Group assignment 1

Advanced Algorithms and Data Structures

Authors: psl788, wlc376, knx373

Hand-in deadline: November 29, 2022

CLRS 26.1-1

Let p be a path in G such that $(u, v) \in p$. Suppose first that $c_f(u, v) = c_f(p)$ – that is, the edge (u, v) is the bottleneck edge of p. Since we have $c(u, x) = c(x, v) = c(u, v)$, we can replace (u, v) with (u, x) and (x, v) without lowering the flow on p to obtain an equivalent path p' with $c_f(p') = c_f(p)$, and this would apply to all paths p' for which $c_f(u, v) = c_f(p)$, hence $|f|$ is the same in G and G'.

Let now p be a path in G such that $(u, v) \in p$ but $c_f(u, v) > c_f(p)$, ie. (u, v) is *not* the bottleneck of this particular path p. Then, because we have $c(u, x) = c(x, v) = c(u, v)$, we have also $c_f(u, x) > c_f(p')$ and $c_f(x, v) > c_f(p')$ in G', meaning the flow through (u, x) and (x, v) in G' will be the same as the flow through (u, v) in G. Once again the flow is unchanged from G to G'.

CLRS 26.1-4

We want to show that if f_1 and f_2 are flows, then so is $\alpha f_1 + (1-\alpha)f_2$ for all α in range $0 \leq \alpha \leq 1$ by investigating whether $\alpha f_1 + (1-\alpha)f_2$ satisfies the flow properties of capacity constraint and flow conservation. Since f_1 and f_2 are flows they already satisfy these properties.

Capacity constraint For the flow $f' = \alpha f_1 + (1-\alpha)f_2$, we require that $0 \leq f'(u,v) \leq c(u,v)$ for all $u, v \in V$:

$$0 \leq \alpha f_1(u,v) + (1-\alpha)f_2(u,v) \leq \alpha c(u,v) + (1-\alpha)c(u,v) = c(u,v)$$

Given that $0 \leq \alpha \leq 1$ and that f_1 and f_2 already satisfy the capacity constraint, the above shows that $\alpha f_1 + (1-\alpha)f_2$ satisfies the capacity constraint.

Flow conservation For the flow $f' = \alpha f_1 + (1-\alpha)f_2$ to satisfy flow conservation, we require that $\sum_{v \in V} f'(v,u) = \sum_{v \in V} f'(u,v)$ for all $u \in V - \{s,t\}$.

First, move constant factors outside of summations:

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

Then, using flow conservation of f_1 and f_2 before moving constants back in:

$$\begin{aligned} &= \alpha \sum_{v \in V} f_1(u,v) + (1-\alpha) \sum_{v \in V} f_2(u,v) \\ &= \sum_{v \in V} \Big(\alpha f_1(u,v) + (1-\alpha)f_2(u,v)\Big). \end{aligned}$$

CLRS 26.1-7

To represent vertex capacities in a flow graph using only edge capacities, we do as such: For every vertex v in G, create two vertices v_{in} and v_{out} in G', where all edges entering v in G now enter v_{in} in G', and all edges leaving v in G now leave v_{out} in G', preserving edge capacities. In addition, for every vertex v in G, create an edge $(v_{\text{in}}, v_{\text{out}})$ with capacity $c(v_{\text{in}}, v_{\text{out}}) = l(v)$ in G'.

More formally, define V' and E' as such:

$$V' = \bigcup_{v \in V} \{v_{\text{in}}, v_{\text{out}}\},$$

$$\begin{aligned} E' = &\{(u_{\text{out}}, v_{\text{in}}) \text{ with } c(u_{\text{out}}, v_{\text{in}}) = c(u,v) \ : \ (u,v) \in E\} \\ &\cup \{(v_{\text{in}}, v_{\text{out}}) \text{ with } c(v_{\text{in}}, v_{\text{out}}) = l(v) \ : \ v \in V\}. \end{aligned}$$

Since the only edge leaving an in-vertex v_{in} is the edge $(v_{\text{in}}, v_{\text{out}})$, any path from s to t in G which includes v must in its representation in G' include both v_{in} and v_{out} and thus the edge $(v_{\text{in}}, v_{\text{out}})$, and since we have $c(v_{\text{in}}, v_{\text{out}}) = l(v)$, we enforce the same capacity upon the flow.

