1 Definitions

1.1 Flow network

1. A flow network $G = (V, E)$ is a graph where each edge $(u, v) \in E$ has non-negative capacity $c(u, v) \geq 0$. It is further required that
   (a) If $(u, v) \in E$ then we have $(v, u) \notin E$ (so there are no reverse edges)
   (b) If $(u, v) \notin E$ then we set $c(u, v) = 0$
   (c) No self loops
2. We have two special vertices in a flow network
   (a) Source s
   (b) Sink t
3. We assume that in a flow network there is a path from the source to any vertex and then to the sink, $\forall v \in V, s \rightsquigarrow v \rightsquigarrow t$. So we have a connected graph
4. Since each vertex has at least one entering edege(the above assumption) except the source, s, then we will have that $|E| \geq |V| - 1$.

1.2 Capacity constraint

For all $u, v \in V$ we require $0 \leq f(u, v) \leq c(u, v)$

That is we have non-negative flow that is bounded by the capacity of an edge. For edges not in the flow network this still hold since point 1.b states we set flow 0 in this case.

1.3 Flow conservation

For all $u \in V \setminus \{s, t\}$ we require that

$$\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$$

And as stated by 1.b, when $(u, v) \notin E$ we set $f(u, v) = 0$

1.4 Value of a flow

We defined the value of flow f as

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

Often the second sum is 0 and there is no flow going into s.

1.5 Antiparallel edges

Two edges $(v1, v2)$ and $(v2, v1)$ are called antiparallel and are not allowed as stated earlier.We can remove antiparallel edges by splitting edge $(v1, v2)$ into $(v1, v')$and $(v', v2)$. We simply set $c(v1, v') = c(v', v2) = c(v1, v2)$. This procedures ensure there is no reverse of an edge.

1.6 Multiple source and sink

1. Add a supersource s
2. Make edges from (s, s_i) for each of the source
3. Set their capacity $c(s, s_i) = \infty$ such that the original flow from $s_i$can flow.

We can do the same for a supersink t that consumes whatever is required for each t_i

1.7 Residual networks

Edges of G_f represent how much we can change the flow along each edge in G. It is made in the following way

1. We place an edge into $G_f$if it has positive "residual capacity" $c_f(u, v) = c(u, v) - f(u, v) > 0$
2. Flows of edges with residual capacity $c_f(u, v) = 0$ are not in G_f

In order to increase flow on a particular edge it might be needed to decrease flow along another edge (u, v). We decrease flow by adding "reverse edges" (v, u)in G_f

1. Therefore we also add edge with residual capacity $c_f(v, u) = f(u, v)$
   (a) Thus we see that (v, u) can at most cancel out the flow of (u, v)
   (b) Sending flow back from edge (sending some flow along $(u, v)$and then send some of it back along (v, u) is equivalent to *decreasing flow.*

1.8 Residual capacity

Due to the need to decrease flow, we define formally the residual capcity of G with flow f as

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E \\ f(v, u) & \text{if } (v, u) \in E \\ 0 & \text{otherwise} \end{cases}$$

Note each case is exclusive and that one case will apply to all pairs of vertices.

The first case holds for regular edges in E, and then the second case will not hold since we require that $(u, v) \in E \implies (v, u) \notin E$.

The second case holds since an edge (u, v) in the residual network is the reverse of an edge in the original network (v, u). If $(u, v) \notin E$ then it might be that (v, u) is in E. This is just the reverse of case 1.

The third case is just for edges not in G, these will neither have reverse edges in $G_f$so case two does neither hold, and we do not draw these edges in G_f

1.9 Residual network formal

Using the formal definition of residual capacity we define the residual network $G_f = (V, E_f)$ where we define

$$E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$$

which is all the strictly positive edges.

Now we see that each edge in the residual network only omits flow greater than 0. Since each edge is either an edge in G or their reversals, then we have

$$|E_f| \leq 2|E|$$

Note G_f is almost a flow network with capcity function c_f. It does not follow the definition since we low reversals of an edge.

