Advanced algorithms and data structures

Lecture 1: Max Flow 1

Jacob Holm (`jaho@di.ku.dk`)

November 20th 2023

Good Afternoon.

Today's Lecture

Introduction

Max flow

- Definitions
- Ford-Fulkerson Method
- Max flow/Min cut Theorem

Summary

Introduction to AADS

This course is mostly about algorithms and how to analyse them.

We want *efficient* solutions when possible, where the meaning of "efficient" may depend on the problem.

In particular, we will focus on Polynomial time algorithms (fast) versus Exponential time algorithms (slow).

This weeks topic is a polynomial time algorithm. Later we will touch on some problems where we don't know or expect expect polynomial time algorithms to exist, and some ways to deal with that.

Introduction to AADS

This course is mostly about algorithms and how to analyse them.

We want *efficient* solutions when possible, where the meaning of "efficient" may depend on the problem.

In particular, we will focus on Polynomial time algorithms (fast) versus Exponential time algorithms (slow).

This weeks topic is a polynomial time algorithm. Later we will touch on some problems where we don't know or expect expect polynomial time algorithms to exist, and some ways to deal with that.

Introduction to AADS

This course is mostly about algorithms and how to analyse them.

We want *efficient* solutions when possible, where the meaning of "efficient" may depend on the problem.

In particular, we will focus on Polynomial time algorithms (fast) versus Exponential time algorithms (slow).

This weeks topic is a polynomial time algorithm. Later we will touch on some problems where we don't know or expect expect polynomial time algorithms to exist, and some ways to deal with that.

Introduction to AADS

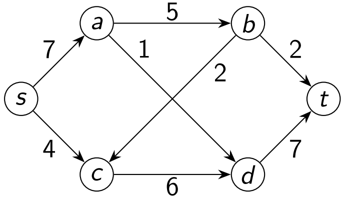
This course is mostly about algorithms and how to analyse them.

We want *efficient* solutions when possible, where the meaning of "efficient" may depend on the problem.

In particular, we will focus on Polynomial time algorithms (fast) versus Exponential time algorithms (slow).

This weeks topic is a polynomial time algorithm. Later we will touch on some problems where we don't know or expect expect polynomial time algorithms to exist, and some ways to deal with that.

Flow network



Definition

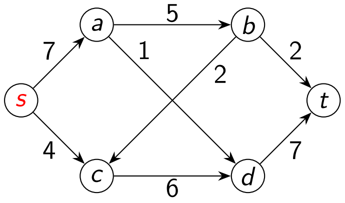
A *flow network* consists of a directed graph $G = (V, E)$, a source $s \in V$, a sink $t \in V \setminus \{s\}$, and a capacity function $c : V \times V \to \mathbb{R}$ such that

- $c(u, v) \geq 0$ for all $u, v \in V$, and
- if $(u, v) \notin E$ then $c(u, v) = 0$

We will assume that G has no self-loops and no antiparallel edges.

Example of a flow network.
Graph with nodes s and t. Send goods/data/water from s to t. Can not accumulate in intermediate nodes, so what goes in must come out (flow conservation).
Capacities, capacity constraint.
Max flow can be used by itself, or as black box for solving other problems.
No self-loops or antiparallel edges allowed by our algorithms/theorems. This is without loss of generality as we can always get rid of them.

Flow network



Definition

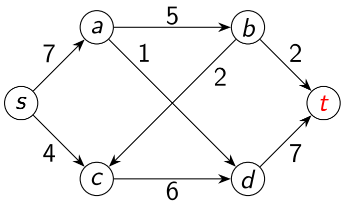
A *flow network* consists of a directed graph $G = (V, E)$, a source $s \in V$, a sink $t \in V \setminus \{s\}$, and a capacity function $c : V \times V \to \mathbb{R}$ such that

- $c(u, v) \geq 0$ for all $u, v \in V$, and
- if $(u, v) \notin E$ then $c(u, v) = 0$

We will assume that G has no self-loops and no antiparallel edges.