Since V' contains two vertices for each vertex in V, and E' contains one edge for each edge in E and each vertex in V, we have:

$$\begin{aligned} |V'| &= 2|V|, \\ |E'| &= |E| + |V|. \end{aligned}$$

CLRS 26.2-2

Lemma 26.4 (CLRS) states that for a given flow f, the net flow across any cut (S, T) (such that $s \in S$ and $t \in T$) is equal to $|f|$, the value of the flow. The flow in figure 26.1(b) has value $|f| = 19$ (as per the CLRS figure text), so the net flow across the cut $(\{s, v_2, v_4\}, \{v_1, v_3, t\})$ is:

$$\begin{aligned} f(\{s, v_2, v_4\}, \{v_1, v_3, t\}) &= f(s, v_1) + f(v_2, v_1) + f(v_4, v_3) + f(v_4, t) - f(v_3, v_2) \\ &= 11 + 1 + 7 + 4 - 4 \\ &= 19 = |f|. \end{aligned}$$

The capacity of the cut is:

$$\begin{aligned} c(\{s, v_2, v_4\}, \{v_1, v_3, t\}) &= \sum_{u \in \{s, v_2, v_4\}} \sum_{v \in \{v_1, v_3, t\}} c(u, v) \\ &= c(s, v_1) + c(s, v_3) + c(s, t) + \\ &\quad\ c(v_2, v_1) + c(v_2, v_3) + c(v_2, t) + \\ &\quad\ c(v_4, v_1) + c(v_4, v_3) + c(v_4, t) \\ &= 16 + 0 + 0 + 4 + 0 + 0 + 0 + 7 + 4 \\ &= 31. \end{aligned}$$

CLRS 26.2-3

Figure 1 shows the initial flow network, with dotted arrows representing residual edges.

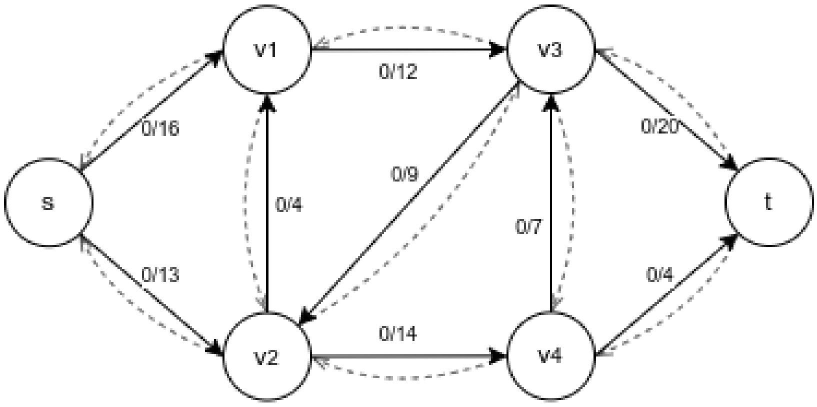

Figure 1: Original flow network

During execution of the BFS step of Edmonds-Karp, we assume that for all $v_i \in V \setminus \{s, t\}$, when iteration reaches v_i, the sink vertex t is always visited first if $(v_i, t) \in E$; otherwise, neighboring vertices v_j for which $(v_i, v_j) \in E$ are visited in order of increasing j.

The following four figures show the execution of Edmonds-Karp on the flow network, where green vertices and edges represent the augmenting paths found by BFS in the given step. Figure captions explain what happens in each step of the algorithm.

