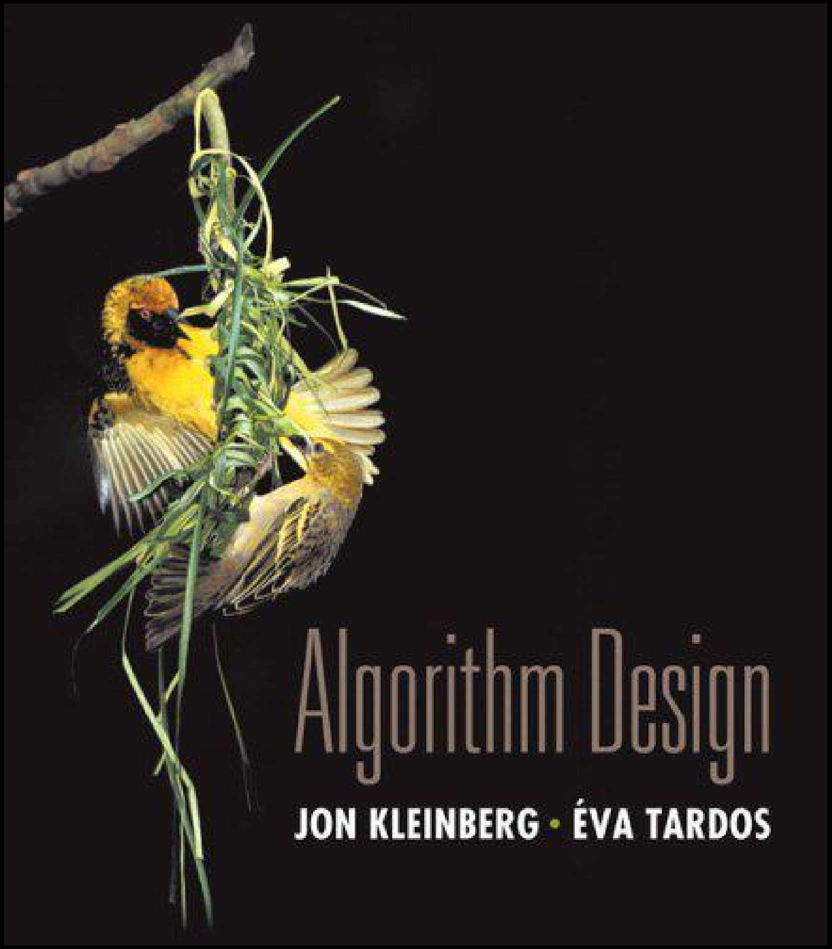


Chapter 7

Network Flow

Slides by Kevin Wayne.

7.3 Choosing Good Augmenting Paths

Choosing good augmenting paths

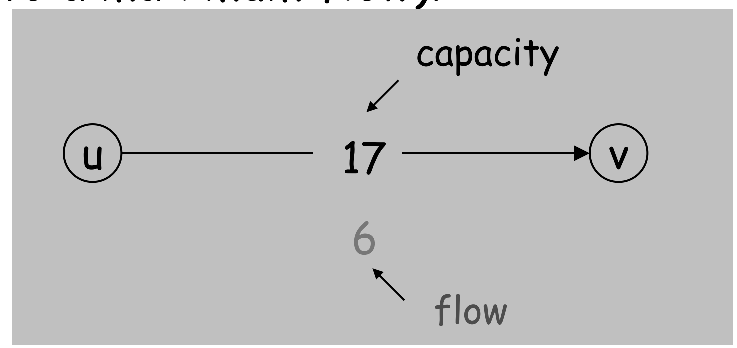
Use care when selecting augmenting paths

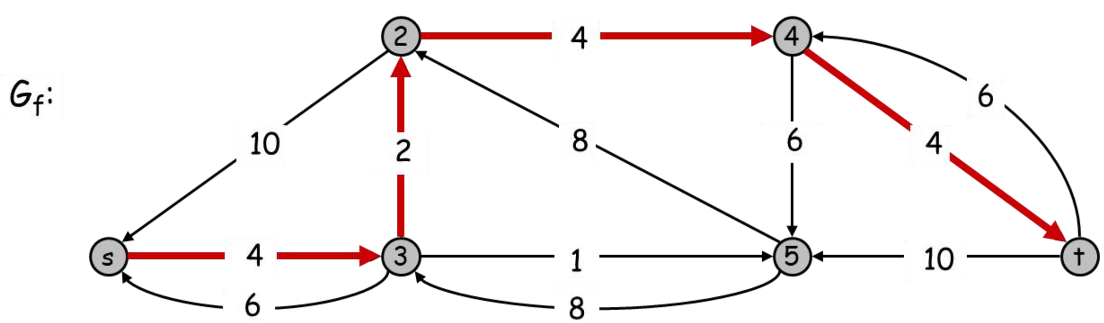
- Some choices lead to exponential algorithms
- Clever choice lead to polynomial algorithms

