AAD Assignment 1

Andreas Hammer (qgk998), Anders Munkvad (tqs326), Johan Topp (fpg662)

November 2023

26.1-1

We have an edge (u, v) in the graph G, with a maximum flow value $|f|$. The flow going through (u, v) in this flow is then defined as $f(u, v)$. We split the edge getting edges (u, x) and (x, v), leaving us with a graph G', with a maximum flow value $|f'|$. Our new vertex x must follow flow conservation, and since it only has one incoming and one outgoing edge it follows that $f'(u, x) = f'(x, v)$. To prove that the max flow value stays the same, we can use a contradiction by first assuming $|f'| < |f|$.
In the case that the flow value of G' is smaller than in G this must mean that the the flow on the path $s \sim u \sim v \sim t$ has decreased, meaning $f(u, x) = f(x, v) < f(u, v)$. Our assumptions state that $c(u, v) = c(u, x) = c(x, v)$, which means that since f is a flow $f(u, v) < c(u, v)$ and thus $f(u, x) = f(x, v) < c(u, x) = c(x, v)$. With these definitions we can conclude $|f'|$ cannot be a max flow for G' since it is possible to increase the values $f(u, x) = f(x, v)$ and still have a flow since the rest of G' is equivalent to G.
For $|f'| > |f|$ the same argument holds, but the other way around. We would be able to increase $f(u, v)$ and still have a valid flow meaning $|f|$ could not have been a max flow for G.

26.1-4

To check if the flows form a convex set we need to check if $\alpha f_1 + (1-\alpha)f_2$ satisfies two constraints, the capacity constraint and the flow conservation constraint. First we start with capacity constraint.
We need to show that $0 \leq \alpha f_1(u, v) + (1-\alpha)f_2(u, v) \leq c(u, v)$ for all (u, v). Since both f_1 and f_2 are flows we know they satisfy these constraints meaning for any values (u, v) their minimal value is 0 meaning that the lower bound can be proven as:

$$\begin{aligned}
0 &\leq \alpha f_1(u, v) + (1-\alpha)f_2(u, v) \\
0 &\leq \alpha \cdot 0 + (1-\alpha) \cdot 0 \\
0 &\leq 0
\end{aligned}$$

For the upper bound we can use that the flows maximum value is $c(u, v)$ for any (u, v).

$$\begin{aligned}
c(u, v) &\geq \alpha f_1(u, v) + (1-\alpha)f_2(u, v) \\
c(u, v) &\geq \alpha \cdot c(u, v) + (1-\alpha)c(u, v) \\
c(u, v) &\geq \alpha \cdot c(u, v) + c(u, v) - \alpha \cdot c(u, v) \\
c(u, v) &\geq c(u, v)
\end{aligned}$$

And thereby the capacity constraint holds for $\alpha f_1 + (1-\alpha)f_2$.
The second constraint is flow conservation we need for all $u \in V - \{s, t\}$,

$$\sum_{v \in V} \alpha f_1(u, v) + (1-\alpha)f_2(u, v) = \sum_{v \in V} \alpha f_1(v, u) + (1-\alpha)f_2(v, u)$$

We can rewrite this sum.

$$\left(\alpha \sum_{v\in V} f_1(u,v)\right) + \left((1-\alpha)\sum_{v\in V} f_2(u,v)\right) = \left(\alpha \sum_{v\in V} f_1(v,u)\right) + \left((1-\alpha)\sum_{v\in V} f_2(v,u)\right)$$

Since both f_1 and f_2 are flows they must satisfy the flow conservation meaning.

$$\alpha \sum_{v\in V} f_1(u,v) = \alpha \sum_{v\in V} f_1(v,u)$$

and

$$(1-\alpha)\sum_{v\in V} f_2(u,v) = (1-\alpha)\sum_{v\in V} f_2(v,u)$$

And thus flow conservation is also proven. So $\alpha f_1 + (1-\alpha)f_2$ does a convex set, since it satisfies both flow conditions.