Example of a flow network.
Graph with nodes s and t. Send goods/data/water from s to t. Can not accumulate in intermediate nodes, so what goes in must come out (flow conservation).
Capacities, capacity constraint.
Max flow can be used by itself, or as black box for solving other problems.
No self-loops or antiparallel edges allowed by our algorithms/theorems. This is without loss of generality as we can always get rid of them.

Flow network



Definition

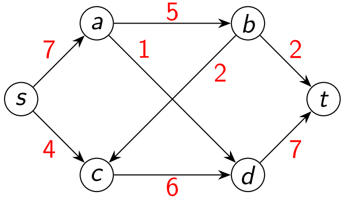
A *flow network* consists of a directed graph $G = (V, E)$, a source $s \in V$, a sink $t \in V \setminus \{s\}$, and a capacity function $c : V \times V \to \mathbb{R}$ such that

- $c(u, v) \geq 0$ for all $u, v \in V$, and
- if $(u, v) \notin E$ then $c(u, v) = 0$

We will assume that G has no self-loops and no antiparallel edges.

Example of a flow network.
Graph with nodes s and t. Send goods/data/water from s to t. Can not accumulate in intermediate nodes, so what goes in must come out (flow conservation).
Capacities, capacity constraint.
Max flow can be used by itself, or as black box for solving other problems.
No self-loops or antiparallel edges allowed by our algorithms/theorems. This is without loss of generality as we can always get rid of them.

Flow network



Definition

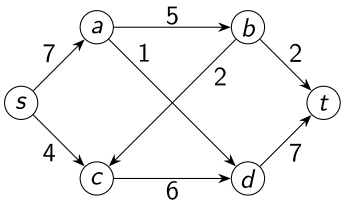
A *flow network* consists of a directed graph $G = (V, E)$, a source $s \in V$, a sink $t \in V \setminus \{s\}$, and a capacity function $c : V \times V \to \mathbb{R}$ such that

- $c(u, v) \geq 0$ for all $u, v \in V$, and
- if $(u, v) \notin E$ then $c(u, v) = 0$

We will assume that G has no self-loops and no antiparallel edges.

Example of a flow network.
Graph with nodes s and t. Send goods/data/water from s to t. Can not accumulate in intermediate nodes, so what goes in must come out (flow conservation).
Capacities, capacity constraint.
Max flow can be used by itself, or as black box for solving other problems.
No self-loops or antiparallel edges allowed by our algorithms/theorems. This is without loss of generality as we can always get rid of them.

Flow network

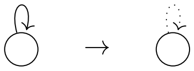


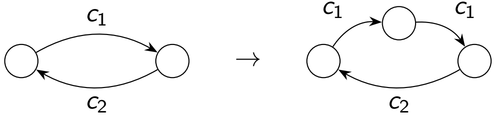
Definition

A *flow network* consists of a directed graph $G = (V, E)$, a source $s \in V$, a sink $t \in V \setminus \{s\}$, and a capacity function $c : V \times V \to \mathbb{R}$ such that

- $c(u, v) \geq 0$ for all $u, v \in V$, and
- if $(u, v) \notin E$ then $c(u, v) = 0$

We will assume that G has no self-loops and no antiparallel edges.





Example of a flow network.
Graph with nodes s and t. Send goods/data/water from s to t. Can not accumulate in intermediate nodes, so what goes in must come out (flow conservation).
Capacities, capacity constraint.
Max flow can be used by itself, or as black box for solving other problems.
No self-loops or antiparallel edges allowed by our algorithms/theorems. This is without loss of generality as we can always get rid of them.

Flow and max-Flow

Could also have defined as net flow entering t.

Definition

A flow in flow network (G, s, t, c) is a function $f : V \times V \to \mathbb{R}$ such that:

1. $\forall u, v \in V$: $0 \leq f(u, v) \leq c(u, v)$ (capacity constraints)
2. $\forall v \in V \setminus \{s, t\}$: $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$ (flow conservation)

   Equivalently: $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$. Why?

