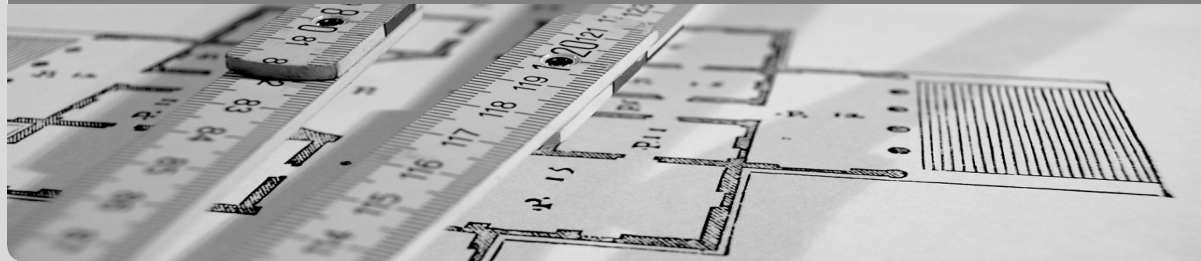


# Graphen 3:

## Maximum Flow, Bipartite Matching

Ford-Fulkerson, Edmond-Karp, Max Flow, Min Cut, MCBM, Bipartite Graphen, Vertex Cover, König  
Peter Koepernik, Robert Brede, Serge Thilges, Jean-Pierre von der Heydt | 12. Juni 2019

BASISPRAKTIKUM ZUM ICPC PROGRAMMIERWETTBEWERB



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Beispielaufgabe

- Gegeben sei ein Netz mit Städten und Straßen mit einer Kapazität

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

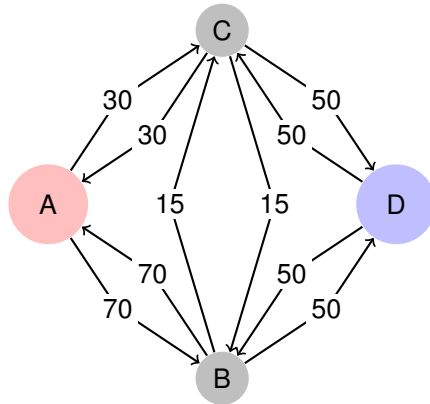
Bipartite Graphen

## Beispielaufgabe

- Gegeben sei ein Netz mit Städten und Straßen mit einer Kapazität
- Für gewisse Städte A und D sucht man die Anzahl Autos die von A nach D fahren können

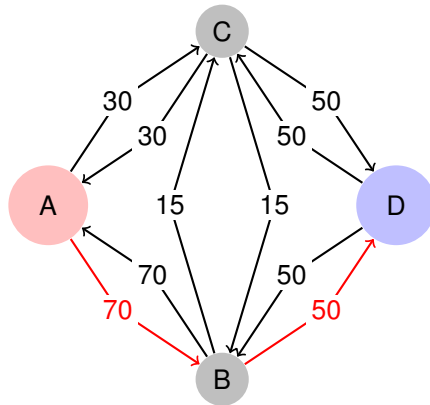
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

### Bipartite Graphen



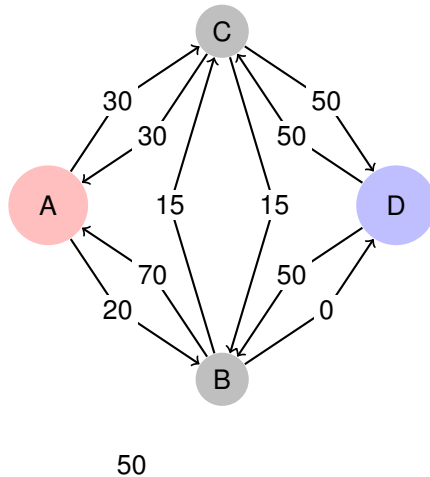
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



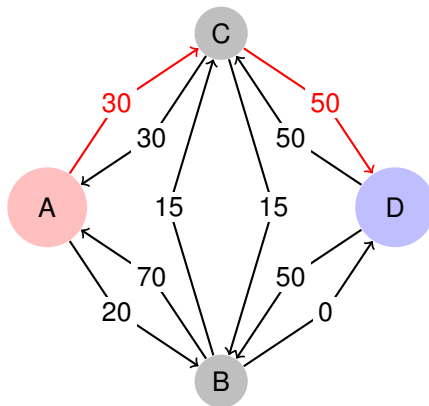
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

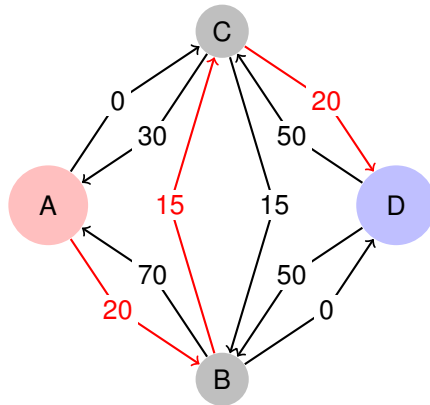
## Bipartite Graphen



50

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen

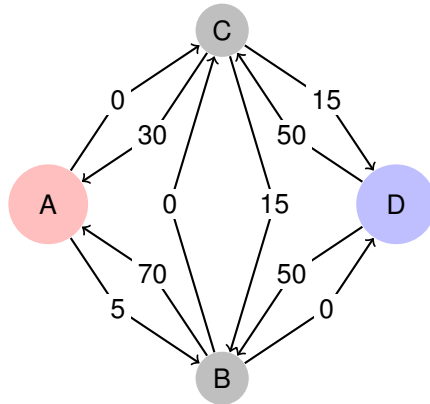


$$50 + 30$$



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



$$50 + 30 + 15 = 95$$

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Ford Fulkerson

- $mf = 0;$

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Ford Fulkerson

- $mf = 0$ ;
- Solange ein steigender Pfad  $p$  ( $s \rightarrow \dots \rightarrow i \rightarrow j \rightarrow \dots \rightarrow t$ ) von source nach  $t$  existiert:

## Ford Fulkerson

- $mf = 0$ ;
- Solange ein steigender Pfad  $p$  ( $s \rightarrow \dots \rightarrow i \rightarrow j \rightarrow \dots \rightarrow t$ ) von source nach  $t$  existiert:
  - 1. finde minimale Kante  $f$  auf dem Pfad

## Ford Fulkerson

- $mf = 0$ ;
- Solange ein steigender Pfad  $p$  ( $s \rightarrow \dots \rightarrow i \rightarrow j \rightarrow \dots \rightarrow t$ ) von source nach  $t$  existiert:
  - 1. finde minimale Kante  $f$  auf dem Pfad
  - 2. Kapazität aller Kanten in Pfadrichtung (z.B.  $i \rightarrow j$ ) um  $f$  reduzieren

## Ford Fulkerson

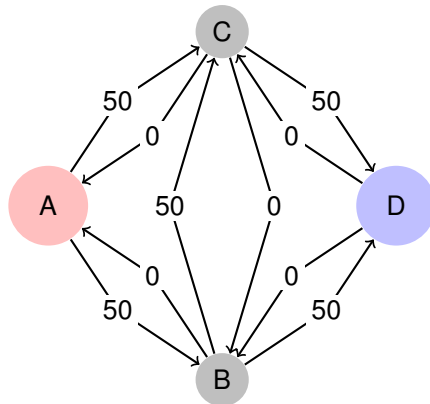
- $mf = 0$ ;
- Solange ein steigender Pfad  $p$  ( $s \rightarrow \dots \rightarrow i \rightarrow j \rightarrow \dots \rightarrow t$ ) von source nach  $t$  existiert:
  - 1. finde minimale Kante  $f$  auf dem Pfad
  - 2. Kapazität aller Kanten in Pfadrichtung (z.B.  $i \rightarrow j$ ) um  $f$  reduzieren
  - 3. Kapazität aller Kanten gegen Pfadrichtung (z.B.  $j \rightarrow i$ ) um  $f$  erhöhen