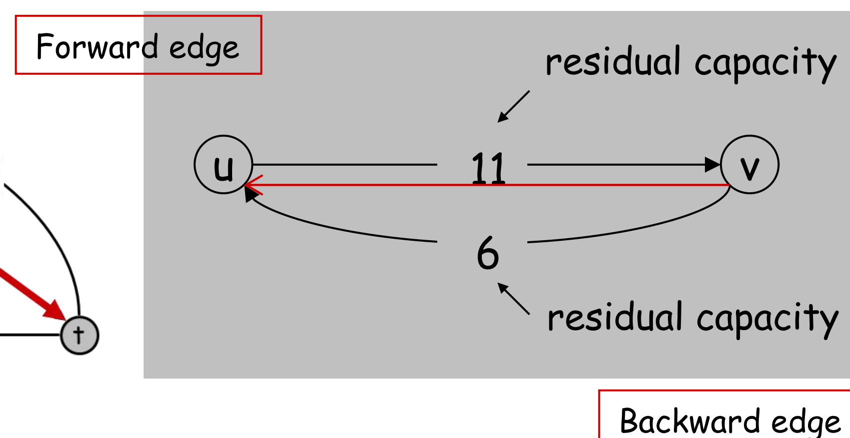
Pathology. When edge capacities can be irrational, no guarantee that Ford-Fulkerson terminates (or converges to a maximum flow)!

Goal. Choose augmenting paths so that:

- Can find augmenting paths efficiently.
- Few iterations







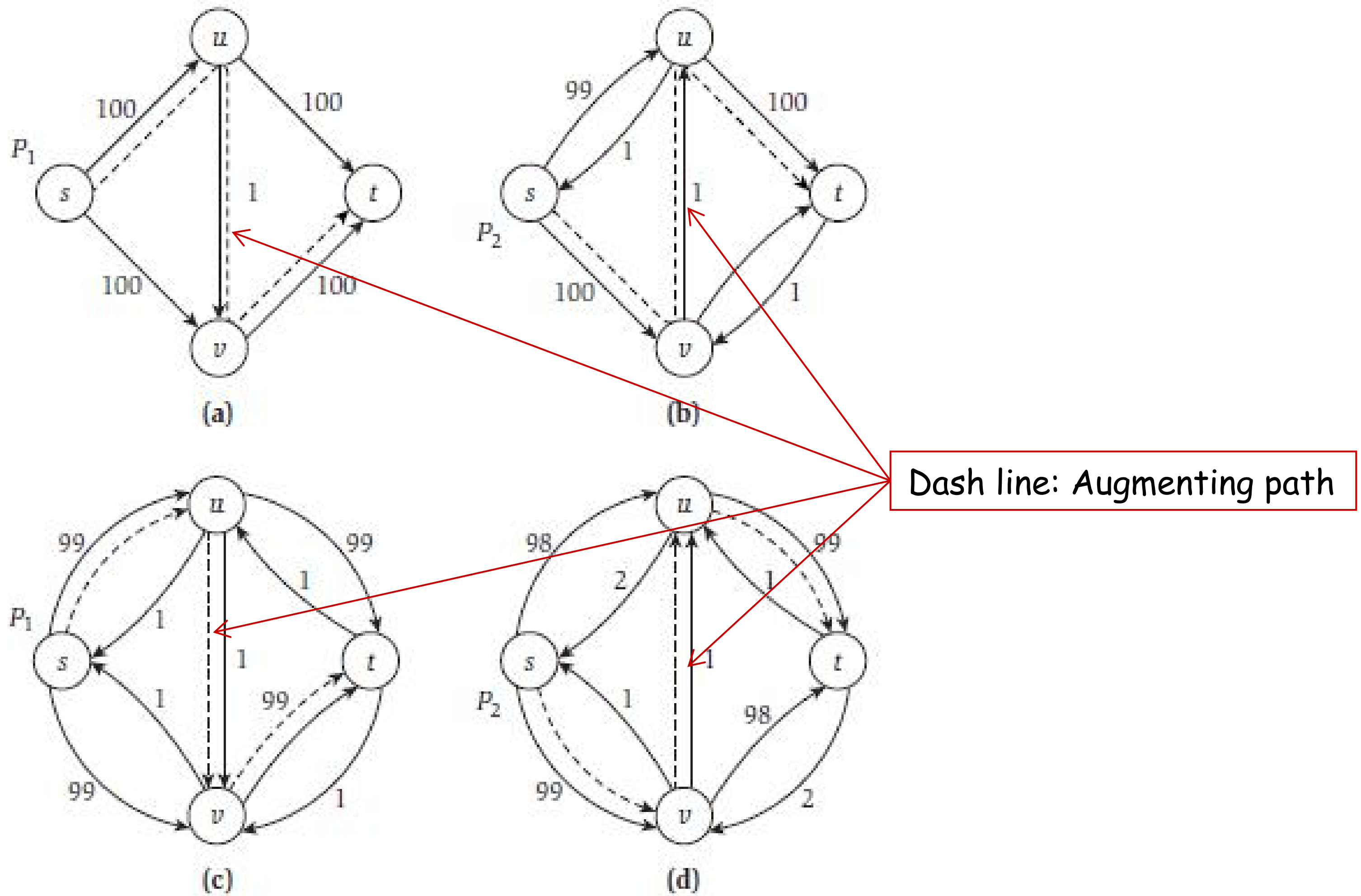


Figure Parts (a) through (d) depict four iterations of the Ford-Fulkerson Algorithm using a bad choice of augmenting paths: The augmentations alternate between the path P_1 through the nodes s, u, v, t in order and the path P_2 through the nodes s, v, u, t in order.

Choosing good augmenting paths

Choose augmenting paths with:

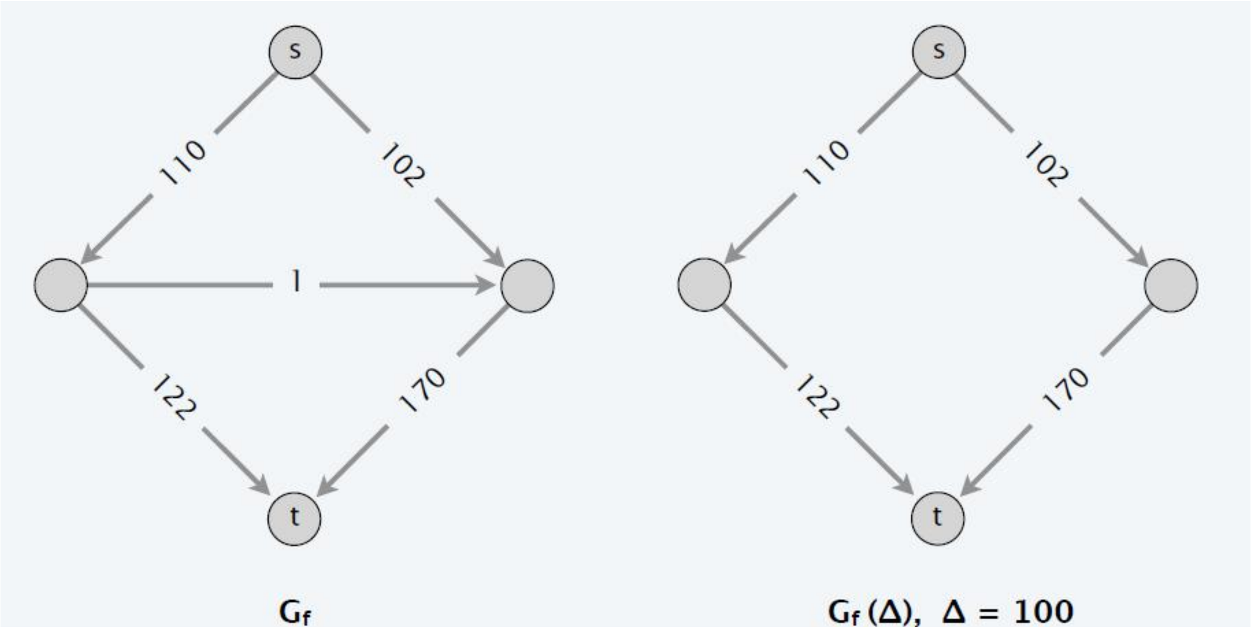
- Max bottleneck capacity ("fattest"). ← how to find?
- Sufficiently large bottleneck capacity. ← next
- Fewest edges. ← ahead

Capacity-scaling algorithm

Overview. Choosing augmenting paths with "large" bottleneck capacity.

though not necessarily largest

- Maintain scaling parameter Δ.
- Let $G_f(\Delta)$ be the part of the residual graph containing only those edges with capacity $\geq \Delta$.
- Any augmenting path in $G_f(\Delta)$ has bottleneck capacity $\geq \Delta$.



```
Scaling Max-Flow
  Initially f(e) = 0 for all e in G
  Initially set Δ to be the largest power of 2 that is no larger
       than the maximum capacity out of s: Δ ≤ max_{e out of s} c_e
   While Δ ≥ 1
      While there is an s-t path in the graph G_f(Δ)
         Let P be a simple s-t path in G_f(Δ)
         f' = augment(f, P)
         Update f to be f' and update G_f(Δ)
      Endwhile
      Δ = Δ/2
   Endwhile
Return f
```

Capacity-scaling algorithm: proof of correctness

Assumption: All edge capacities are integers between 1 and C.