Definition

The value $|f|$ of a flow f is defined as:

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

Definition

A max-flow is a flow of maximum value.

Flow and max-Flow

Could also have defined as net flow entering t.

Definition

A flow in flow network (G, s, t, c) is a function $f : V \times V \to \mathbb{R}$ such that:

1. $\forall u, v \in V$: $0 \leq f(u, v) \leq c(u, v)$ (capacity constraints)
2. $\forall v \in V \setminus \{s, t\}$: $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$ (flow conservation)

   Equivalently: $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$. Why?

Definition

The value $|f|$ of a flow f is defined as:

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

Definition

A max-flow is a flow of maximum value.

Flow and max-Flow

Could also have defined as net flow entering t.

Definition

A flow in flow network (G, s, t, c) is a function $f : V \times V \to \mathbb{R}$ such that:

1. $\forall u, v \in V$: $0 \leq f(u, v) \leq c(u, v)$ (capacity constraints)
2. $\forall v \in V \setminus \{s, t\}$: $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$ (flow conservation)

   Equivalently: $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$. Why?

Definition

The value $|f|$ of a flow f is defined as:

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

Definition

A max-flow is a flow of maximum value.

Flow and max-Flow

Could also have defined as net flow entering t.

Definition

A flow in flow network (G, s, t, c) is a function $f : V \times V \to \mathbb{R}$ such that:

1. $\forall u, v \in V$: $0 \leq f(u, v) \leq c(u, v)$ (capacity constraints)
2. $\forall v \in V \setminus \{s, t\}$: $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$ (flow conservation)

   Equivalently: $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$. Why?

Definition

The value $|f|$ of a flow f is defined as:

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

Definition

A max-flow is a flow of maximum value.

Flow and max-Flow

Definition
A flow in flow network (G, s, t, c) is a function $f : V \times V \to \mathbb{R}$ such that:

1. $\forall u, v \in V$: $0 \le f(u, v) \le c(u, v)$ (capacity constraints)
2. $\forall v \in V \setminus \{s, t\}$: $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$ (flow conservation)

   Equivalently: $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w)$. Why?

Definition
The value $|f|$ of a flow f is defined as:

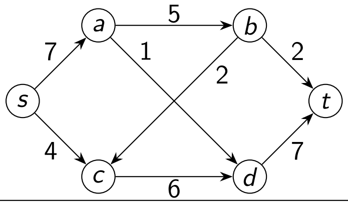
$$|f| := \sum_{v \in V} f(s, v) - \sum_{u \in V} f(u, s) = \sum_{v \in V} \big(f(s, v) - f(v, s)\big)$$

Definition
A max-flow is a flow of maximum value.

Could also have defined as net flow entering t.

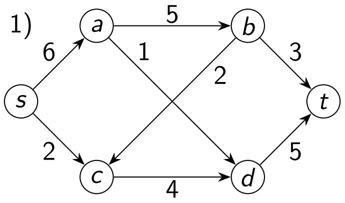
Examples: Which of these are flows? What are the values?

Flow network
(capacities on edges)

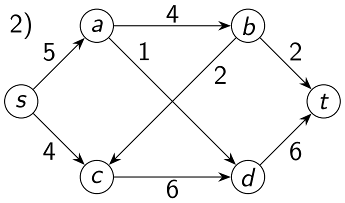


Candidate flows (flow on edges)

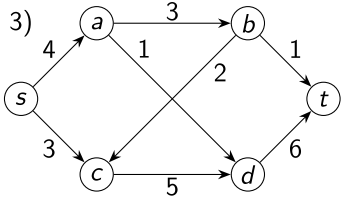
1)



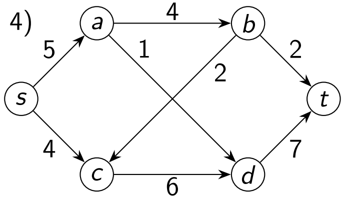
2)



3)



4)



Ex 1: No. Capacity violation at (b, t).