## Ford Fulkerson

- $mf = 0$ ;
- Solange ein steigender Pfad  $p$  ( $s \rightarrow \dots \rightarrow i \rightarrow j \rightarrow \dots \rightarrow t$ ) von source nach  $t$  existiert:
  - 1. finde minimale Kante  $f$  auf dem Pfad
  - 2. Kapazität aller Kanten in Pfadrichtung (z.B.  $i \rightarrow j$ ) um  $f$  reduzieren
  - 3. Kapazität aller Kanten gegen Pfadrichtung (z.B.  $j \rightarrow i$ ) um  $f$  erhöhen
  - $mf += f$ ;

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

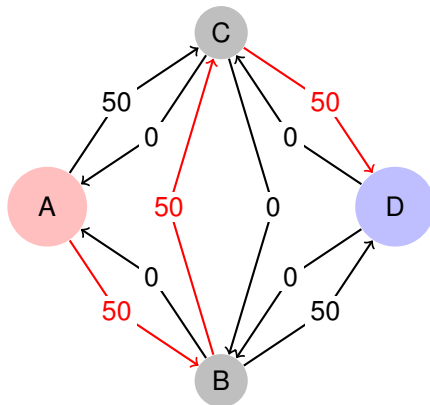
## Bipartite Graphen





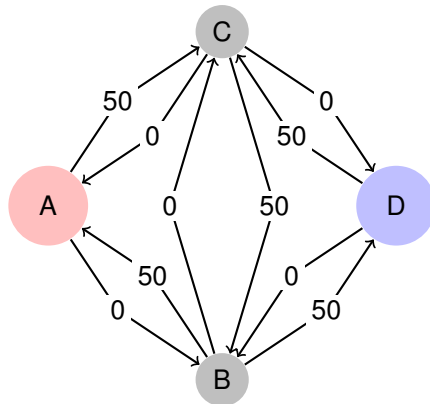
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



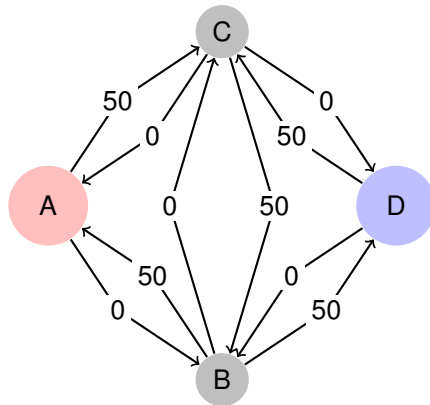
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



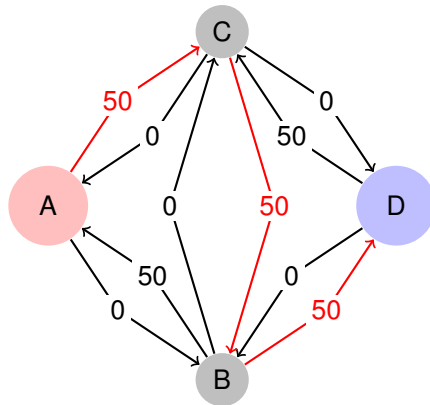
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



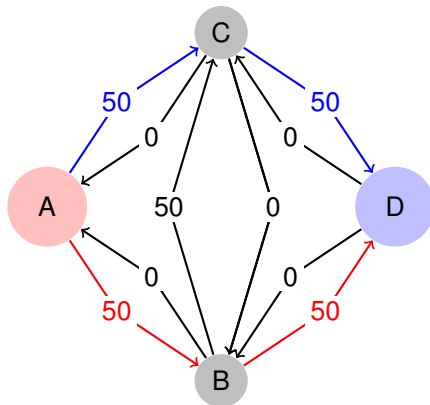
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

### Laufzeit Ford Fulkerson

- $O(ES)$

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

### Laufzeit Ford Fulkerson

- $O(ES)$
- wobei  $S$  die Lösung ist

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

### Laufzeit Ford Fulkerson

- $O(ES)$
- wobei  $S$  die Lösung ist
- $O(S)$  mal Tiefensuche, was in  $O(E)$  läuft, da  $E \geq V - 1$



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

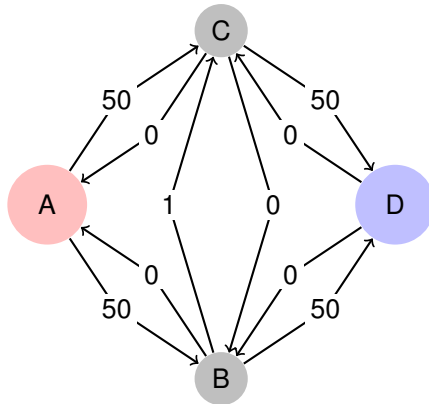
## Laufzeit Ford Fulkerson

- $O(ES)$
- wobei  $S$  die Lösung ist
- $O(S)$  mal Tiefensuche, was in  $O(E)$  läuft, da  $E \geq V - 1$

⇒ kann sehr groß werden

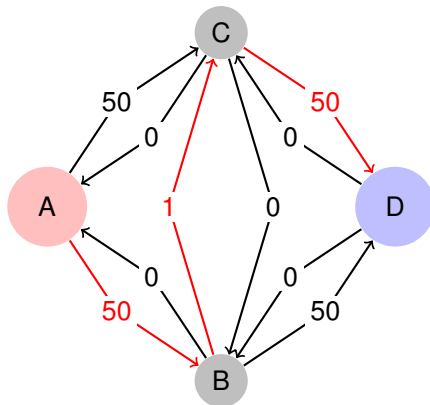
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



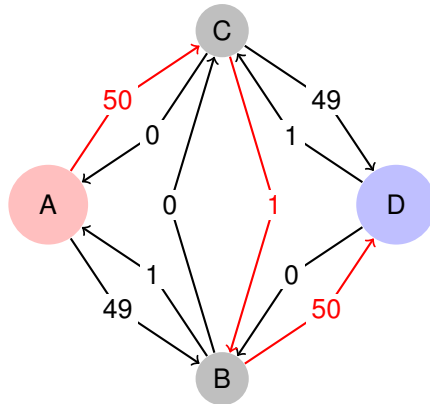
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



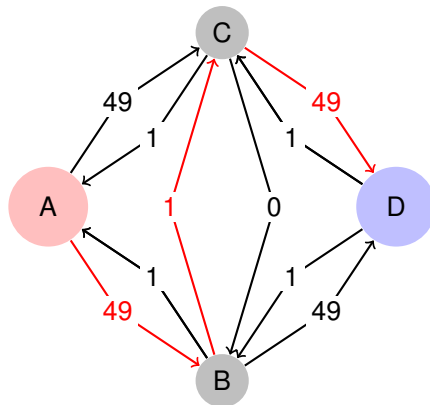
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



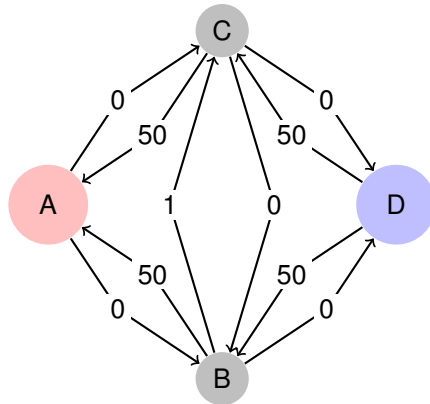
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepf, Robert  
 Brede, Serge Thilges,  
 Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Unterschied zu Ford Fulkerson