1.10 Flow Augmentation

We can thing of G_f as a map of how much flow can be added to each edge in G. If we have a flow f' in G_f we define the augmentation of a flow as

$$(f \uparrow f')(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u) & if (u, v) \in E \\ 0 & otherwises \end{cases}$$

note how the subtraction of the reversal of an edge in this definition shows how it is equivalent to just having less flow along $(u, v) \in E_f$.

1.11 Augmenting Path

A simple path from s to t in the residual network G_f. Sine the path is made up of edges in $G_f$which denote how much more we can increase flow in G, we can increase the flow on an edge (u, v) on the path p by up to $c_f(u, v)$ without we violate the capacity constraint of (u, v) in the original network. That is, we for edge (u, v) on path p update the flow on edge (u, v) or (v, u) (subtract flow) by $c_f(u, v)$ then we do not violate the capacity constraint $c(u, v)$ in the original network.

1.12 Residual capacity formal

The maximum amount of flow we can increase on each edge along an augmenting path p

$$c_f(p) = \min(c_f(u, v) : (u, v)\, is\, on\, p)$$

1.13 Cut of a flow

A cut (S,T) of a flow network $G=(V,E)$ is a partitioning of vertices into the sets $s \in S$ and $t \in T = V - S$.

1.14 Net flow

The flow across the cut is defined as

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

1.15 Capacity of a cut

Capacity of a cut (S,T) is defined as

$$cS,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

It is intentional that we only count capacity going from S to T. This is unlike the net-flow definition.

1.16 Minimum cut

A cut of a flow network, is a cut (S,T) with minimum capacity.

2 Ford fulkerson method

1. Start with $f(u,v) = 0$ for all $u, v \in V$
2. while there exists an augmenting path p in residual network G_f
   (a) Augment flow f along p
3. return f

3 Proofs

3.1 Leamma 26.1

Let f be a flow in G and f' be a flow in G_f. The function $f \uparrow f'$ is a flow in G with value $|f \uparrow f'| = |f| + |f'|$.

Capacity constraint proof

Here we must prove $0 \leq f' \leq c_f$.

First we prove that the capacity is non-negative. If we have an edge $(u, v) \in E$ in the original flow network, and we use the formal definition of residual capacity, the $c_f(v, u) = f(u, v)$ using the second case((v, u) cannot be in E but (u, v) is). Since the flow is never larger than the capacity we have $f'(v, u) \leq c_f(v, u) = f(u, v)$. Inserting this into flow augmentation gives us

$$\begin{aligned}(f \uparrow f')(u, v) &= f(u, v) + f'(u, v) - f'(v, u) \\ &\geq f(u, v) + f'(u, v) - f(u, v)(\text{using the aboe note}) \\ &= f'(u, v) \\ &\geq 0\end{aligned}$$

where the last inequality holds since the lemma assume we have a flow f' in G_f.

Next we show that the **flow is bounded by the capacity.**

$$\begin{aligned}&(f \uparrow f')(u, v) = f(u, v) + f'(u, v) - f'(v, u) \\ &\leq f(u, v) + f'(u, v)(\text{f' is a flow and hence nonnegative}) \\ &\leq f(u, v) + c_f(u, v)(\text{f' is a flow so capacity constraint holds}) \\ &= f(u, v) + c(u, v) - f(u, v) \text{ by definition} \\ &\quad c(u, v)\end{aligned}$$

so the resuling flow is also bound be the capacity. This holds for all edges, since we picked any edge. ***The capacity constraint therefore holds-***

Next we show **flow conservation holds** for all $u \in V - \{s, t\}$

$$\begin{aligned}\sum_{v \in V} (f \uparrow f')(u, v) &= \sum_{v \in V} f(u, v) + f'(u, v) - f'(v, u) \\ &= \sum_{v \in V} f(u, v) + \sum_{v \in V} f'(u, v) - \sum_{v \in V} f'(v, u) \\ &= \sum_{v \in V} f(v, u) + \sum_{v \in V} f'(v, u) - \sum_{v \in V} f'(u, v) \\ &= \sum_{v \in V} f(v, u) + f'(v, u) - f'(u, v) \\ &= \sum_{v \in V} (f \uparrow f')(u, v)\end{aligned}$$

where the third line holds since we now each flow has conservation so the flow into v equals the flow out of v.