Ex 2: No. Flow conservation violation at d.

Ex 3: Yes. Value 7

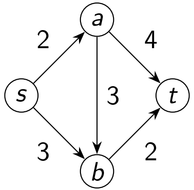
Ex 4: Yes. Value 9. Actually a max flow.

Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

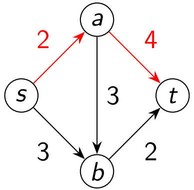
function $\textsc{Ford-Fulkerson}(G = (V, E), s, t, c)$
    $f \leftarrow 0$
    while $\exists$ augmenting path p from s to t **do**
        Send as much flow as possible along p and "add" this to f.
    return f



Ford-Fulkerson Method (informal)

Example 1:
Example 2: cancelling flow

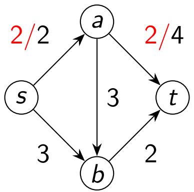
function FORD-FULKERSON($G = (V, E)$, s, t, c)
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f



Ford-Fulkerson Method (informal)

Example 1:
Example 2: cancelling flow

function FORD-FULKERSON($G = (V, E)$, s, t, c)
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f

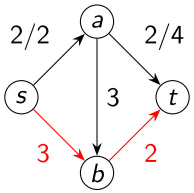


Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

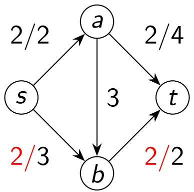
function $\textsc{Ford-Fulkerson}(G = (V, E),\ s,\ t,\ c)$
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f



Ford-Fulkerson Method (informal)

Example 1:
Example 2: cancelling flow

function Ford-Fulkerson($G = (V, E)$, s, t, c)
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f

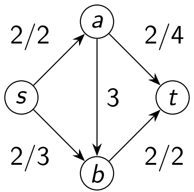


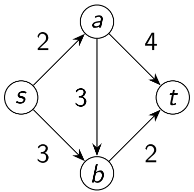
Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

function $\textsc{Ford-Fulkerson}(G = (V, E), s, t, c)$
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and “add” this to f.
  return f



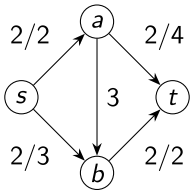


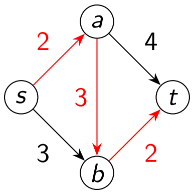
Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

function FORD-FULKERSON($G = (V, E)$, s, t, c)
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f



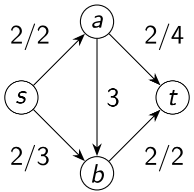


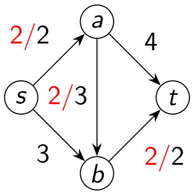
Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

function FORD-FULKERSON($G = (V, E)$, s, t, c)
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f

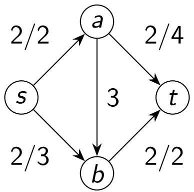


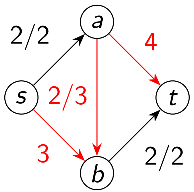


Ford-Fulkerson Method (informal)

Example 1:
Example 2: cancelling flow

function $\text{Ford-Fulkerson}(G = (V, E), s, t, c)$
  $f \leftarrow 0$
  while $\exists$ augmenting path p from s to t **do**
    Send as much flow as possible along p and "add" this to f.
  return f



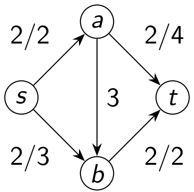


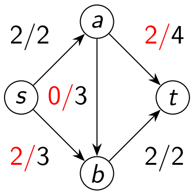
Ford-Fulkerson Method (informal)

Example 1:

Example 2: cancelling flow

function $\textsc{Ford-Fulkerson}(G = (V, E), s, t, c)$
    $f \leftarrow 0$
    while $\exists$ augmenting path p from s to t **do**
        Send as much flow as possible along p and "add" this to f.
    return f





Break

Example: What are the edges and residual capacities.