- Breitensuche statt Tiefensuche

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

### Unterschied zu Ford Fulkerson

- Breitensuche statt Tiefensuche
- Laufzeit  $O(VE^2)$



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

### Unterschied zu Ford Fulkerson

- Breitensuche statt Tiefensuche
- Laufzeit  $O(VE^2)$
- $O(VE)$  mal Breitensuche, was in  $O(E)$  läuft

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## UVa 10779 - Collector's Problem

- Unterschiedliche Karten zum Sammeln

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## UVa 10779 - Collector's Problem

- Unterschiedliche Karten zum Sammeln
- Alle Karten sind gleich viel Wert, Tausch 1 zu 1

## UVa 10779 - Collector's Problem

- Unterschiedliche Karten zum Sammeln
- Alle Karten sind gleich viel Wert, Tausch 1 zu 1
- Andere Sammler tauschen nur eigene Duplikate gegen Karten, die sie noch nicht besitzen

### UVa 10779 - Collector's Problem

- Unterschiedliche Karten zum Sammeln
- Alle Karten sind gleich viel Wert, Tausch 1 zu 1
- Andere Sammler tauschen nur eigene Duplikate gegen Karten, die sie noch nicht besitzen
- Bob tauscht beliebig (auch Einzelstücke ein und gegen Karten, die er bereits besitzt)

## UVa 10779 - Collector's Problem

- Unterschiedliche Karten zum Sammeln
- Alle Karten sind gleich viel Wert, Tausch 1 zu 1
- Andere Sammler tauschen nur eigene Duplikate gegen Karten, die sie noch nicht besitzen
- Bob tauscht beliebig (auch Einzelstücke ein und gegen Karten, die er bereits besitzt)
- Wie viele unterschiedliche Karten kann Bob maximal besitzen?

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



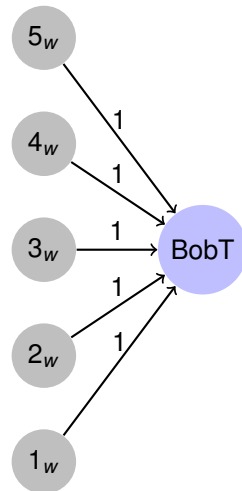
BobS



BobT

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

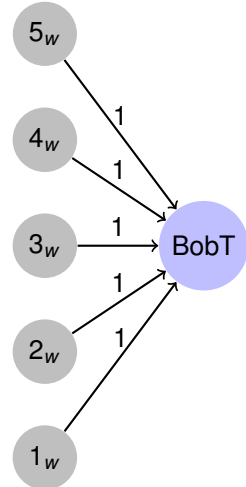
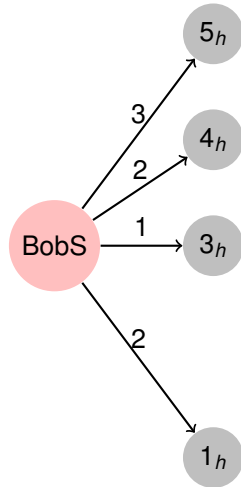
Bipartite Graphen





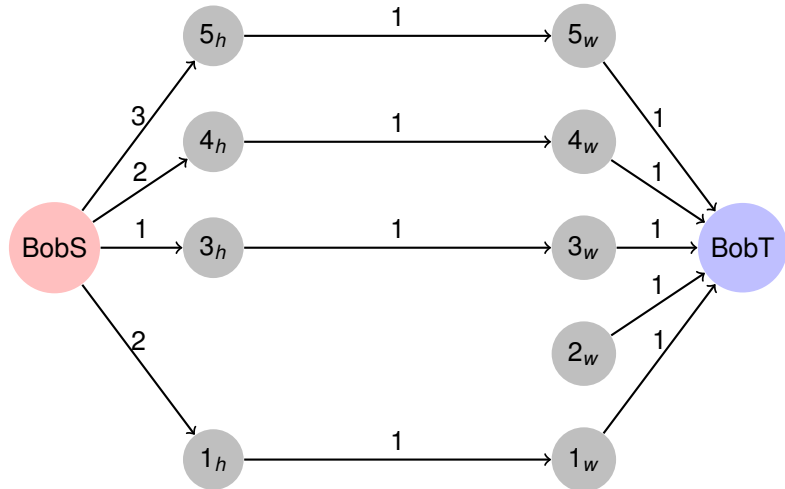
Peter Koepf, Robert  
 Brede, Serge Thilges,  
 Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



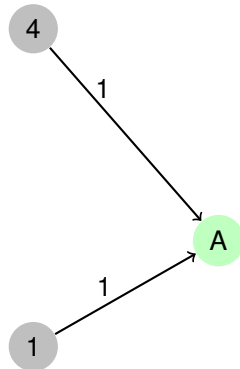
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



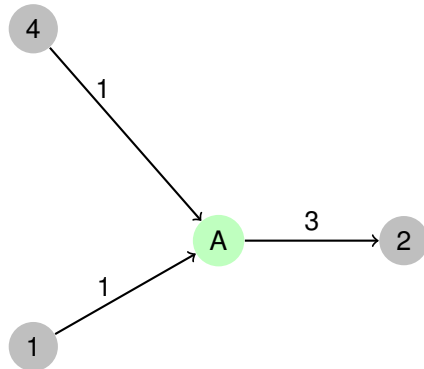
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



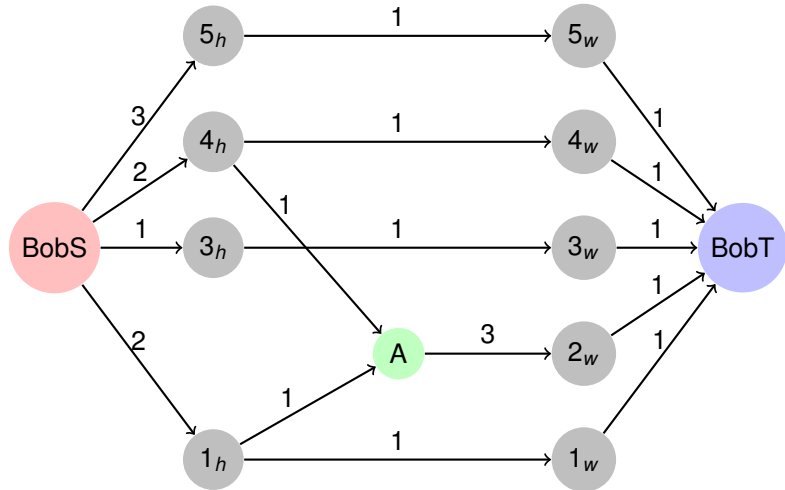
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



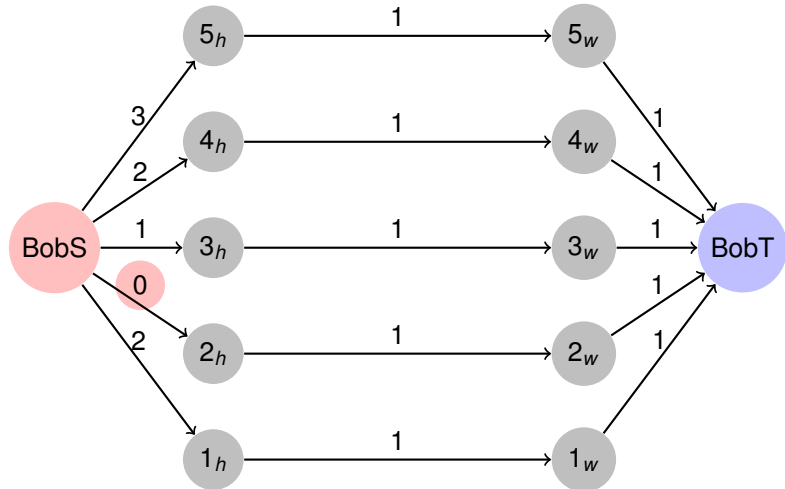
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