Next we must compute the **value of the flow**. Since antiparallel edges are disallowed then we know that either (u, v) or (v, u) can be amongst the edges of G but never both. This is not the case for G_f, however.

We define, for a fixed vertex u the set $V_1(u) = \{v : (u, v) \in E\}$ to be the set of vertices with an edge from u. Likewise we define $V_2(u) = \{v : (v, u) \in E\}$ to be the set of vertices with an edge going into u.Using the knowledge that only one of (u, v) or (v, u) can be in E, then we know that $V_1(u) \cup V_2(u) \subseteq V$ and that $V_1(u) \cap V_2(u) = \emptyset$.

By the definition of flow augmentation $(f(u,v)+f'(u,v)-f'(v,u)\, if\ (u,v) \in E\, 0\, otherwise)$ we know that only vertices from $V_1(u)$ can have positive flow augmentation by $(f \uparrow f')(u, v)$, since these are the only vertices with an edge from u and therefore all other edges will have (u, v') will have augmented flow 0. Likewise only vertices in $V_2(u)$ can have positive values for $f \uparrow f')(v, u)$ since these are the only vertices that have an edge into u (making (v, u) an edge in E) and all others will have 0 flow augmentation. Using this we get

$$\sum_{v \in V} (f \uparrow f')(s, v) - \sum_{v \in V} (f \uparrow f')(v, s)$$
$$= \sum_{v \in V_1(u)} (f \uparrow f')(u, v) - \sum_{v \in V_2(u)} (f \uparrow f')(v, u)$$

Now we can use the fact that for all $v \in V_1(u)$ then $(u, v) \in E$. Likewise for all $v \in V_2(u)$ then $(v, u) \in E$, and we can therefore use the definition of flow augmentation.

$$\begin{aligned}
&\sum_{v \in V} (f \uparrow f')(u, v) - \sum_{v \in V} (f \uparrow f')(v, u) \\
&= \sum_{v \in V_1(u)} (f(u,v) + f'(u,v) - f'(v,u)) - \sum_{v \in V_2(u)} (f(v,u) + f'(v,u) - f'(u,v)) \\
&= \sum_{v \in V_1(u)} f(u,v) + \sum_{v \in V_1(u)} f'(u,v) - \sum_{v \in V_1(u)} f'(v,u) \\
&- \sum_{v \in V_2(u)} f(v,u) - \sum_{v \in V_2(u)} f'(v,u) + \sum_{v \in V_2(u)} f'(u,v) \\
&= \sum_{v \in V_1} f(u,v) - \sum_{v \in V_2} f(v,u) - \sum_{v \in V_1 \cup V_2} f'(v,u) + \sum_{v \in V_1 \cup V_2} f'(u,v)
\end{aligned}$$

where we can pull the f' sums together since $V_1(u)$ and $V_2(u)$ are disjoint as argued, so we do end up summing over each set. The way we partitioned $V_1(u)$and $V_2(u)$ ensured that we only look at edges going out of u and into u respectively. This means, that if we instead just sum over the entire V, then we loop over edges that are not in G so the flow is zero. The same goes for $V_1(u) \cup V_2(u)$. These are the only vertices with an edge that has endpoint in s in the residual network. So all other terms in the sum is 0.

$$\sum_{v \in V} (f \uparrow f')(u, v) - \sum_{v \in V} (f \uparrow f')(v, u)$$
$$= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) - \sum_{v \in V} f'(v, s) + \sum_{v \in V} f'(s, v)$$

Using the definition of the **value of a flow** we can now show that value of the flow to be

$$\begin{aligned}|f \uparrow f'| &= \sum_{v \in V} (f \uparrow f')(u,v) - \sum_{v \in V} (f \uparrow f')(v,u) \\ &= \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) - \sum_{v \in V} f'(v,s) + \sum_{v \in V} f'(s,v) \\ &= |f| + |f'|\end{aligned}$$