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

$$c_f(u, v) := \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E \text{ (i.e. how much more could be sent)} \\ f(v, u) & \text{if } (v, u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).

Example: What are the edges and residual capacities.

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).

Example: What are the edges and residual capacities.

Example: What are the edges and residual capacities.

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

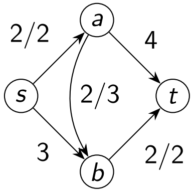
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

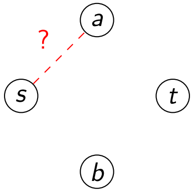
$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).





Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

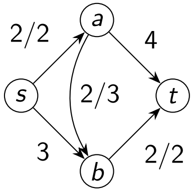
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

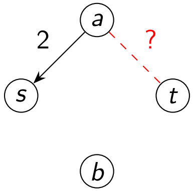
Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).



Example: What are the edges and residual capacities.



Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

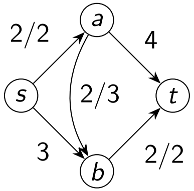
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

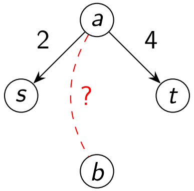
$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).





Example: What are the edges and residual capacities.

Example: What are the edges and residual capacities.

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

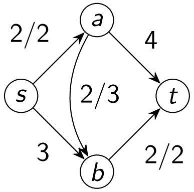
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

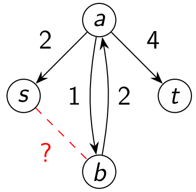
$$c_f(u, v) := \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E \text{ (i.e. how much more could be sent)} \\ f(v, u) & \text{if } (v, u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).





Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

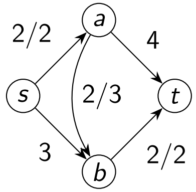
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

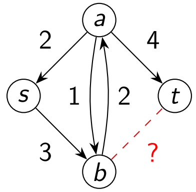
$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).





Example: What are the edges and residual capacities.

Residual network

Recall that G has no self-loops or anti-parallel edges.

Definition

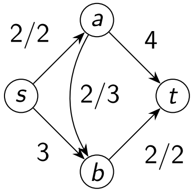
Given a flow f in (G, s, t, c), the residual capacity is the function $c_f : V \times V \to \mathbb{R}$ defined by

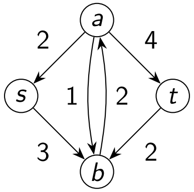
$$c_f(u,v) := \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \text{ (i.e. how much more could be sent)} \\ f(v,u) & \text{if } (v,u) \in E \text{ (i.e. how much can be cancelled)} \\ 0 & \text{otherwise} \end{cases}$$

Definition

The residual network consist of the graph $G_f := (V, E_f)$ where $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$, together with s, t, and the capacity function c_f.

Note that (G_f, s, t, c_f) is a flow network (but may have antiparallel edges).





Example: What are the edges and residual capacities.

Ford-Fulkerson Method

function FORD-FULKERSON($G = (V, E)$, s, t, c)
    $f \leftarrow 0$
    while $\exists$ (augmenting) path p from s to t in G_f **do**
        Find a max flow f_p along p in G_f.
        $f \leftarrow f \uparrow f_p$
    return f

What is the max flow along a path?

Illustrate "max flow along p".
$f \uparrow f'$ defined later.
Not an algorithm because we don't specify how to pick the path p.

Ford-Fulkerson Method

function $\textsc{Ford-Fulkerson}(G = (V, E), s, t, c)$
  $f \leftarrow 0$
  while $\exists$ (augmenting) path p from s to t in G_f **do**
    Find a max flow f_p along p in G_f.
    $f \leftarrow f \uparrow f_p$
  return f

What is the max flow along a path?

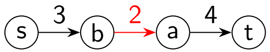
$$s \xrightarrow{3} b \xrightarrow{2} a \xrightarrow{4} t$$

Illustrate "max flow along p".
$f \uparrow f'$ defined later.
Not an algorithm because we don't specify how to pick the path p.