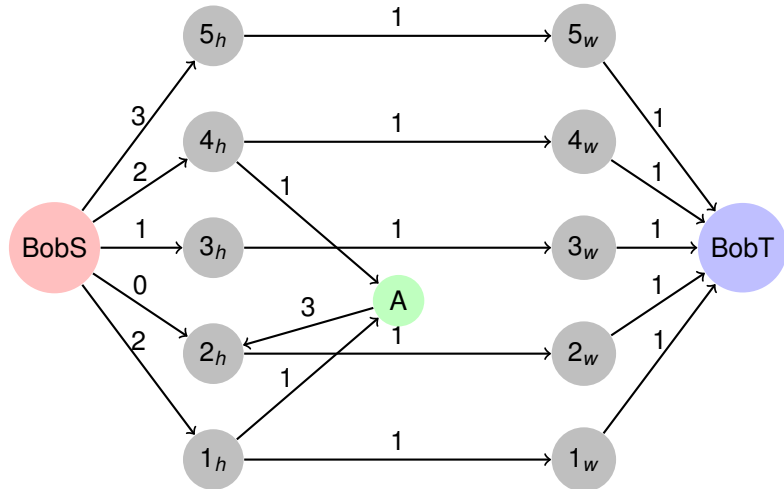
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

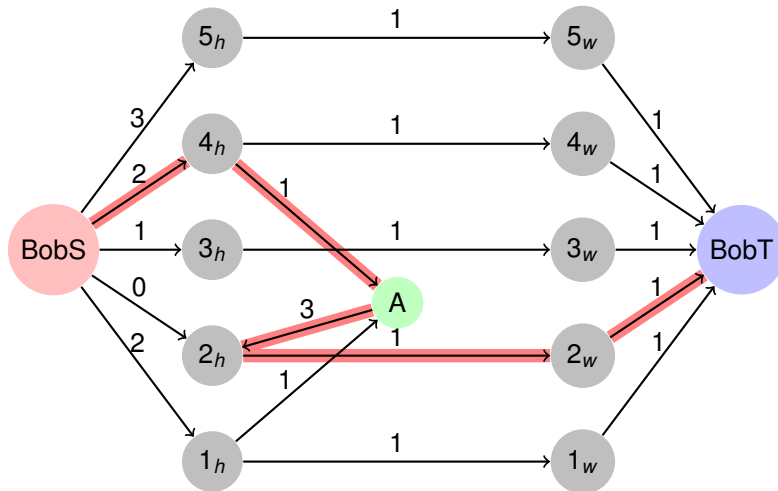
Bipartite Graphen





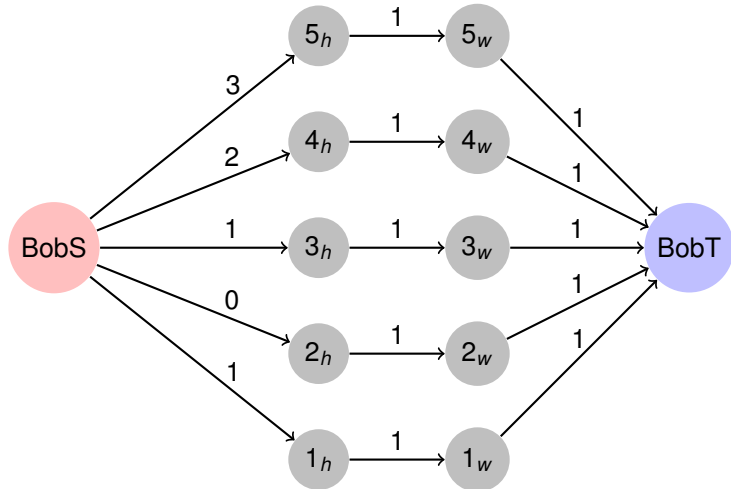
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



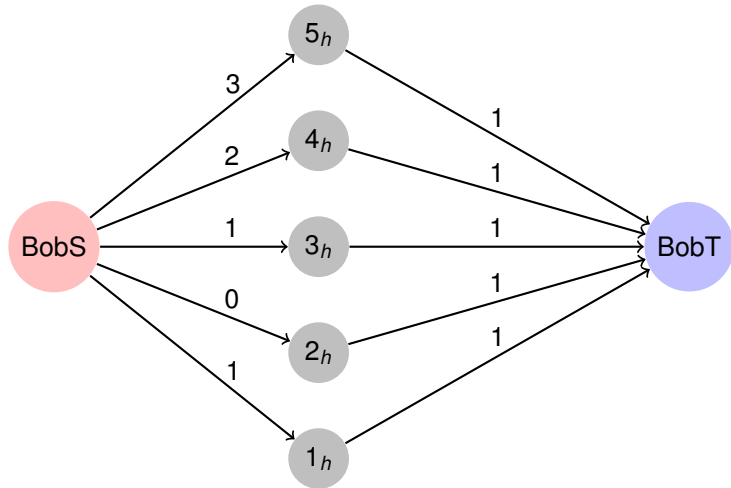
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



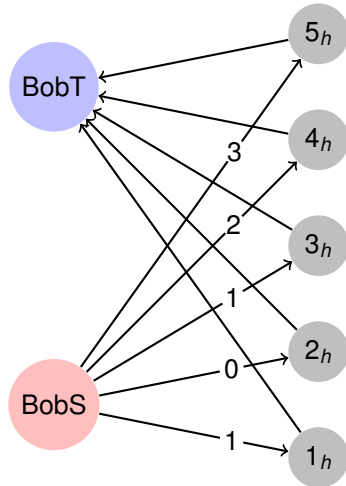
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



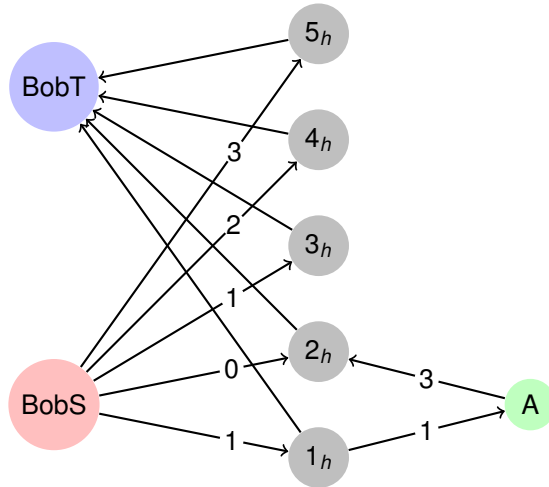
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



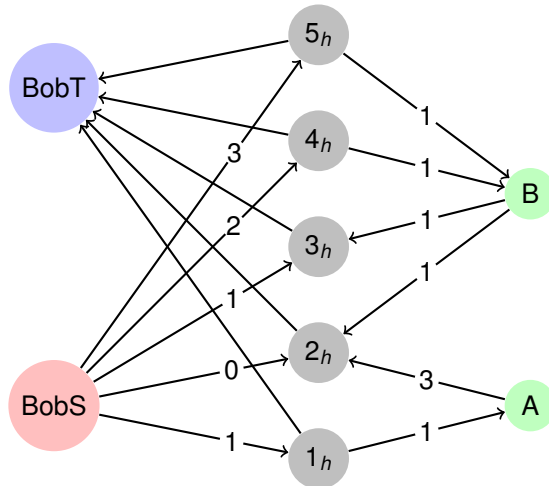
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