For **flow conservation** we first now that the definition can be rewritten

$$\sum_{v \in V \setminus \{s,t\}} f(u,v) = \sum_{v \in V \setminus \{s,t\}} f(v,u) \iff \sum_{v \in V \setminus \{s,t\}} f(u,v) - \sum_{v \in V \setminus \{s,t\}} f(v,u) = 0$$

then using the definition and using the difference we used in calculating the value of the flow

$$\begin{aligned}&\sum_{v \in V} (f \uparrow f')(u,v) - \sum_{v \in V} (f \uparrow f')(v,u) \\ &\qquad = \left(\sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) \right) - \sum_{v \in V} f'(v,s) + \sum_{v \in V} f'(s,v)\end{aligned}$$

then we know the part within parenthesis is 0 for $v \in V \setminus \{s,t\}$, since f is a flow and since flow conservation holds for f. For the part outside the parenthesis then f' is also a flow and flow conservation holds.

$$\begin{aligned}&\sum_{v \in V \setminus \{s,t\},} (f \uparrow f')(u,v) - \sum_{v \in V \setminus \{s,t\},} (f \uparrow f')(v,u) \\ &\qquad = \sum_{v \in V \setminus \{s,t\},} f(s,v) - \sum_{v \in V \setminus \{s,t\},} f(v,s) - \sum_{v \in V \setminus \{s,t\},} f'(v,s) + \sum_{v \in V \setminus \{s,t\},} f'(s,v) \\ &\qquad = 0\end{aligned}$$

thereby we have shown flow conservation.

3.2 Lemma 26.2

The following is a flow in G_f with value $|f_p| = f_f(p) > 0$

$$f_p(u,v) = \begin{cases} c_f(p) & if\,(u,v)\,is\,on\,p \\ 0 & otherwise \end{cases}$$

remember that $c_f(p) = \min(c_f(u,v) : (u,v)\,is\,on\,p)$.

3.3 Corollary 26.3

If we have a flow network G and an augmenting path p in the residual network G_f and define f_p as in lemma 26.2, then augmenting $f \uparrow f_p$is a flow in G with value

$$|f \uparrow f_p| = |f| + |f_p|$$

Proof

By lemma 26.2 we know that $f_p$is a flow in G_f. Lemma 26.1 tells us if we have a flow in G_f and a flow in G, then augmenting the flow $f \uparrow f'$ has the value $|f \uparrow f'| = |f| + |f'|$. We can just use $f_p = f'$ as the flow in G_f and we are done.

3.4 Lemma 26.4

The net flow across a cut (S, T) in a flow network G is $f(S, T) = |f|$.

Proof

Using the definition of the flow conservation we can write

$$0 = \sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u)$$

and using this in part of the definition of value of a flow gives

$$\begin{aligned} |f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{u \in S \setminus \{s\}} 0 \\ \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u) \end{aligned}$$

we can expand the outer sum

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(u, v) - \sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(v, u)$$

and changing the order of the sums(commutativity) gives us

$$\begin{aligned} |f| &= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{v \in V} \sum_{u \in S \setminus \{s\}} f(u, v) - \sum_{v \in V} \sum_{u \in S \setminus \{s\}} f(v, u) \\ &= \sum_{v \in V} \left(f(s, v) + \sum_{u \in S \setminus \{s\}} f(u, v) \right) - \sum_{v \in V} \left(f(v, s) + \sum_{u \in S \setminus \{s\}} f(v, u) \right) \\ &= \sum_{v \in V} \sum_{u \in S} f(u, v) - \sum_{v \in V} \sum_{u \in S} f(v, u) \end{aligned}$$

where the last equality holds because the sum of $f(s,v)$ and $\sum_{u \in S \setminus \{s\}} f(u,v)$ is essentially just saying the flow from s to vertex v plus all other flows along a single edge to v. This is the same as having the inner sum include s. The same goes for the flow going into s - the flow from v to s plus the flow from v into all other vertices than s is just the flow from v into all vertices. By all other vetices in mean all vertices in S.