26.1-7

For any vertex v with limit $l(v)$ in G we can split it into two vertexes, which we will call V_{in} and V_{out}. $\forall u \in V - \{v\}$, $(u,v) = (u, v_{in})$. (all incoming edges into v), we can redirect them into v_{in}. for $\{v,u\} \in G$ we replace v with (v_{out}).
Finally an edge is added between v_{in}, v_{out} with capacity $l(v)$.
Under the assumption that the limit also affects s and t, we split s into s and s_{out} where the edge $\{s, s_{out}\} = l(s)$, and for the t we split it into t and t_{in} and have an edge from $\{t_{in}, t\}$ with capacity $l(t)$. We get the final graph G' with $|V'| = 2 \cdot |V|$ and $|E'| = V + E$.

26.2-2

In order to determine the flow and capacity across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$ we look at all of the edges that crosses this cut and their corresponding flow and capacity, hence:
The edges crossing the cut are Edges crossing the cut:

$$(s,v_1),(v_2,v_1),(v_3,v_2)(v_4,v_3)(v_4,t)$$

11 + 1 + 4 + 7 + 4=27

Here all edges are crossing from S to T except for (v_3, v_2). So we add all other edge flow values together and subtract this one.

$$\text{Flow across the cut:} \quad 11 + 1 - 4 + 7 + 4 = 19$$

For the capacity we only add edges that cross from S to T, so add all edge capacities, but omit edge (v_3, v_2)

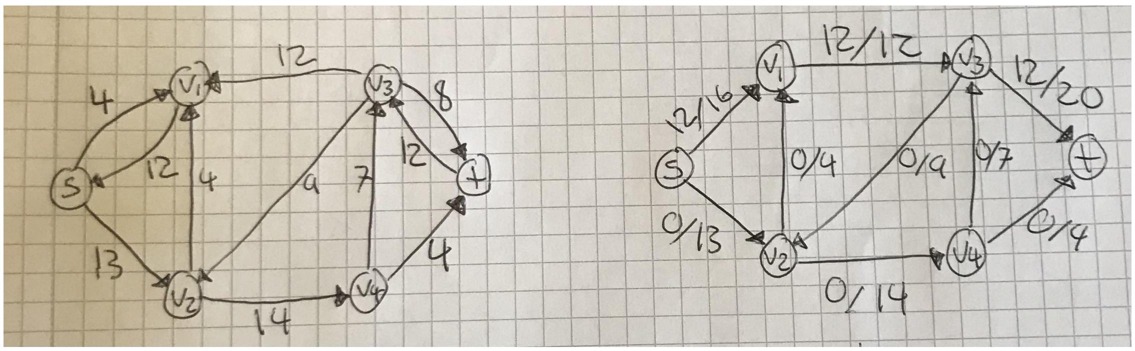
$$\text{Capacity of this cut:} \quad 16 + 4 + 7 + 4 = 31$$