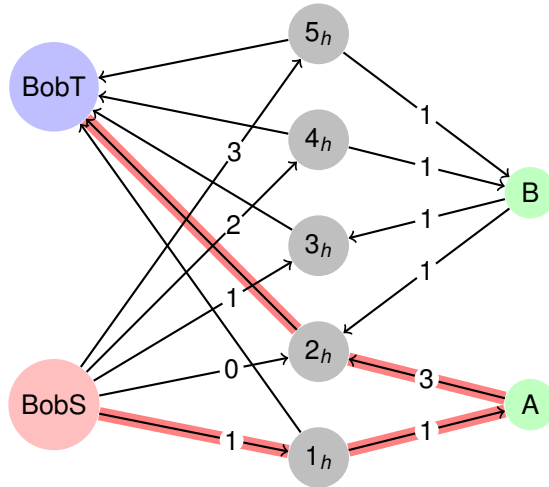
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



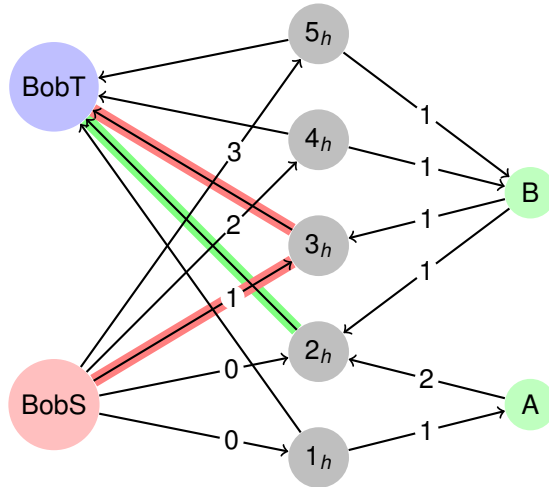
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

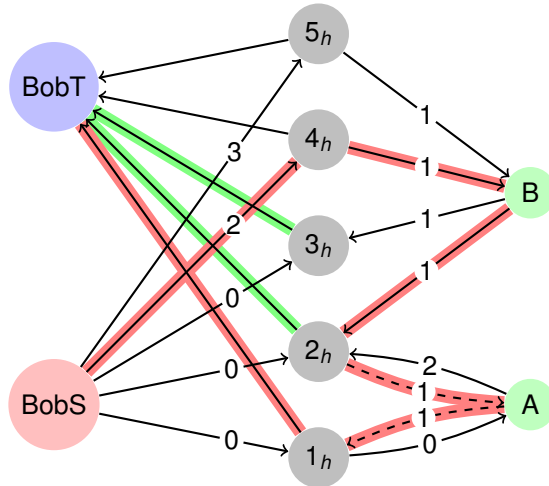
Bipartite Graphen





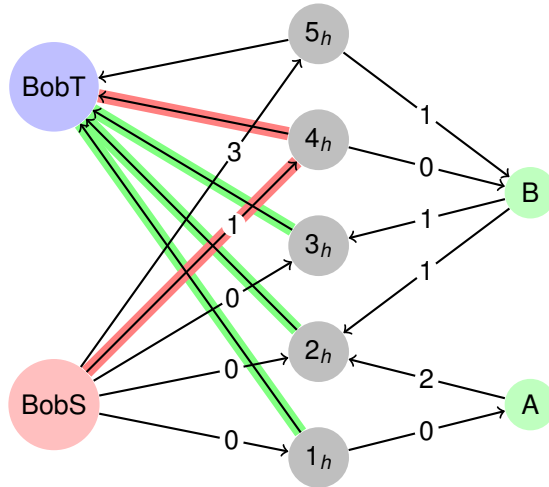
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



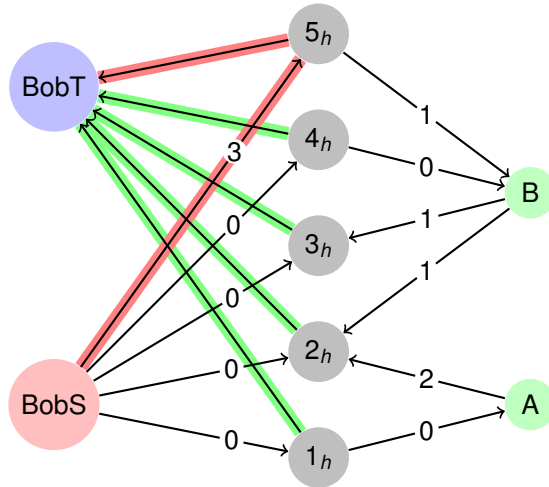
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



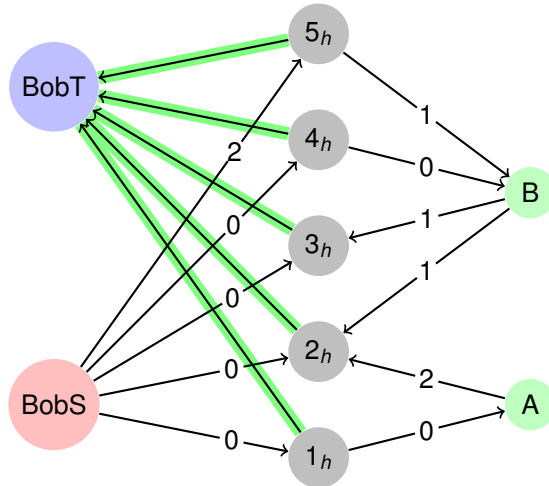
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

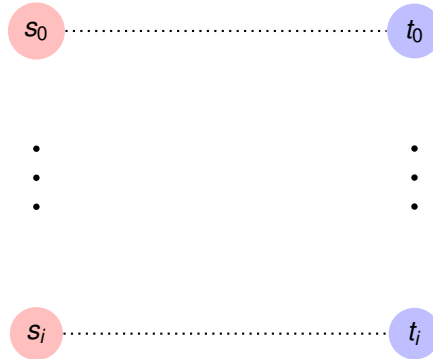


Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

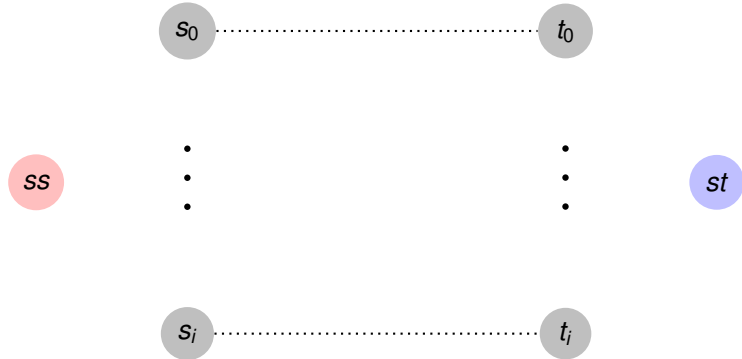
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



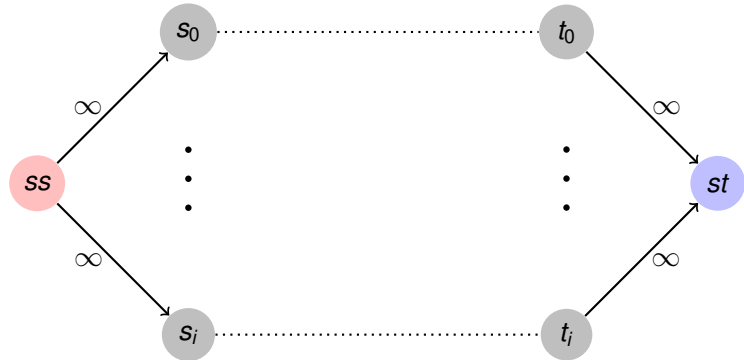
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