We have that $V = S \cup T$ and that $S \cap T = \emptyset$ due to the way we defined the cut. Therefore, we can split the sum over V into two sums.

$$
\begin{aligned}
|f| &= \sum_{v \in S} \sum_{u \in S} f(u,v) + \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u) - \sum_{v \in T} \sum_{u \in S} f(v,u) \\
&= \left(\sum_{v \in S} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u) \right) + \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u)
\end{aligned}
$$

within the parenthesis the doubly nested sum will loop over all combinations of pairs of vertices. This means that it will look at all combinations of flows between pairs of vertices, so it does not matter if you take $f(v,u)$ or $f(u,v)$ inside the sum, since each v and u will at some point have had all values vertices in V. Hence this sum cancels out and we get

$$
\begin{aligned}
|f| &= \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) \\
&= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\
&= f(S,T)
\end{aligned}
$$

3.5 Corollary 26.5

The value of any flow f in G is upper bounded by the capacity of any cut in G. That is $|f| \leq c(S,T)$

Proof

$$
\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\
&\leq \sum_{u \in S} \sum_{v \in T} f(u,v) \\
&\leq \sum_{u \in S} \sum_{v \in T} c(u,v) \\
&= c(S,T)
\end{aligned}
$$

where the first inequality holds by nonnegativity of flows and the second holds by definition.

3.6 Theorem 26.6

For a flow network then the following are equivalent

1. f is a maximum flow in G
2. The residual network $G_f$contains no augmenting paths
3. $|f| = c(S, T)$ for some cut (S, T) of G

Proving 1=>2

We suppose for contradiction we have a max flow f while there still is an augmenting path p. Recall that f_p is a flow in G_f consisting of the residual capacities of the edges along p. Therefore, we can use corollary 26.3 to create and even better flow with value $|f \uparrow f_p| = |f| + |f_p| > |f|$. This is a contradiction to f being a max flow, so it must be a max flow.

Proving 2=>3

Suppose there is no augmenting path in G_f, that is, no simple path from $s$to t in G_f. Next we define a cut where we select $S = \{v \in V : there\, exists\, a\, path\, from\, s\, to\, v\, in\, G_f\}$ and $T = V - S$. This is a cut, since S contains s(the path consists of no edges) and $t \notin S$ since there is no path from s to t. Likewise $t \in T$ since it is in V and not in S.

We now pick a pair of vertices $u \in S$ and $v \in T$. If the edge $(u, v) \in E$ then since u, v are on different sides of the cut, there is no path from u to v and thereby no edge (u, v) in the residual network, G_f. Additionally we defined $c_f(u, v) = c(u, v) - f(u, v)$, and since there is no edge this is 0 causing $f(u, v) = c(u, v)$.

On the other hand if $(v, u) \in E$, then we must have $f(v, u) = 0$. Using the second case of residual capacity, we would otherwise have that $c_f(u, v) = f(v, u)$, which would place an edge from $(u, v) \in E_f$. This cannot be possible since u, v are on different sides of the cut, and there is therefore no path or edge from u to v. There is an edge case in which (u, v) and (v, u) are not in E in which case we also have $f(u, v) = f(v, u) = 0$. Writing the flow of the cut

$$
\begin{aligned}
f(S, T) = &\sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} (f(v, u) \\
= &\sum_{u \in S} \sum_{v \in T} f(u, v) - 0 \\
= &c(S, T)
\end{aligned}
$$

where we used $f(u, v) = c(u, v)$ as argued and that $f(v, u) = 0$ as argued. Hence we have shown no augmenting path means the net-flow is the capacity of a cut.

Proving 3=>1

Corollary 26.5 states that $|f| \leq c(S,T)$. Therefore, if we have that $|f| = c(S,t)$ then we have a maximum flow f.