26.2-4

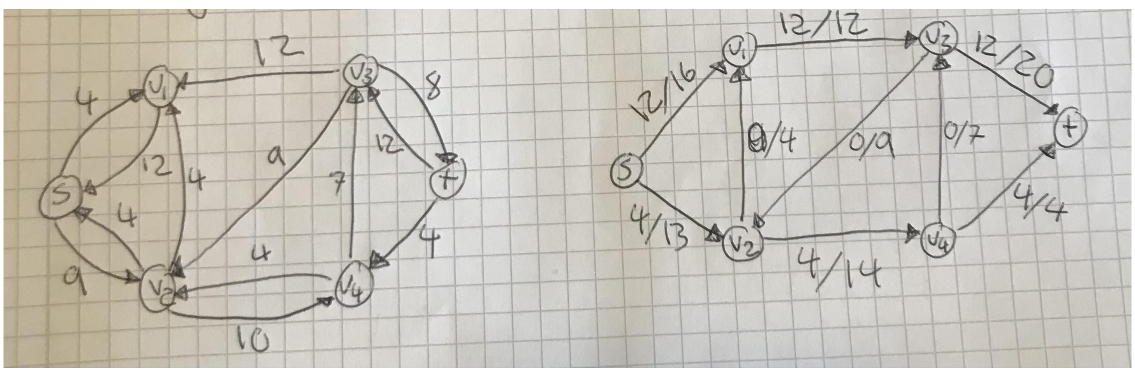
The minimum cut corresponding the maximum flow in figure 26.6:
The min cut is placed between $S = \{s, v_1, v_2, v_4\}$ and $T = \{v_3, t\}$ and the min cut value is $\{v_1, v_3\}$ + $\{v_4, v_3\}$ + $\{v_4, t\}$ = 12+7+4 = 23
Augmented flow:
The only path resulting in an augmented flow is the path in c where the path from $\{v_2, v1\}$ and $\{v_3, v_2\}$ are reduced by 4 each, so where b pushes from $\{v_3, v_2\}$ and $\{v_2, v_1\}$, the updated flow of c is not pushing anything through these two edges.

26.2-3

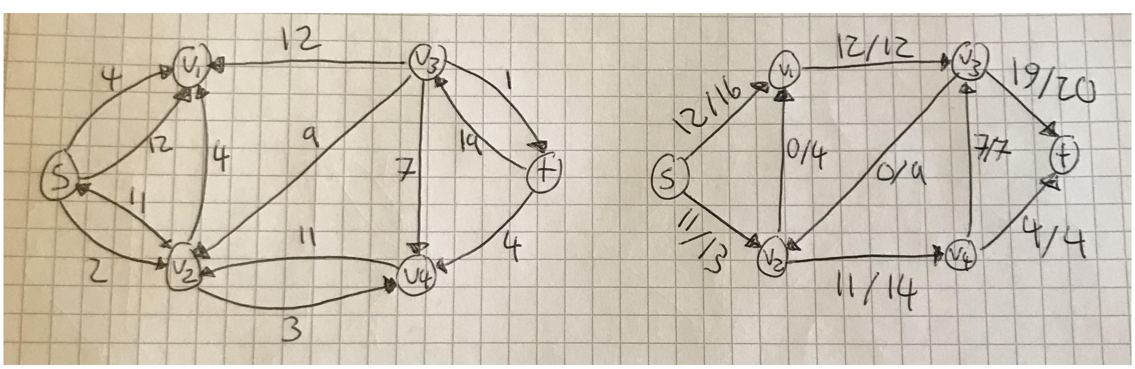
Execution of the algorithm can be seen by the figures below. The figure on the left is the residual graph and the graph on the right is the corresponding flow network after each step in the algorithm. We start by choosing the shortest augmented path using BFS from s to t, namely $s \to v_1 \to v_3 \to t$. We see that the given capacity of this augmented path is 12, as it is bounded by $v_1 \to v_3$:



We then choose the next shortest augmented path on the residual network, namely $s \to v_2 \to v_4 \to t$. We see that the given capacity of this augmented path is 4, as it is bounded by $v_4 \to t$. We update the flow network accordingly:



Once again we choose the shortest augmented path on the new residual network, $s \to v_2 \to v_4 \to v_3 \to t$. We see that the given capacity of this augmented path is 7 as it is bounded by $v_4 \to v_3$:



Finally, we see that there are no way of getting from s to t on the residual graph, hence the execution of the algorithm has finished. The final flow network and residual graph can be seen above. We see that the maximal flow of the network will be 23.

