

The algorithm is as follows:

1. Form the residual graph.
2. Find a shortest $s - t$ path in the residual graph, and augment along that.
3. If a path was found, then repeat.

As will be seen, this is similar to the Ford-Fulkerson algorithm, except that a shortest path is used, whereas the Ford-Fulkerson algorithm allowed any path whatsoever.

Theorem 1 *The KE algorithm finds the maximum flow in time $O(\min(F, mn)(m+n))$, where F is the value of the maximum flow, and m, n are the number of edges and vertices in the graph.*

Proof: We show in Lemma 1 that the length of the shortest path never decreases, as the algorithm executes. We also show in Lemma 2 that the length can stay the same for at most m steps. Given that the length can at most change by n , it follows that there are at most mn flow augmentations. Since each augmentation requires a breadth first search, the total time is at most $O((m+n)mn)$. But, the time is also $O(F(m+n))$, because this is a special case of the Ford-Fulkerson algorithm. ■

Note that since the time is $O((m+n)mn)$, this is a polynomial time algorithm rather than pseudo-polynomial time.

Lemma 1 *The length of the shortest path in the $i+1$ th iteration is at least the length in the i th iteration.*

Proof: Consider the i th iteration. In this we executed a breadth first search to find the shortest path. Because of BFS, we may think of the nodes as situated in levels, say with s in level 0 and t in level k . Further, all the edges of the residual graph are constrained to be *forward* edges which go from some

level j to level $j + 1$, and *other* edges which go from level j to level j or lower. The key point is that there are no edges going from level j to any level higher than $j + 1$.

Colour the forward edges black, and other edges red.

Now, as a result of the augmentation, we will (a) remove some of the forward edges, (b) insert back edges going from level $j + 1$ to j for every edge in the augmenting path. The inserted edges will be coloured red. Let G_i denote the residual graph obtained at this point, along with its colours.

Now suppose P denotes the shortest path found in the $i + 1$ th iteration. P starts in level 0 of G_i and goes to level k . It must make k forward displacements. Forward displacements can be made only using black edges, and these only go forward by one level. Thus P must use at least k black edges. ■

Lemma 2 *There can be at most m iterations in which the path length remains the same.*

Proof: Consider the graph G_i constructed above in some iteration i . If in any subsequent iteration, a red edge is used, we know that the path length must increase, because as discussed, k black edges are needed to make the k forward displacements. But then, in each such iteration, at least one black edge is removed, the one whose capacity is minimum along the path. But there are at most m black edges to begin with, either each edge in the original graph or its reverse edge. Thus after m such iterations the black edges will all disappear and the augmenting path will have to use some red edge. Thus at that time the path length would have to increase. ■

1 Concluding Remarks

There exist substantially faster algorithms for maxflow. But it is worth understanding how BFS improves the Ford Fulkerson algorithm.

I don't know if there exists a graph needing $\Omega(mn)$ augmentations. I would like to know.