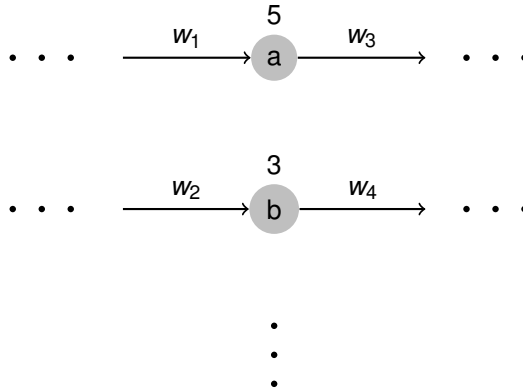
Bipartite Graphen





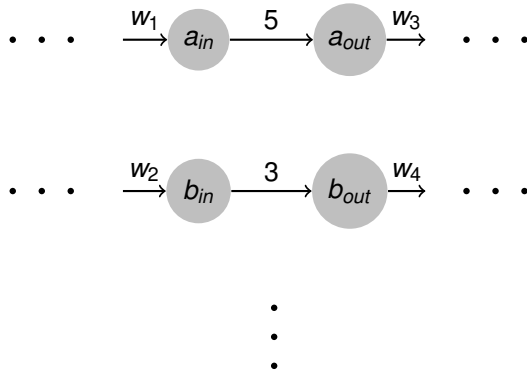
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Definition

## Definition

Ist  $V = S \dot{\cup} T$  eine Partition von  $V$  mit  $s \in S$ ,  $t \in T$ , so heißt  $C := (S, T)$  ein  **$s$ - $t$  cut** (oder  **$s$ - $t$  Schnitt**).

## Definition

Ist  $V = S \dot{\cup} T$  eine Partition von  $V$  mit  $s \in S$ ,  $t \in T$ , so heißt  $C := (S, T)$  ein  **$s$ - $t$  cut** (oder  **$s$ - $t$  Schnitt**).

Das zu  $C$  gehörige **cut-set** ist

$$X_C := \{(u, v) \in E \mid u \in S, v \in T\} = (S \times T) \cap E$$

## Definition

Ist  $V = S \dot{\cup} T$  eine Partition von  $V$  mit  $s \in S$ ,  $t \in T$ , so heißt  $C := (S, T)$  ein  **$s$ - $t$  cut** (oder  **$s$ - $t$  Schnitt**).

Das zu  $C$  gehörige **cut-set** ist

$$X_C := \{(u, v) \in E \mid u \in S, v \in T\} = (S \times T) \cap E$$

Die **Kosten** des Schnittes sind definiert durch  $c(S, T) := \sum_{(u,v) \in X_C} c(u, v)$

## Definition

Ist  $V = S \dot{\cup} T$  eine Partition von  $V$  mit  $s \in S$ ,  $t \in T$ , so heißt  $C := (S, T)$  ein **s-t cut** (oder **s-t Schnitt**).

Das zu  $C$  gehörige **cut-set** ist

$$X_C := \{(u, v) \in E \mid u \in S, v \in T\} = (S \times T) \cap E$$

Die **Kosten** des Schnittes sind definiert durch  $c(S, T) := \sum_{(u,v) \in X_C} c(u, v)$   
Ein **Min Cut** ist ein s-t cut  $C = (S, T)$  mit minimalen Kosten.

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

- Nebenprodukt von Max Flow



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

- Nebenprodukt von Max Flow
- DFS/BFS von  $s$  ausführen (nur Kanten mit streng positiver restlicher Kapazität traversierbar)

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

- Nebenprodukt von Max Flow
- DFS/BFS von  $s$  ausführen (nur Kanten mit streng positiver restlicher Kapazität traversierbar)
- Jeder gefundene Knoten ist in  $S$

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

### Bipartite Graphen

- Nebenprodukt von Max Flow
- DFS/BFS von  $s$  ausführen (nur Kanten mit streng positiver restlicher Kapazität traversierbar)
- Jeder gefundene Knoten ist in  $S$
- $T = V \setminus S$

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

### Bipartite Graphen

- Nebenprodukt von Max Flow
- DFS/BFS von  $s$  ausführen (nur Kanten mit streng positiver restlicher Kapazität traversierbar)
- Jeder gefundene Knoten ist in  $S$
- $T = V \setminus S$
- Alle Kanten in  $X_C$  haben Restkapazität 0  $\implies$  Min Cut = Max Flow

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

### Bipartite Graphen

- Nebenprodukt von Max Flow
- DFS/BFS von  $s$  ausführen (nur Kanten mit streng positiver restlicher Kapazität traversierbar)
- Jeder gefundene Knoten ist in  $S$
- $T = V \setminus S$
- Alle Kanten in  $X_C$  haben Restkapazität 0  $\implies$  Min Cut = Max Flow
- Max-Flow-Min-Cut Theorem

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## UVa 11506 - Angry Programmer

- Gefeuerter Programmierer will sich rächen und Netzwerk zerstören

## UVa 11506 - Angry Programmer

- Gefeuerter Programmierer will sich rächen und Netzwerk zerstören
- Kann Computer und Kabel (verbinden je einen Computer mit einem Anderen) zerstören, jeweils mit bekannten Kosten

## UVa 11506 - Angry Programmer

- Gefeuerter Programmierer will sich rächen und Netzwerk zerstören
- Kann Computer und Kabel (verbinden je einen Computer mit einem Anderen) zerstören, jeweils mit bekannten Kosten
- Computer des Chefs und Server sind unzerstörbar und Verbindung soll getrennt werden



## UVa 11506 - Angry Programmer

- Gefeuerter Programmierer will sich rächen und Netzwerk zerstören
- Kann Computer und Kabel (verbinden je einen Computer mit einem Anderen) zerstören, jeweils mit bekannten Kosten
- Computer des Chefs und Server sind unzerstörbar und Verbindung soll getrennt werden
- Was sind die minimalen Kosten um die Verbindung zu zerstören?

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

UVa 11506 - Angry Programmer - Lösung

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## UVa 11506 - Angry Programmer - Lösung

- Computer sind Knoten, Kabel sind Kanten

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## UVa 11506 - Angry Programmer - Lösung

- Computer sind Knoten, Kabel sind Kanten
- Aufteilen der Knoten mit Gewicht in in- & out-Knoten mit gewichteter Kante

## UVa 11506 - Angry Programmer - Lösung

- Computer sind Knoten, Kabel sind Kanten
- Aufteilen der Knoten mit Gewicht in in- & out-Knoten mit gewichteter Kante
- Min Cut

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

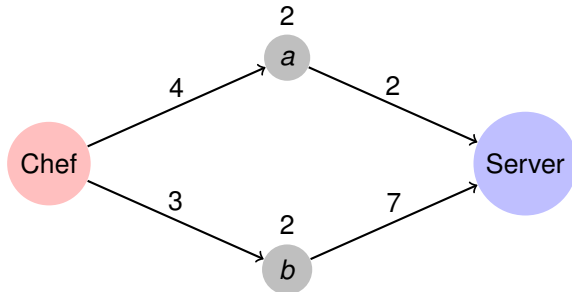


Chef

Server

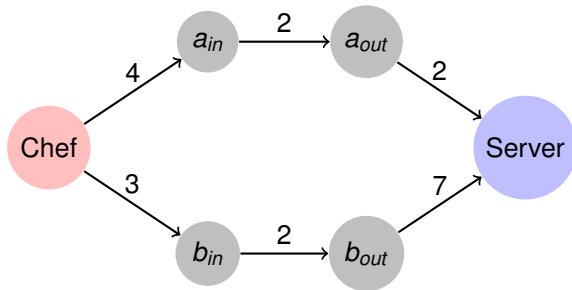
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