26.2-7

An augmenting path is a simple path between s and t in G_f. We define all other edges i G_f, not on the path to have value 0, and they will thus not impact flow conservation and capacity constraint.
We define every edge on the path to have same value $f_p(u,v) = c(f_p)$, this means we can rewrite the flow conservation constraint to require each vertex $u \in V - s, t$ to have the same number of non-zero edges in as they have out. And from the definition of a path we know that all vertexes on the path will have the same amount edges on the path in as out, and since these edges are the only non-zero edges, and therefore flow conservation is fulfilled.
For the capacity constraint all edges have flow $f_p(u,v) = c(f_p)$, and $c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$, so by definition of $c_f(p)$ we cannot exceed the capacity of any of the edges.
So f_p must be a flow in G_f since it satisfies the flow conservation and capacity constraint.

26.2-9

Definitions, properties and requirements:
1:
Flow conservation is the property that for $u \in V - \{s,t\} : \sum_{v \in V} f(u,v) = \sum_{v \in V} f(u,v)$. This does not depend on the capacity, but states that anything that enters a vortex will also leave it.

2:
Definition 26.4:

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) \text{ if } (u,v) \in E \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

3:
"If E contains an edge (u,v), then there is no edge (v,u) in the reverse direction"[1]

Deduction
Flow conservation:
Each of the flows f and f' in G satisfies the *capacity constraint* and *flow conservation*.
Since both f and f' are flows in G, not in the residual graph, then if there is an edge $f'(u,v)$ then there is no edge $f'(v,u)$, so the summation when doing flow conservation will be:

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

From the flow conservation we know that the first term on the left and right hand side are equal, and that the second argument on the left and right hand side are equal. Therefore flow conservation will be preserved.

Capacity constraint The capacity constraint might be broken, in case that f and f' both are max flows in the graph G, then increasing the flow will break the capacity constraint. From

[1]CLRS 709

Theorem 26.6: $|f| = c(S,T)$ for some cut (S,T) of G this will be the case for both f and f'. but the total flow would be $f + f' = 2c(S,T)$ which breaks the capacity constraint.

26.3-2

Theorem 26.10: First part:
$\forall u, v \in G\ : c(u,v) \in Z \Rightarrow |f| \in Z$
Second part:
$\forall u, v \in G\ : f(u,v) \in Z$

The Ford-Fulkerson algorithm[2] has an initial value of all edges $f(u,v) = 0$. The proof will show that each iteration of the while loop in line 3-8, the values that $f(u,v) \in Z$ for all updates and that when terminating $f(u,v) \in Z$.
Initially $f(u,v) = 0 \Rightarrow f(u,v) \in Z$ First min $c_f(u,v)|u,v \in p$ where p is a path from s to t. Since $c(u,v) \in Z$ this value is an integer.
The update in line 7 and 8 will in the first iteration be $0 + / - c_f(u,v)$ where both terms $\in Z$ so the updated result will also be an integer.
subsequent iterations the updated value of $(u,v).f$ and $(v,u).f$ can only be modified by $min\{c_f(u,v)\} : (u,v) \in p$ which is an integer and only addition/subtraction is done, then the updated version will also be an integer.

1 Andreas Hammer Disposition

- Flow constraints
- Residual network
- Augmentation of flow proof
- Max-flow min-cut theorem
- Edmond carp complexity analysis

[2]CLRS 724

2 Johan

- Flow networks, What do we want to model?
- Properties
  - Capacity constraint
  - Flow conservation
- Anti-parallel edges
- Residual network
- Augmented flow
- iterations and $f \uparrow f'(u, v)$
  - Time for Lemma 26.1? Probably not.
  - Remember antiparallel edges in G but not G_f
- Max-flow min-cut teorem 26.6. If one holds, all holds. Prove it.
- Use figure 26.6 to show min-cut/max-flow
- Time complexity of FF and EC

3 Anders Munkvad

- Introduction
  - Properties
  - Ford-Fulkerson
  - Residual network
- Example of finding the max flow (Use one of the figures from the book)
- Edmond Carp
  - Proof of running time
  - (More analysis/proofs if time permits)