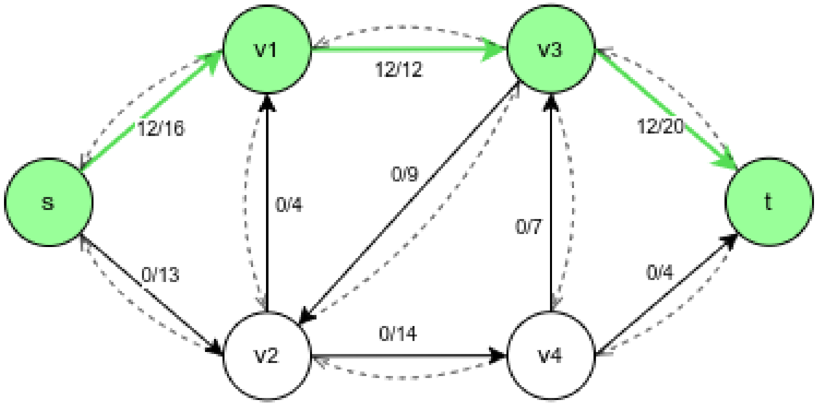

Figure 2: An initial path $p_0 = \{s, v_1, v_3, t\}$ with $c_f(p_0) = c_f(v_1, v_3) = 12$ is found using BFS.

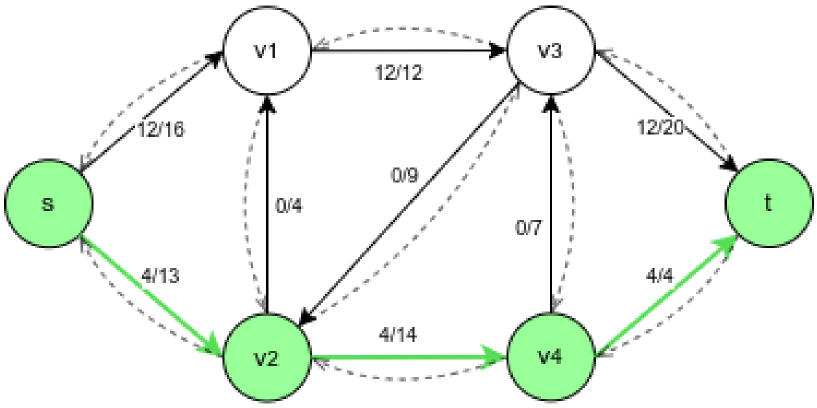

Figure 3: An augmenting path $p_1 = \{s, v_2, v_4, t\}$ is found with $c_f(p_1) = c_f(v_4, t) = 4$.

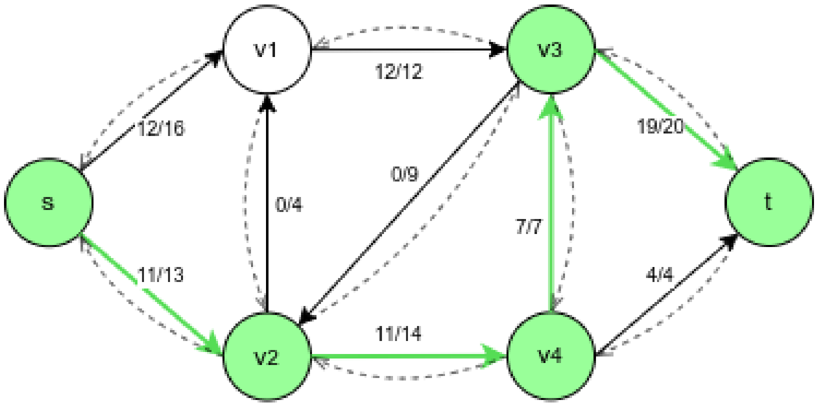

Figure 4: Another augmenting path $p_2 = \{s, v_2, v_4, v_3, t\}$ is found with $c_f(p_2) = c_f(v_4, v_3) = 7$.