Invariant. The scaling parameter Δ is a power of 2.
Pf. Initially a power of 2 (largest power of 2 $\leq C$); each phase divides Δ by exactly 2.

Integrality invariant. Throughout the algorithm, every edge flow f(e) and residual capacity $c_f(e)$ is an integer.
Pf. Same as for genetic Ford-Fulkerson.

Theorem. If capacity-scaling algorithm terminates, then f is a max flow.
Pf.

- By integrality invariant, when $\Delta = 1 \Rightarrow G_f(\Delta) = G_f$.
- Upon termination of $\Delta = 1$ phase, there are no augmenting paths.
- Result follows augmenting path theorem.

Capacity-scaling algorithm: analysis of running time

Lemma 1. There are $1 + \lfloor \log_2 C \rfloor$ scaling phases.

Pf. Initial $C/2 < \Delta \leq C$; Δ decreases by a factor of 2 in each iteration.

Lemma 2. Let f be the flow at the end of a Δ-scaling phase, then the max-flow value $\leq v(f) + m\Delta$.

Pf. Next slide.

or equivalently, at the end of a 2 Δ-scaling phase

Lemma 3. There are $\leq 2m$ augmentations per scaling phase.

Pf.

- Let f be the flow at the beginning of a Δ-scaling phase.
- Lemma 2 ➔ max-flow value $\leq v(f) + m(2\Delta)$.
- Each augmentation in a Δ-scaling phase increases $v(f)$ by at least Δ.

Theorem. The capacity-scaling algorithm takes $O(m^2 \log C)$ time.

Pf.

- Lemma 1 + Lemma 3 ➔$O(m \log C)$ augmentations.
- Finding an augmenting path takes $O(m)$ time.

Capacity-scaling algorithm: analysis of running time

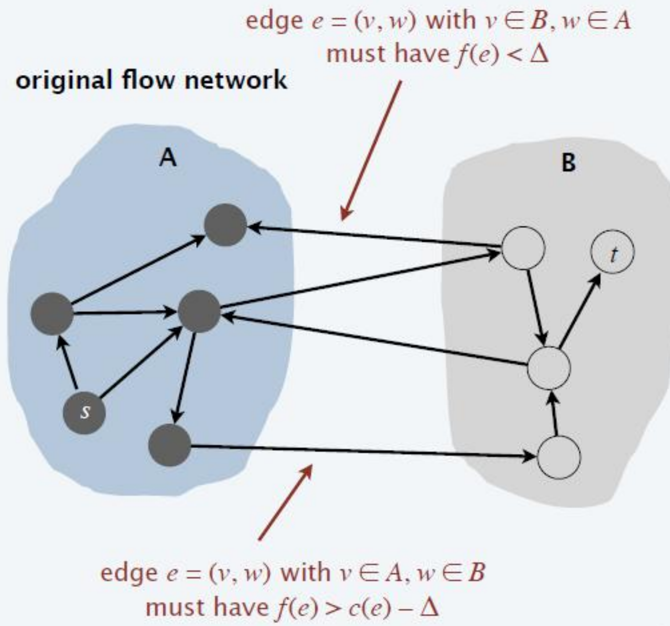
Lemma 2. Let f be the flow at the end of a Δ-scaling phase, then the max-flow value $\le v(f) + m\Delta$.

Pf.

- We show there exists a cut(A,B) such that cap(A,B) $\le v(f) + m\,\Delta$.
- Choose A to be the set of nodes reachable from s in $G_f(\Delta)$.
- By definition of flow f: $t \notin A$.

$$
\begin{aligned}
val(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\
&\ge \sum_{e \text{ out of } A} (c(e) - \Delta) - \sum_{e \text{ in to } A} \Delta \\
&\ge \sum_{e \text{ out of } A} c(e) - \sum_{e \text{ out of } A} \Delta - \sum_{e \text{ in to } A} \Delta \\
&\ge cap(A,B) - m\Delta \quad \blacksquare
\end{aligned}
$$

flow value lemma



Residual capacity:

$$
c_f(e) = \begin{cases} c(e) - f(e) & \text{if } e \in E \\ f(e) & \text{if } e^R \in E \end{cases}
$$

Shortest augmenting path

Q. How to choose next augmenting path in Ford-Fulkerson?
A. Pick one that uses the fewest edges.

can find via BFS

SHORTEST-AUGMENTING-PATH(G)

FOREACH $e \in E$: $f(e) \leftarrow 0$.

$G_f \leftarrow$ residual network of G with respect to flow f.

WHILE (there exists an $s \rightarrow t$ path in G_f)

$P \leftarrow$ BREADTH-FIRST-SEARCH(G_f).

$f \leftarrow$ AUGMENT(f, c, P).

Update G_f.

RETURN f.
