. If we find an augmenting path from s to t in G' , then the flow increases by 1, thus resulting in the same max flow as before; otherwise, it remains the same, thus resulting in an overall flow that has decreased by 1.

The above algorithm performs two searches in G_f , one from t to v and the other from u to s . Each of these takes $O(V + E)$ time. And finally, we perform one iteration of the Ford-Fulkerson algorithm, which takes $O(V + E)$ time. The total running time is thus $O(V + E)$.

3. (10 + 5 = 15 points) Tight-knit communities in social networks

Social networks are all the craze these days. Assuming that friendship links are symmetric, one can represent a social network such as Facebook as an undirected unweighted graph G . Identifying clusters, communities, and other connectivity properties is a hot topic of research.

Suppose you want to identify a tight-knit community in a social network G . For a given set S of nodes in G , let $n(S)$ denote the number of edges with both endpoints in S . Define the *connectedness* of any set S of nodes in G to be $n(S)/|S|$. Larger the $n(S)$, the more tight-knit $n(S)$ is; dividing by $|S|$ normalizes across sets of different sizes.

- (a) Give a polynomial time algorithm that takes a rational number α and determines whether there exists a set S with connectedness at least α .

Answer: We went through this in class, so here we will just give a sketch. Consider the flow network N with the vertex set equal to the vertex set V of G plus a source s and a sink t ,

and with the following edges: directed edges (u, v) and (v, u) for every undirected edge (u, v) in G , and edges (s, u) and (u, t) for every vertex u in G . The capacities are as follows: 1 for each edge (u, v) , $d(u)$ for edge (s, u) , and 2α for edge (u, t) , where $d(u)$ is the degree of u in G .

Consider an s - t cut $(S \cup \{s\}, \bar{S} \cup \{t\})$, where \bar{S} is $V \setminus S$. The capacity of this cut equals $d(\bar{S}) + n(S, \bar{S}) + 2\alpha|S|$ where $d(X)$ denote the sum of degrees of vertices in X , and $n(X, Y)$ denotes the number of edges from X to Y . Thus, the capacity of this cut is

$$\begin{aligned} d(\bar{S}) + n(S, \bar{S}) + 2\alpha|S| &= d(V) - d(S) + n(S, \bar{S}) + 2\alpha|S| \\ &= d(V) - n(S, S) + 2\alpha|S| \end{aligned}$$

Thus the capacity of min-cut is at most $d(V)$ if and only if there exists an S for which $n(S, S) \geq 2\alpha|S|$, which is equivalent to the condition that there exists a set S with connectedness at least α .

- (b) Use the above algorithm to find a set S of nodes with maximum connectedness.

Answer: The number of possible values for the connectedness of a set is at most $n \times m$. By conducting a binary search over these values using the algorithm of part (a), we obtain the desired set S with maximum connectedness.

4. (5 + 2 + 3 = 10 points) Pathological example for the basic augmenting path algorithm

In the residual network G_f corresponding to a flow f , define an *augmenting walk* as a directed walk from source s to sink t that visits any arc at most once (it might visit nodes multiple times—in particular, an augmenting walk might visit nodes s and t multiple times). Define the *residual capacity* of an augmenting walk to be the residual capacity (i.e., capacity in G_f) of the minimum-capacity edge in the walk.

- (a) Consider the network shown in Figure 1(a) with the edges labeled a, b, c , and d ; note that one arc capacity is irrational. Show that this network contains an infinite sequence of augmenting walks whose residual capacities sum to the maximum flow value. (*Hint* : Each augmenting walk of the sequence contains exactly two arcs from node s to node t with finite residual capacities.)

Answer: Let us refer to the “reverse” edges corresponding to a, b, c , and d as $\bar{a}, \bar{b}, \bar{c}$, and \bar{d} , respectively. We note that the maxflow achievable is $1 + \sqrt{2}$.

- **Iteration 1:** Suppose we send a flow of 1 along path $\langle a, b, c \rangle$. This sets the residual capacities of these edges to be 0, ∞ , and $\sqrt{2} - 1$, respectively, and that of \bar{a}, \bar{b} , and \bar{c} to be 1, 1, and 1, respectively. The capacities of the two finite-capacity edges from s to t are $\sqrt{2} - 1$ and 1; the min-capacity is $\sqrt{2} - 1$. At this point, total flow sent is 1.
- **Iteration 2:** Next, we send a flow of $\sqrt{2} - 1$ along the path $\langle c, d, \bar{b} \rangle$. This sets the residual capacities of these edges to be 0, ∞ , and $2 - \sqrt{2}$, respectively, and that of \bar{c}, \bar{d} , and b to be $\sqrt{2}, \sqrt{2} - 1$, and ∞ , respectively. The total flow sent thus far is $1 + \sqrt{2} - 1 = \sqrt{2}$. So a flow of 1 still remains to be sent. The capacities of the two finite-capacity edges from s to t are $2 - \sqrt{2}$ and $\sqrt{2} - 1$; the min-capacity is $\sqrt{2} - 1$.