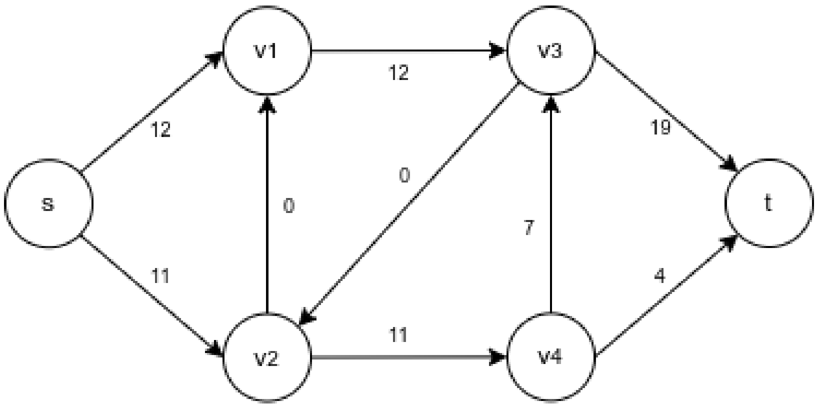

Figure 5: No more augmenting paths exist in G, and the algorithm converges. The final flow has value $|f| = c_f(p_0) + c_f(p_1) + c_f(p_2) = 23$.

CLRS 26.2-4

The max-flow min-cut theorem, Theorem 26.6 in CLRS, gives that the value of a maximum flow f in a flow network $G = (V, E)$ with source s and sink t is equal to the capacity of a minimum cut (S, T), $c(S, T)$ in G. In Figure 26.6 e) the value of the maximum flow is $|f| = 23$. From inspection of the residual network in Figure 26.6 f) the vertices v_4 and t are the only vertices that are not reachable from s, this gives that the minimum cut corresponding to the maximum flow is $(S, T) = (\{s, v_1, v_2, v_4\}, \{v_3, t\})$, which has capacity $c(S, T) = c(v_1, v_3) + c(v_4, v_3) + c(v_4, t) = 12 + 7 + 4 = 23$, as expected.

In Figure 26.6 the only augmenting path that cancels flow is the flow given in Figure 26.6 c), as it is the only flow that utilizes the reverse edges in the residual network. It sends a flow of value 4 in the residual network along the two reverse edges (v_1, v_2) and (v_2, v_3).

CLRS 26.2-7

Lemma 26.2 in CLRS states that for a flow network $G(V,E)$ with a flow f in G and an augmenting path p in G_f, the function $f_p : V \times V \to \mathbb{R}$ defined by

$$f_p = \begin{cases} c_f(p) & \text{if } (u,v) \text{ is on } p, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$.

Proof: From the definition of residual networks it is given that all edges, (u,v), with a capacity $c_f(u,v) = 0$ are not in G_f. An augmenting path, p, is a simple path in G_f from s to t. Together with the definition $c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$ gives that $c_f(p) > 0$.

A flow in a residual network can be defined as one that satisfies the definition of a flow, but with respect to capacities c_f in G_f.

Capacity constraint in G_f: For all $u,v \in V$, it is required that $0 \leq f(u,v) \leq c_f(u,v)$. From the definition of f_p, equation (1), the flow along any edge, (u,v), not on p will have a flow of 0. For all edges, (u,v) on p the flow will be $c_f(p)$. The residual capacity of p is defined as $c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$ together with the fact that $c_f(u,v) > 0$ for all edges in G_f. This gives that the capacity constraint is fulfilled for all edges on p and all edges not on p, which gives that capacity constraint is therefore fulfilled for f_p.

Flow conservation: For all $u,v \in V - \{s,t\}$ it is required that

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

Since p is an augmenting path it is a simple path from s to t and therefore consists of only distinct edges and distinct vertices. For all $v \in V - \{s,t\}$ that are not part of p all $f(v,u) = 0$ and all $f(u,v) = 0$, which gives that flow conservation is fulfilled for these vertices. For all $v \in V - \{s,t\}$ part of the augmenting path p together with the definition of f_p and the fact that p is a simple path from s to t gives that there are exactly one nonzero ingoing flow and one outgoing flow for v, which has value $c_f(p)$

$$\sum_{v \in V} f_p(v,u) = \sum_{v \in V} f_p(u,v) = c_f(p).$$

Flow conservation is therefore fulfilled for f_p.