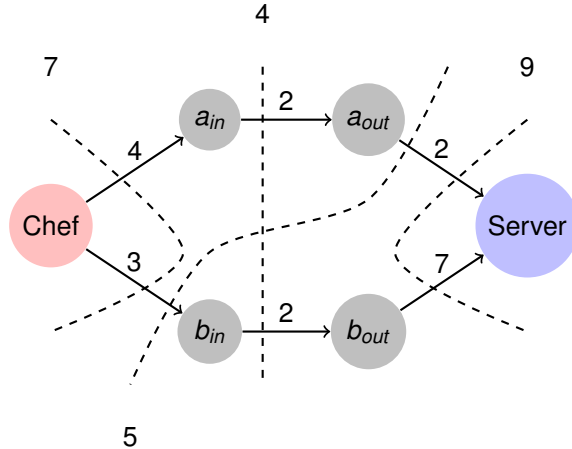
Bipartite Graphen





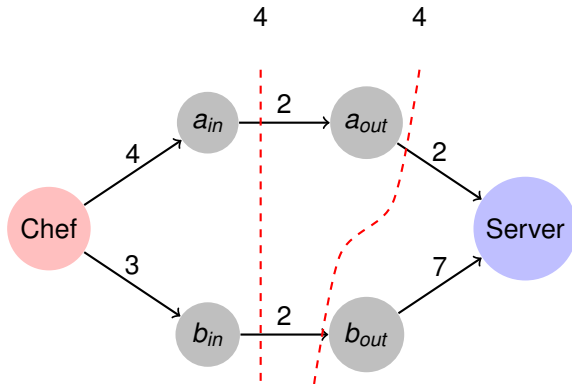
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



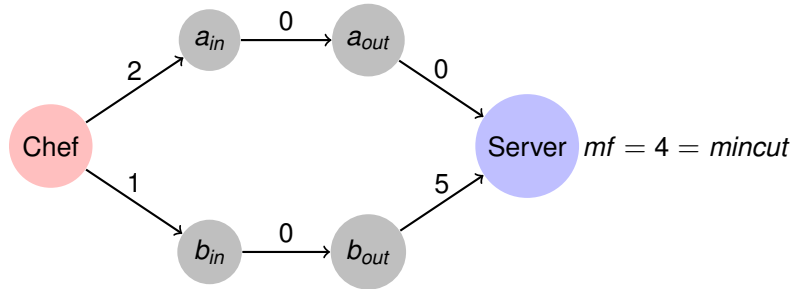
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



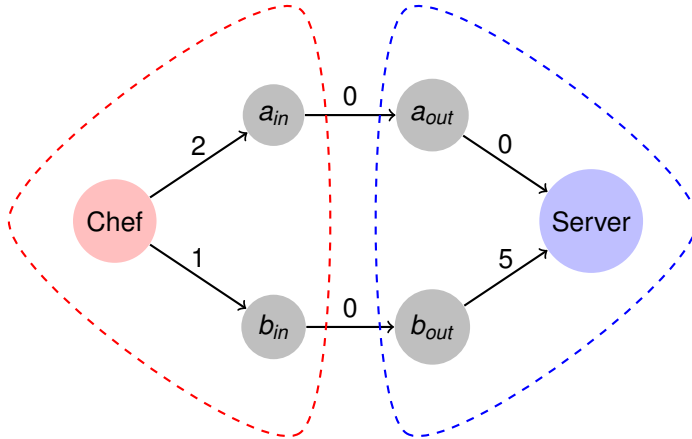
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



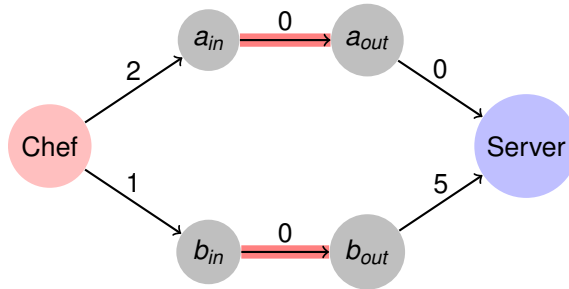
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



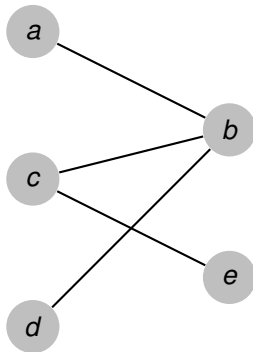
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



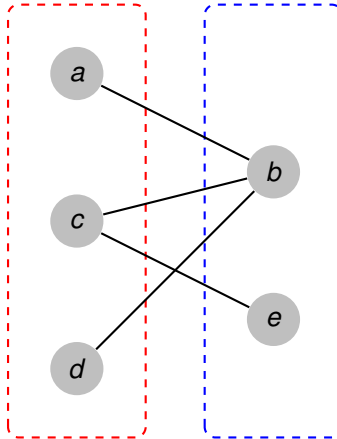
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



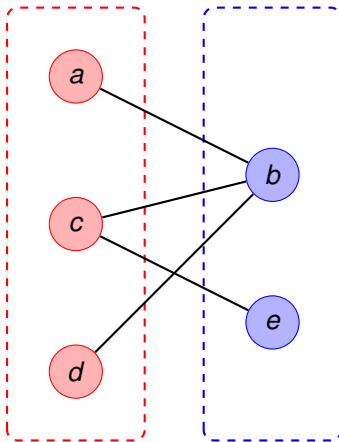
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
 Brede, Serge Thilges,  
 Jean-Pierre von der Heydt

## Bipartite Graphen





### Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.

### Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

$M \in \mathcal{M} := \{M \subseteq E \mid M \text{ ist Matching}\}$  heißt **inklusionsmaximal**

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

$M \in \mathcal{M} := \{M \subseteq E \mid M \text{ ist Matching}\}$  heißt **inklusionsmaximal**, falls

$$\forall M' \in \mathcal{M} : M \subseteq M' \Rightarrow M = M'.$$

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

$M \in \mathcal{M} := \{M \subseteq E \mid M \text{ ist Matching}\}$  heißt **inklusionsmaximal**, falls

$$\forall M' \in \mathcal{M} : M \subseteq M' \Rightarrow M = M'.$$

$M \in \mathcal{M}$  heißt **kardinalitätsmaximal**

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

$M \in \mathcal{M} := \{M \subseteq E \mid M \text{ ist Matching}\}$  heißt **inklusionsmaximal**, falls

$$\forall M' \in \mathcal{M} : M \subseteq M' \Rightarrow M = M'.$$

$M \in \mathcal{M}$  heißt **kardinalitätsmaximal**, falls

$$\forall M' \in \mathcal{M} : |M| \geq |M'|$$

## Definition

Sei  $G = (V, E)$ ,  $E \subseteq \{\{u, v\} \mid u, v \in V\}$  ein ungerichteter Graph.  
Eine Menge von Kanten  $M \subseteq E$  heißt **Matching**, falls

$$\forall e_1, e_2 \in M : e_1 \neq e_2 \Rightarrow e_1 \cap e_2 = \emptyset.$$

$M \in \mathcal{M} := \{M \subseteq E \mid M \text{ ist Matching}\}$  heißt **inklusionsmaximal**, falls

$$\forall M' \in \mathcal{M} : M \subseteq M' \Rightarrow M = M'.$$

$M \in \mathcal{M}$  heißt **kardinalitätsmaximal**, falls

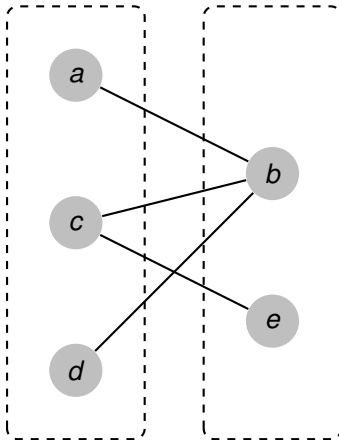
$$\forall M' \in \mathcal{M} : |M| \geq |M'|$$

Für  $G$  bipartit: „Maximum Cardinality Bipartite Matching“, kurz **MCBM**.