3.7 Lemma 26.7

For all vertices $v \in V \setminus \{s,t\}$ the shortest path distance $\delta_f(s,v)$ in the residual network G_f increase monotonically with each flow augmentation.

Proof

The proof is based on contradiction that the shortests bath distance decreases and the we arrive at a contradiction.

Let f be the flow just before the first augmentation that decreased some shortest path distance in G_f, and let f' be the flow after the augmentation. Let v be the vertex with minimum $\delta_{f'}(s,v)$ whose distance was decreased by the augmentation (amongst all the vertices that had their distance shortened by the augmentation, we pick the one that has the shortest path from s). Since the distance was decreased, we know that $\delta_{f'}(s,v) < \delta_f(s,v)$. We now let the path $p = s \rightsquigarrow u \rightarrow v$ be a shortest path from s to v in $G_{f'}$ containing edge $(u,v) \in E_{f'}$ and with

$$\delta_{f'}(s,u) = \delta_{f'}(s,v) - 1$$

meaning the shortest path in $G_{f'}$ to v visits u first. We chose v as the vertex whose shortest path was decreased by the augmentation with the minimum unit-distance after augmentation. We see that u has shorter distance than v, so it must not have decreased its shortest path

$$\delta_{f'}(s,u) \geq \delta_f(s,u)$$

We must have that $(u,v) \notin E_f$(that is we actually introduced the edge (u,v) when augmenting flow) since otherwise if we did have $(u,v) \in E$ we would get

$$\begin{aligned}\delta_f(s,v) &\leq \delta_f(s,u) + 1 \\ &\leq \delta_{f'}(s,u) + 1 \text{ due to the way we picked u. see above} \\ &= \delta_{f'}(s,v)\end{aligned}$$

which contradicts that v had its shortest distance decreased $\delta_{f'}(s,v) < \delta_f(s,v)$. So we have argued that $(u,v) \notin E_f$ and $(u,v) \in E_{f'}$. This means the algorithm has augmented flow along (u,v) and the edge appears by the flow augmentation. Edmonds karp always chooses $p = s \rightsquigarrow t$ as the shortest distance, so the augmentation augmented flow along the shortest path in E_f.Since the residual network G_f does not have have (u,v) it will instead have $(v,u) \in E_f$. This tells us

$$\begin{aligned}\delta_f(s,v) &= \delta_f(s,u) - 1 \text{ since there is an edge (v,u)} \\ &\leq \delta_{f'}(s,u) - 1 \\ &= \delta_{f'}(s,v) - 1 - 1 \text{ use prev eq}\end{aligned}$$

the first inequality holds since we have a shorts path to v, and there is an edge (v, u), So the shortest distance $\delta_f(s, u) = \delta_f(s, v) + 1$. The first inequality holds since u did not get a smaller shortest path by the augmentation. The second equality holds because of $\delta_{f'}(s, u) = \delta_{f'}(s, v) - 1$ from previous. The above however contradicts the original assumption $\delta_{f'}(s, v) < \delta_f(s, v)$. We have shown the opposite is true, and the shortest path monotonically increase with each flow augmentation.

3.8 Theorem 26.8

If Edmonds-karp algorithm is run on a flow network, then the total number of flow augmentations will be $O(VE)$.

Proof

We say an edge on an augmenting path is critical when $c_f(p) = c_f(u, v)$. That is when the residual capacity of p is the capacity of (u, v). Critical edges disappear from the residual network after flow augmentation. Why? Since $f(u, v) = c(u, v)$ which means that $c_f(u, v) = c(u, v) - f(u, v) = 0$. There will always be at least one critical edge. Why? This is how we choose the flow f_p. We choose the flow to be the minimum of all $c_f(u, v)$ for the edges (u, v) in p. Therefore, at least the edge we took the minimum of will be critical.