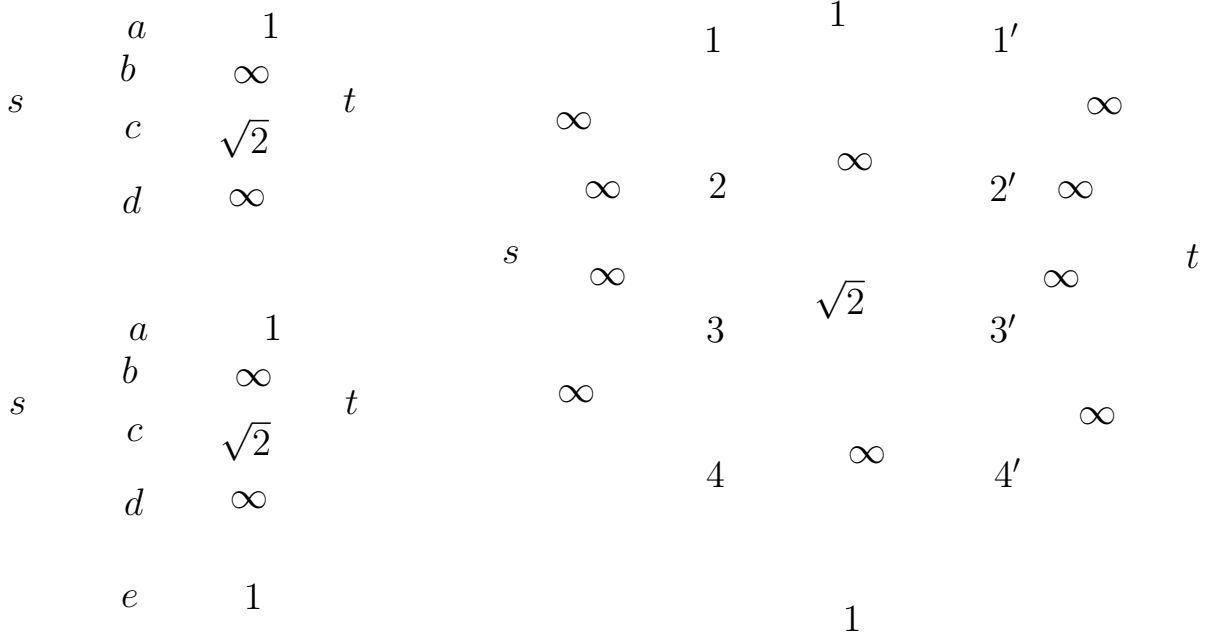


Figure 1: A pathological example for the basic augmenting path algorithm.

- **Iteration 3:** Next, we send a flow of $\sqrt{2} - 1$ along the path $\langle \bar{d}, \bar{a}, \bar{b} \rangle$. This sets the residual capacities of these edges to be 0, $2 - \sqrt{2}$, and $3 - 2\sqrt{2}$ and that of d , a , and b to be ∞ , $\sqrt{2} - 1$, and ∞ , respectively. The total flow sent thus far is $2\sqrt{2} - 1$. So a flow of $2 - \sqrt{2}$ still remains to be sent. The capacities of the two finite-capacity edges from s to t are $3 - 2\sqrt{2}$ and $\sqrt{2} - 1$; the min-capacity is $3 - 2\sqrt{2} < 2 - \sqrt{2}$.
- **Iteration 4:** Next, we send a flow of $3 - 2\sqrt{2}$ along the path $\langle \bar{b}, \bar{d}, \bar{a} \rangle$. This sets the residual capacities of these edges to be 0, ∞ , and $3\sqrt{2} - 4$, and that of b , \bar{d} , and \bar{a} , to be ∞ , $3 - 2\sqrt{2}$, and $5 - 3\sqrt{2}$, respectively. The total flow sent thus far is 2. So a flow of $\sqrt{2} - 1$ still remains to be sent. The capacities of the two finite-capacity edges from s to t are $3 - 2\sqrt{2}$ and $3\sqrt{2} - 4$; the min-capacity is $3 - 2\sqrt{2} < \sqrt{2} - 1$.
- **Iteration 5:** Next, we send a flow of $3 - 2\sqrt{2}$ along the path $\langle \bar{d}, \bar{b}, \bar{a} \rangle$. This sets the residual capacities of these edges to be 0, ∞ , and $5\sqrt{2} - 7$, and that of d , \bar{b} , and \bar{a} , to be ∞ , $3 - 2\sqrt{2}$, and $8 - 5\sqrt{2}$, respectively. The total flow sent thus far is $5 - 2\sqrt{2}$. So a flow of $3\sqrt{2} - 4$ still remains to be sent. The capacities of the two finite-capacity edges from s to t are $5\sqrt{2} - 7$ and $3 - 2\sqrt{2}$; the min-capacity is $5\sqrt{2} - 7 < 3\sqrt{2} - 4$.

