An Alternative Proof of the Running Time of the Edmonds-Karp Algorithm

Christian Wulff-Nilsen

1 Introduction

These notes for the course Advanced Algorithms and Data Structures provide an alternative proof that the Edmonds-Karp algorithm runs in time $O(VE^2)$. At the exam, you are free to choose between this proof and that in CLRS (Lemma 26.7, Theorem 26.8, and their proofs). We use the same notation as in CLRS; for instance $\delta_f(u,v)$ is the shortest path distance in G_f from u to v where each edge has weight 1.

2 Bounding running time

Let $G=(V,E)$ with source s, sink t, and capacity function c be the flow network that we run the Edmonds-Karp algorithm on. For a given residual network G_f, a *forward edge* of G_f is an edge $(u,v) \in E_f$ such that $\delta_f(s,v) = \delta_f(s,u)+1$.

We need the following lemma (see Figure 1 for an illustration):

Lemma 1. *Let G_{f_0} be a residual network obtained at some point in the algorithm and let $f_1, \ldots, f_k$ be the sequence of flows obtained during the next $k \geq 0$ iterations. Let $d = \delta_{f_0}(s,t)$ and assume that $\delta_{f_i}(s,t) = d$ for $i = 1, \ldots, k-1$. Then in each of the k iterations, flow is only pushed along edges that are forward edges in G_{f_0}. Furthermore, $\delta_{f_k}(s,t) \geq d$.*

Proof. The proof is by induction on $k \geq 0$. The proof is clear for $k=0$ since then no iterations have been executed and $G_{f_k} = G_{f_0}$.

Now assume that $k > 0$ and that the claim holds for $k-1$. By the induction hypothesis, flow has only been pushed along edges that are forward edges in G_{f_0} during the first $k-1$ iterations.

To obtain G_{f_k}, the algorithm augments flow along a shortest path p from s to t in $G_{f_{k-1}}$. For any edge $(u,v) \in p$, we must have $\delta_{f_0}(s,v) \leq \delta_{f_0}(s,u)+1$; this follows since flow has only been pushed along forward edges of G_{f_0} so (u,v) is either an edge of G_{f_0} or the reversal of a forward edge of G_{f_0}.

Since p has length d and since $\delta_{f_0}(s,t) = d$, the only possibility is that $\delta_{f_0}(s,v) = \delta_{f_0}(s,u)+1$ for every edge $(u,v) \in p$. Hence, p contains only edges that are forward edges in G_{f_0}. This shows the first part of the induction step.

The same argument as above shows that for any edge (u,v) of a shortest path p from s to t in G_{f_k}, $\delta_{f_0}(s,v) \leq \delta_{f_0}(s,u)+1$. Combining this with the fact that $\delta_{f_0}(s,t) = d$, it follows that p must have at least d edges and therefore $\delta_{f_k}(s,t) \geq d$. This completes the induction step and thus the proof. □

We can now obtain the desired time bound.

Theorem 1. *The Edmonds-Karp algorithm runs in $O(VE^2)$ time.*

Proof. Each iteration of the algorithm takes $O(V+E) = O(E)$ time using breadth-first search in the residual network. It thus suffices to show that the number of iterations is $O(VE)$.

Consider a maximum-length sequence of consecutive flows $f_0, \ldots, f_k$ obtained by the algorithm such that $\delta_{f_i}(s,t) = d$ for $i = 1, \ldots, k-1$, where $d = \delta_{f_0}(s,t)$.

By Lemma 1, we must have $k \leq |E|$ since G_{f_0} has at most $|E|$ forward edges and each of the k iterations removes at least one such edge (if an iteration did not remove at least one such edge, it would not have sent the maximum amount of flow along an augmenting path); once a forward edge disappears from the residual network, it cannot reappear during the k iterations since otherwise, flow would have been pushed along the reverse of a forward edge, contradicting Lemma 1. Lemma 1 also gives $\delta_{f_k}(s,t) \geq d$. By maximality of the sequence $f_0, \ldots, f_k$, we cannot have $\delta_{f_k}(s,t) = d$ so $\delta_{f_k}(s,t) \geq d+1$.

We have shown that the distance from s to t in the residual network can never decrease and that every $|E|$ iterations, this distance must have increased by at least 1. The distance can never exceed $|V|-1$ unless it is infinite in which case the algorithm terminates. We conclude that the number of iterations is $O(VE)$, as desired. □

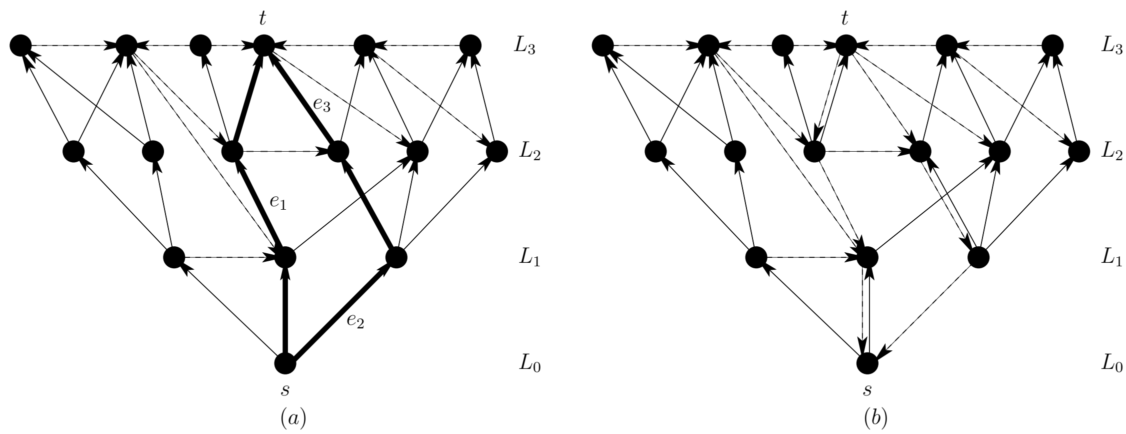

Figure 1: Illustration of Lemma 1 with $k = 2$. (a): G_{f_0} where solid edges are forward edges (residual capacities not shown). The fat paths are the next two augmenting paths chosen by the algorithm and they consist of forward edges only. Edges e_1, e_2, and e_3 are filled to their residual capacity by these flow augmentations. (b) G_{f_2} where solid edges are the remaining forward edges of G_{f_0}. Lemma 1 states that $\delta_{f_2}(s,t) \geq d$ with $d = \delta_{f_0}(s,t) = \delta_{f_1}(s,t)$ (in this case, $\delta_{f_2}(s,t) = 4$ and $d = 3$). In both figures, L_i is the ith layer of the breadth-first search tree from s in G_{f_0}, i.e., $L_i = \{v \in V | \delta_{f_0}(s,v) = i\}$, for $i = 0, 1, 2, 3$.