Let u, v be vertices in V that are connected by an edge $(u, v) \in E$. Edmonds-karp chooses shortest paths as augmenting paths, so when (u, v) becomes critical for the first time (it will at some point become critical since it will have augmented flow at some point otherwise the flow would always be 0), then we have

$$\delta_f(s, v) = \delta_f(s, u) + 1$$

After (u, v) is critical and flow is augmented along it, the edge disappears from G_f. It will only come back if we need to remove flow from its reversal (v, u). This happens when (v, u) appears on a (shortest) augmenting path. Let say this happens when the flow in G is f' then we have

$$\delta_{f'}(s, u) = \delta_{f'}(s, v) + 1 \text{ we take (v,u) on the SP}$$

Since the shortest paths are monotonically increasing by lemma 26.7 we get $\delta_f(s, v) \leq \delta_{f'}(s, v)$. Using this gives us

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) + 1 \\ &\geq \delta_f(s, v) + 1 \\ &= \delta_f(s, u) + 1 + 1 \end{aligned}$$

showing us that by the time an edge becomes critical to the next time it can become critical, the shortest distance has increased by at least 2. The distance from s to u is initially at least 0. Intermediate vertices on a shortest path from s to u cannot contain s, u, t (it is implied that $u \neq t$ since it is on an augmenting path and otherwise we cannot have (u, v)). Until u, if ever, becomes unreachable its distance from s to u is at most $|V| - 2$ (it cannot be s or t and so a shortest path can at most visit all vertices except these). After the first time it has become critical, it can at most become critical $(|V| - 2)/2 = |V|/2 - 1$ times This gives a total of $|V|$ times an edge can become critical. Why? Its distance is at most $|V| - 2$ and each augmentation increases the distance by 2. So the fraction gives how many times we can increase the distance by 2 until we hit the max distance of $|V| - 2$.

There are $O(E)$ (remember upper bound of $2E$) pairs of vertices that can have an edge between them in the residual network, The total number of critical edges is $O(VE)$ since $O(V)$ times can an edge become critical and this can happen for each of the $O(E)$ edges.

If we continue to use a breath first search in $O(E)$ time to find the shortest path, then we know Edmonds-Karp will only run the while loop $O(VE)$ times making the total running time $O(VE^2)$.

4 Ford Fulkerson algorithm

The algorithm keeps finding augmenting paths p, and then it updates the flow by the augmentation $f \uparrow f_p$, where we find f_p using lemma 26.2. Corollary 26.3 then tells us this gives us a flow with higher value. Theorem 26.6 now also gives us a stopping criteria when there are no more augmenting paths.

The actual algorithm looks as follows

1. for each edge $(u, v) \in G.E$
   (a) $(u, v).f = 0$
2. while there exists a path p from s to t in G_f
   (a) $c_f(p) = \min(c_f(u, v) : (u, v)\, is\, in\, p)$
   (b) for each (u, v)in p
      i. if $(u, v) \in E$
         A. $(u, v).f = (u, v).f + c_f(p)$
      ii. else
         A. $(u, v).f = (u, v).f - c_f(p)$

when (u, v) is also an edge in E(2.b.i) then we augment more flow along this path. If the edge along p is not in E then it is a reversal of an edge. This was equivalent to removing flow, so this is done.

4.1 Running time

It is assumed that all edges have integer capacities. Let us say the value of the maximum flow is f^*. We always increase the flow along an edge in the original network. Since we work with integral, we increase with at least 1, so in worst case we run the while loop f^*iterations.

We keep a graph $G' = (V, E')$where $E' = \{(u,v) : (u,v) \in E\, or\, (v,u) \in E\}$. G' will contain all edges of G, so we can keep capacities and flows in G'. Given flow f in G, then the residual network consists of the edges in G' where $c_f(u,v) > 0$ ($c_f$min of capacities on p). Using breadth first search in $O(V+E) = O(E)$ time rooted at s we can find a path to t. This makes the total running time $O(E \cdot |f^*|)$.

There are however problems if f^* is large and we each time only augments flow by 1.

5 Edmonds-Karp algorithm

It is basically ford fulkersons algorithm but we choose the augmenting path as the shortest path from s to t where the weights of edges is unit-distance, i.e 1.