Existence of an infinite sequence of augmenting walks: The above sequence suggests that we can continue the process to obtain an infinite sequence of augmenting walks. We now prove that we can indeed extend the above approach to derive an infinite sequence.

We will maintain the invariant that at the start of any iteration i , there will be exactly two non-zero capacity edges from s to t . This claim is true at the start of the augmentation. We now argue that the invariant holds after iteration i . At the start of iteration i , let α_i and β_i denote the smaller and larger capacities, respectively. In iteration i , we will send flow along

an augmenting walk that consists of the edge of capacity α_i , followed by an edge that is the reverse of a capacity 0 s - t edge, followed by the edge of capacity β_i . This will imply that at the start of iteration $i + 1$, we still have two edges of non-zero capacity from s to t ; they have capacities $\beta_i - \alpha_i$ and α_i . In particular, we have:

$$\alpha_{i+1} = \min\{\alpha_i, \beta_i - \alpha_i\}; \beta_{i+1} = \max\{\alpha_i, \beta_i - \alpha_i\} \quad (1)$$

This completes the proof of the invariant.

Equation 1 also tells us how the capacities α_i and β_i change. In particular, the iterative algorithm indicated by Equation 1 is precisely Euclid's algorithm for finding the greatest common divisor. Initially, α_1 and β_1 are 1 and $\sqrt{2}$, respectively. Since one number is rational and the other irrational, Euclid's algorithm will never terminate. It thus follows that neither α_i and β_i will ever be 0.

It remains to show that as i tends to infinity, the sum of the residual capacities of the augmenting walks tends to $1 + \sqrt{2}$. For this, we note that the capacity of the residual network at the start of iteration i is $\alpha_i + \beta_i$. For any $\varepsilon > 0$, we can argue that there exists an i such that $\alpha_i < \varepsilon$; suppose not, then we obtain that within iteration $1/\varepsilon$, β_i decreases to less than ε , which contradicts the inequality $\beta_i > \alpha_i$. Similarly, one can argue that the limit of β_i , as i tends to infinity, is 0. We thus have that the min-cut of the residual network is 0 in the limit. This establishes that the infinite sequence of augmenting walks yields the desired maximum flow.

- (b) Now consider the network shown in Figure 1(b). Show that this network contains an infinite sequence of augmenting walks whose residual capacities sum to a value different than the maximum flow value.

Answer: Part (b) follows directly from part (a), since the infinite sequence of augmenting walks in (a) have a total residual capacity of $1 + \sqrt{2}$ when the maximum flow is $2 + \sqrt{2}$.

- (c) Next consider the network shown in Figure 1(c); in addition to the edges shown, the network contains an infinite capacity edge connecting each node pair in the set $\{1, 2, 3, 4\}$ as well as each node pair in the set $\{1', 2', 3', 4'\}$. Show that each augmenting walk in the solution of part (b) corresponds to an augmenting path in Figure 1(c). Conclude the basic augmenting path algorithm, when applied to a maximum flow problem with irrational capacities, might perform an infinite sequence of augmentations and the terminal flow value might be different than the maximum flow value.

Answer: We can map augmenting walks in the network of Figure 1(b) to augmenting paths in the network of Figure 1(c) as follows. Edge a (\bar{a}) is mapped to edge $(1, 1')$ (edge $(1', 1)$); edge b (\bar{b}) is mapped to edge $(2', 2)$ (edge $(2, 2')$); edge c (\bar{c}) is mapped to edge $(3, 3')$ (edge $(3', 3)$); edge d (\bar{d}) is mapped to edge $(4', 4)$ (edge $(4, 4')$). An augmenting walk from s to t in the network of part (b) is mapped to an augmenting path by mapping the edges from $\{a, b, c, d, \bar{a}, \bar{b}, \bar{c}, \bar{d}\}$ as above and then connecting them using an edge from s to a node in $\{1, 2, 3, 4\}$, an appropriately chosen edge connecting a pair of nodes in $\{1, 2, 3, 4\}$, an appropriately chosen edge connecting a pair of nodes in $\{1', 2', 3', 4'\}$, and an edge connecting a node in $\{1', 2', 3', 4'\}$ to t .

The desired conclusion follows from the claim of part (b) and the above mapping.