Ling Li, ling@cs.caltech.edu
February 16, 2001

13.1 Maximum augmenting path

(a) Construct $G_f = (V_f, E_f, c_f)$ from $G = (V, E, c)$ and f. Initially $V_f = V$, $E_f = \emptyset$. For each edge (u, v) with capacity $c(u, v)$, calculate $r(u, v) = c(u, v) - f(u, v)$ and $r(v, u) = -r(u, v)$. If $r(u, v) \neq 0$, then add edge (u, v) into E_f and set $c_f(u, v) = r(u, v)$. Otherwise drop this edge. Thus we get G_f in $O(m)$.

(b) Use (a) to construct the residual graph G_f in $O(m)$ and apply the algorithm below to (G_f, s):

Modified Dijkstra (G, u)*: Begin with all the vertices of* G *in the priority queue* P *with key function* k*,* u *with key 0, and all other vertices with key* ∞*. Repeat until* P *is empty:*

1. $a :=$ *DeleteMin*P.
2. *For each* $(a, b) \in E$*: DecreaseKey*$(b, \max\{k(a), -c(a, b)\})$*. Note that* $c(a, b) = 0$ *if edge* (a, b) *is not in* G_f*.*

Here the only difference is that $\max\{k(a), -c(a, b)\}$ is used instead of $k(a) + d(a, b)$. This makes the algorithm find the minimum negative bottleneck capacity instead of minimum length. Thus in $O((n + m)\log n)$ time the algorithm will find a path with minimum negative bottleneck capacity from s to t, which is the maximum bottleneck capacity path.

The proof is very similar to that of the original Dijkstra algorithm.

Ling Li, ling@cs.caltech.edu

13.2 s, t-connectivity problem

Assign a unit capacity to each edge in G and find a max flow f of G. If there exist k edge-disjoint paths from s to t, then we can construct a flow by allowing one unit of flow along each of those k paths. So $|f| \geq k$. (Thus if $|f| < k$, there do not exist k edge-disjoint paths from s to t.)

If $|f| \geq k$, by Lemma 17.4 in Kozen's book, we can decomposite f into several path flows and a flow f' with $|f'| = 0$. Since any edge in G has a unit capacity, a path flow in G can only have value 1 and two path flows extracted from f can not be edge-joint. Thus f consists of exactly $|f| \geq k$ edge-disjoint paths from s to t. Thus we found k edge-disjoint paths.

The time complexity is $O(m^2 n)$, since the Edmonds-Karp algorithm is $O(m^2 n)$ and for each of at most m path flows we need $O(m)$ (to construct the new graph and find *any* path from s to t). Here m is the number of edges and n is the number of vertices in G.

For undirected graph, we can construct a directed graph G' from G (see homework 13.3) and apply the above algorithm to G'.

Ling Li, ling@cs.caltech.edu

13.3 Min cut for undirected graph

We can construct a directed graph G' from the given undirected graph $G = (V, E)$ and apply the maximum flow to $G' = (V', E')$. Then we will get the min cut from the residual graph associated with the max flow in G'. This is the main idea.

$G \to G'$: $V' = V$. For each edge $\{u, v\}$ with weight w in E, create two directed edges (u, v) and (v, u) with capacity w for E'. Then in $O(m + n)$ time, we create G', where m is the number of edges and n is the number of vertices.

Run Edmonds-Karp algorithm on G' to get the max flow f of G'. The time complexity is $O(m^2 n)$.

From homework 13.1, we can construct the residual graph G'_f in $O(m)$ time.

Let A consist of all vertices reachable from s by paths in G'_f. Let $B = V - A$. Thus by the Max Flow-Min Cut Theorem, this is the min cut in G'. Thus it is also a min cut for G. The complexity for this step is $O(m + n)$.

Proof of the correctness: It is obvious since the two graphs, G and G', share the same topology structure and the weights are also the same for corresponding edges.

Ling Li, ling@cs.caltech.edu

13.4 Girls' diplomas

For the sake of convenience, let's change the procedure of passing diplomas a little: all the girls will *first* form a circle and then those who get their diplomas drop out. This is exactly the same as the procedure in the problem. However, now it is easier to number the girls.

In the initial circle, number all the girls counterclockwise with numbers 0 to $n-1$. The girl at position i is denoted by g_i. Note that all calculations on the subscript (the index of the girl) are $\bmod n$. Thus g_i will pass her diploma to g_{i+1} if neither g_i nor g_{i+1} gets her own diploma. Let d_i be the initial position of g_i's diploma. That is, at first g_{d_i} has the diploma of g_i.

Let random variable $X_{i,k}$ equal 1 if g_i ever passes g_j's diploma, and be 0 otherwise, where $j = i+k$ (remember the $\bmod n$), $0 \le i < n$ and $1 \le k < n$. Thus the number of handoffs made by g_i is $\sum_{k=1}^{n-1} X_{i,k}$ and the total number of individual handoffs is $\sum_{i=0}^{n-1}\sum_{k=1}^{n-1} X_{i,k}$.

For any $0 \le i < n$, $1 \le k < n$ and $j = i+k$, there are $n(n-1)$ combinations of d_i and d_j, since $d_i \ne d_j$. We want to find out which combinations would have $X_{i,k} = 1$. Here are 3 cases:

- $d_i = i$. Then g_i drops out immediately and obviously $X_{i,k} = 0$.
- $d_i \in \{j-n, j-n+1, \ldots, i-1\}$ (again, take $\bmod n$). Visually, position d_i is left to g_i and right to or at g_j. To have the chance for g_i to pass the diploma of g_j before she gets his own diploma, d_j must be left to or at g_i and right to d_i. That is, $X_{i,k} = 1$ iff $d_j \in \{d_i+1, \ldots, i-1, i\}$.
- $d_i \in \{i+1, i+2, \ldots, j-1\}$. For similar reason, $X_{i,k} = 1$ still requires $d_j \in \{d_i+1, \ldots, i-1, i\}$. However, if $d_j \in \{d_i+1, \ldots, j\}$, before meeting g_i, the diploma of g_j will first meet g_j and then drop out together with g_j. So in this case, $d_j \in \{j+1, \ldots, i-1, i\}$.

Totally there are

$$\sum_{d_i=j-n}^{i-1} (i-d_i) + \sum_{d_i=i+1}^{j-1} (i+n-j) = \sum_{t=1}^{n-k} t + (k-1)(n-k) = \frac{(n-k)(n+k-1)}{2}$$

combinations of d_i and d_j that make $X_{i,j} = 1$. Thus we have

$$E(X_{i,k}) = P(X_{i,k} = 1) = \frac{(n-k)(n+k-1)}{2n(n-1)} = \frac{1}{2} - \frac{k^2-k}{2n(n-1)}.$$

So

$$\begin{aligned}
E\left(\sum_{i=0}^{n-1}\sum_{k=1}^{n-1} X_{i,k}\right) &= \sum_{i=0}^{n-1}\sum_{k=1}^{n-1} E(X_{i,k}) \\
&= \frac{n(n-1)}{2} - \frac{1}{2(n-1)}\left[\sum_{k=1}^{n-1}(k^2-k)\right] \\
&= \frac{n(n-1)}{2} - \frac{1}{2(n-1)}\left[\frac{(n-1)n(2n-1)}{6} - \frac{(n-1)n}{2}\right] \\
&= \frac{n(2n-1)}{6}.
\end{aligned}$$

That is, the expectation of the total number of individual handoffs is $\frac{n(2n-1)}{6}$.