Ford-Fulkerson Method

function $\textsc{Ford-Fulkerson}(G = (V, E), s, t, c)$
    $f \leftarrow 0$
    while $\exists$ (augmenting) path p from s to t in G_f **do**
        Find a max flow f_p along p in G_f.
        $f \leftarrow f \uparrow f_p$
    return f

What is the max flow along a path?



Illustrate "max flow along p".
$f \uparrow f'$ defined later.
Not an algorithm because we don't specify how to pick the path p.

Augmented flow, Lemma 1

Definition

Given a flow f in G and a flow f' in G_f, the augmented flow $f \uparrow f'$: $V \times V \to \mathbb{R}$ is

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

Lemma

$f \uparrow f'$ is a flow in G of value $|f \uparrow f'| = |f| + |f'|$.

Proof.

Maybe later. Need to prove capacity constraints are satisfied and flow conservation holds. □

Augmented flow, Lemma 1

Definition

Given a flow f in G and a flow f' in G_f, the augmented flow $f \uparrow f' \colon V \times V \to \mathbb{R}$ is

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

Lemma

$f \uparrow f'$ is a flow in G of value $|f \uparrow f'| = |f| + |f'|$.

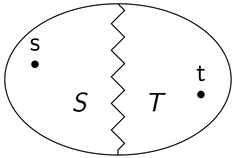
Proof.

Maybe later. Need to prove capacity constraints are satisfied and flow conservation holds. □

Cut, flow across and capacity of

Definition

A cut is a partition of V into subsets $S \ni s$ and $T \ni t$.



Definition

Given a flow f and a cut (S, T) we define the net flow across (S, T) as

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

Definition

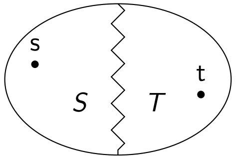
Given a cut (S, T) we define the capacity of (S, T) as

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

Cut, flow across and capacity of

Definition

A cut is a partition of V into subsets $S \ni s$ and $T \ni t$.



Definition

Given a flow f and a cut (S, T) we define the net flow across (S, T) as

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

Definition

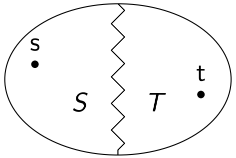
Given a cut (S, T) we define the capacity of (S, T) as

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

Cut, flow across and capacity of

Definition

A cut is a partition of V into subsets $S \ni s$ and $T \ni t$.



Definition

Given a flow f and a cut (S, T) we define the net flow across (S, T) as

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

Definition

Given a cut (S, T) we define the capacity of (S, T) as

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

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

□

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

□

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

□

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

$\square$

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

□

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S, T) we have $f(S, T) = |f|$.

Proof.

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

□

Lemma 2 (net flow value)

Lemma

Given a flow f in G, for all cuts (S,T) we have $f(S,T)=|f|$.

Proof.

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

□

Corollary (flow value upper bounded by cut capacity)

Corollary

For any flow f and any cut (S,T), $|f| \le c(S,T)$.

Proof.

$$
\begin{aligned}
|f| &= f(S,T) && \text{(By Lemma 2)}\\
&= \sum_{u\in S}\sum_{v\in T}(f(u,v) - f(v,u)) && \text{(By definition of } f(S,T))\\
&\le \sum_{u\in S}\sum_{v\in T}(c(u,v) - 0) && \left(\begin{array}{l}\text{Since } f(u,v) \le c(u,v)\\ \text{and } -f(v,u) \le 0 \text{ by the}\\ \text{capacity constraints}\end{array}\right)\\
&= c(S,T) && \square
\end{aligned}
$$

Illustrate on number line, all possible flow values are left of all possible cut capacities.

Max flow/Min cut Theorem says they meet in the middle.

Corollary (flow value upper bounded by cut capacity)

Corollary

For any flow f and any cut (S, T), $|f| \leq c(S, T)$.