Value of the flow: Since p is a simple path from s to t the path only has one edge with starting point s and no edges with end point s. The value of the flow f_p therefore becomes

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

This gives that f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$.

CLRS 26.2-9

The flows f and f' satisfy both the properties of flow conservation and capacity constraint by definition. Combined with the fact that both f and f' are flows in the graph G, the augmented flow $f \uparrow f'$ will also satisfy the property of flow conservation.

However, the augmented flow will not necessarily satisfy the capacity constraint. Consider the case where f is a maximum flow. Then, augmenting f by f' will result in a new flow whose value is larger than $|f|$:

$$|f \uparrow f'| = |f| + |f'| > |f|$$

But f was assumed to be a maximum flow, and hence there is a contradiction. Thus, the augmented flow violates the capacity constraint.

CLRS 26.3-2

As hinted in CLRS, we will prove theorem 26.10 using induction over the number of iterations of the while loop, and we use the pseudocode given on p. 724 as a basis.

Induction base case

Before the while-loop, the flow f is zero-initialized (lines 1-2). Inside the while-loop body, the flow for each edge (u, v) in the newly discovered augmenting path p_1 is updated by either adding $c_f(p_1)$ to the edge flow *or* by subtracting $c_f(p_1)$ from the corresponding edge (v, u) in the residual graph. Since all edge flows were 0 prior to the first iteration, all edge flows will be either zero, $c_f(p_1)$, or $-c_f(p_1)$ after the first iteration, and since c takes only integral values, we have $(u, v).f = f(u, v) \in \mathbb{Z}$ for all vertex pairs u, v.

Since the value of the flow $|f|$ is a sum of flows, and because a sum of integers is an integer, we also have that $|f|$ is integral.

Induction step

Assume that after the n'th iteration of the while-loop body, we have $(u, v).f = f(u, v) \in \mathbb{Z}$ for all pairs of vertices (u, v), and that $|f|$ is integral.

In iteration $(n + 1)$, each flow on the augmenting path p_{n+1} is updated by addition of either $c_f(p_{n+1})$ or $-c_f(p_{n+1})$, while edges *not* on p_{n+1} is not updated. Because each flow in the network was integral after the n'th iteration and changed by an integral amount during the $(n + 1)$'st iteration, we have that each flow $f(u, v)$ is integral after the $(n + 1)$'st iteration. As before the value of the flow $|f|$ is a sum of flows and thus remains integral.

Summaries

psl788

- Short introduction to the max flow problem
- Ford-Fulkerson method and show example
- Max-flow min-cut theorem and proof

wlc376 – max flow presentation disposition

- present problem: given flow network (digraph whose edges have flow/-capacity, and distinct source/target vertices); compute the maximum flow through the flow network.
- useful for various problems; even has application in image segmentation.
- draw very simple example with eg. 4 vertices and 5 edges; give formula for $|f|$.
- describe Ford-Fulkerson and Edmonds-Karp: repeatedly find new paths p and augment flow – $|f \uparrow f_p| = |f| + |f_p|$ – until no new paths.
- mention residual graph..? probably not relevant, but keep in mind in case of questions.
- explain that Ford-Fulkerson and Edmonds-Karp have runtime complexities of $O(|E| \cdot |f|)$ and $O(|V| \cdot |E|^2)$, respectively.
- show Edmonds-Karp complexity.

knx373 - Max Flow

- Introduction to the max flow problem and definition of a flow:
  - Capacity constraint
  - Flow conservation
  - Value of flow
- Show a simple network, add a shortest path flow and introduce the concept of the residual graph through the residual capacity.
- Introduce Ford Fulkerson and how to add a flow from G_f to existing flow in G.

- Run Edmonds-Karp on example, example is chosen such that maximum 3 iterations are needed to terminate.
- Max-flow min-cut theorem with proof and show the min-cut for the example.
- If time allows it, explain time complexity of Ford-Fulkerson and Edmonds-Karp, prioritize Edmonds-Karp. Edmonds-Karp independent of max flow for the network.