Proof.

$$
\begin{aligned}
|f| &= f(S, T) && \text{(By Lemma 2)} \\
&= \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u)) && \text{(By definition of } f(S, T)\text{)} \\
&\leq \sum_{u \in S} \sum_{v \in T} (c(u, v) - 0) && \left(\begin{array}{l}\text{Since } f(u, v) \leq c(u, v) \\ \text{and } -f(v, u) \leq 0 \text{ by the} \\ \text{capacity constraints}\end{array}\right) \\
&= c(S, T) && \square
\end{aligned}
$$

Illustrate on number line, all possible flow values are left of all possible cut capacities.

Max flow/Min cut Theorem says they meet in the middle.

Corollary (flow value upper bounded by cut capacity)

Corollary

For any flow f and any cut (S, T), $|f| \leq c(S, T)$.

Proof.

$$
\begin{aligned}
|f| &= f(S, T) && \text{(By Lemma 2)} \\
&= \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u)) && \text{(By definition of } f(S, T)\text{)} \\
&\leq \sum_{u \in S} \sum_{v \in T} (c(u, v) - 0) && \left(\begin{array}{l}\text{Since } f(u, v) \leq c(u, v) \\ \text{and } -f(v, u) \leq 0 \text{ by the} \\ \text{capacity constraints}\end{array}\right) \\
&= c(S, T) && \square
\end{aligned}
$$

Illustrate on number line, all possible flow values are left of all possible cut capacities.

Max flow/Min cut Theorem says they meet in the middle.

Corollary (flow value upper bounded by cut capacity)

Corollary

For any flow f and any cut (S, T), $|f| \leq c(S, T)$.

Proof.

$$
\begin{aligned}
|f| &= f(S,T) && \text{(By Lemma 2)}\\
&= \sum_{u \in S} \sum_{v \in T} (f(u,v) - f(v,u)) && \text{(By definition of } f(S,T)\text{)}\\
&\leq \sum_{u \in S} \sum_{v \in T} (c(u,v) - 0) && \left(\begin{array}{l}\text{Since } f(u,v) \leq c(u,v)\\ \text{and } -f(v,u) \leq 0 \text{ by the}\\ \text{capacity constraints}\end{array}\right)\\
&= c(S,T) && \square
\end{aligned}
$$

Illustrate on number line, all possible flow values are left of all possible cut capacities.

Max flow/Min cut Theorem says they meet in the middle.

Corollary (flow value upper bounded by cut capacity)

Corollary

For any flow f and any cut (S, T), $|f| \leq c(S, T)$.

Proof.

$$
\begin{aligned}
|f| &= f(S, T) && \text{(By Lemma 2)} \\
&= \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u)) && \text{(By definition of } f(S, T)\text{)} \\
&\leq \sum_{u \in S} \sum_{v \in T} (c(u, v) - 0) && \left(\begin{array}{l}\text{Since } f(u, v) \leq c(u, v) \\ \text{and } -f(v, u) \leq 0 \text{ by the} \\ \text{capacity constraints}\end{array}\right) \\
&= c(S, T) && \square
\end{aligned}
$$

Illustrate on number line, all possible flow values are left of all possible cut capacities.

Max flow/Min cut Theorem says they meet in the middle.

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?

Max flow/Min cut Theorem

Theorem (Max flow/Min cut Theorem)

Let f be a flow in (G, s, t, c). Then the following 3 statements are equivalent:

1. *f is a max flow.*
2. *There is no augmenting path (in G_f).*
3. *There exists a cut (S, T) such that $|f| = c(S, T)$.*

Q: What does this say about Ford-Fulkerson?
A: If it terminates, it returns the correct result.

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm

Summary

Todays topic was Max Flow. We have covered

- Definition of flow network, flow, etc
- The Ford-Fulkerson Method
- The Max flow/Min cut Theorem

Next time:

- Proof of Max flow/Min cut Theorem
- Worst case analysis of Ford-Fulkerson
- Edmonds-Karp Algorithm