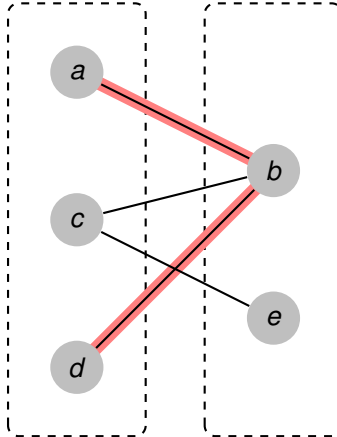
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



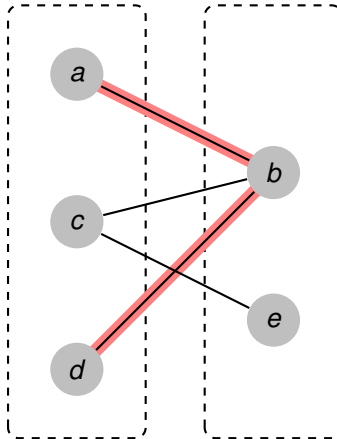
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

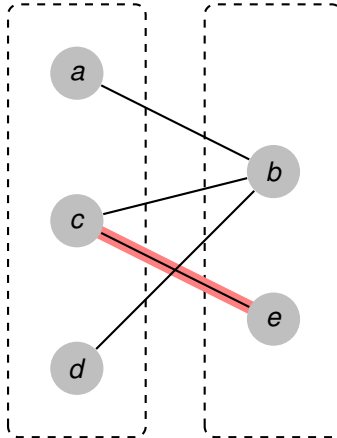
Bipartite Graphen



Kein Matching

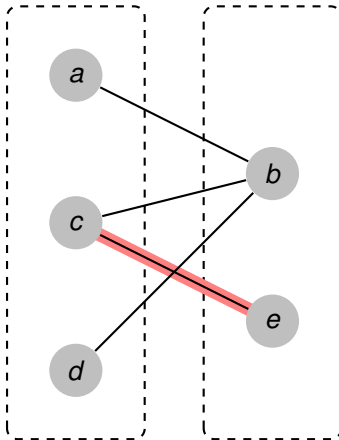
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
 Brede, Serge Thilges,  
 Jean-Pierre von der Heydt

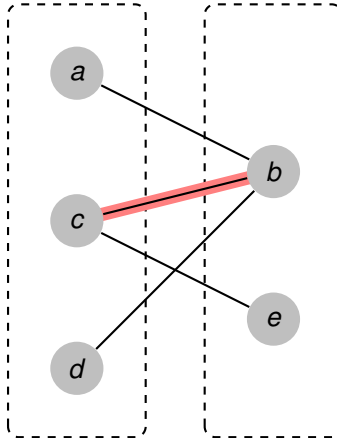
Bipartite Graphen



Matching, aber weder inklusions- noch kardinalitätsmaximal

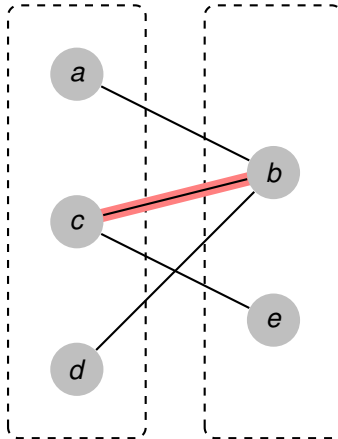
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

## Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

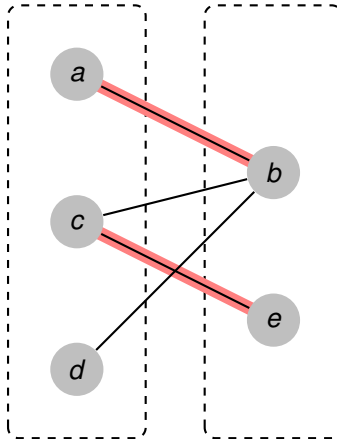
## Bipartite Graphen



Inklusions-, aber nicht kardinalitätsmaximales Matching

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

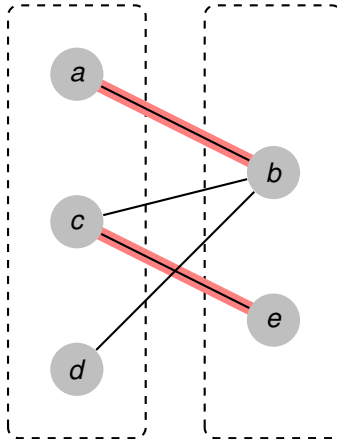
## Bipartite Graphen





Peter Koepf, Robert  
 Brede, Serge Thilges,  
 Jean-Pierre von der Heydt

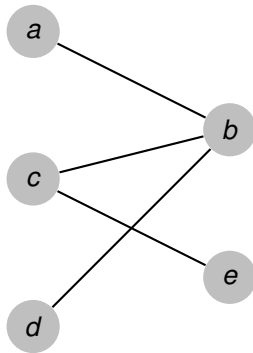
Bipartite Graphen



Kardinalitätsmaximales Matching

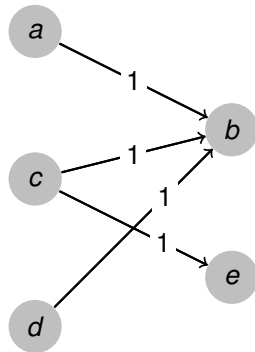
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

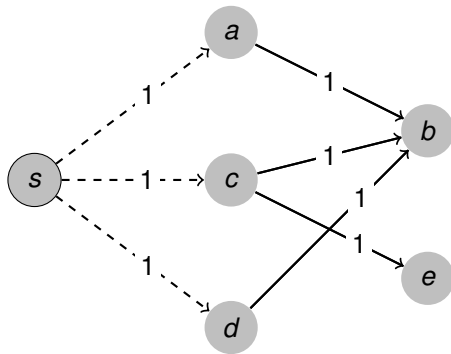
Bipartite Graphen



Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

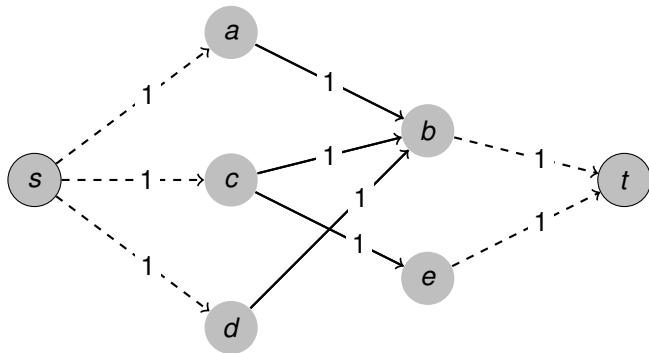
Bipartite Graphen





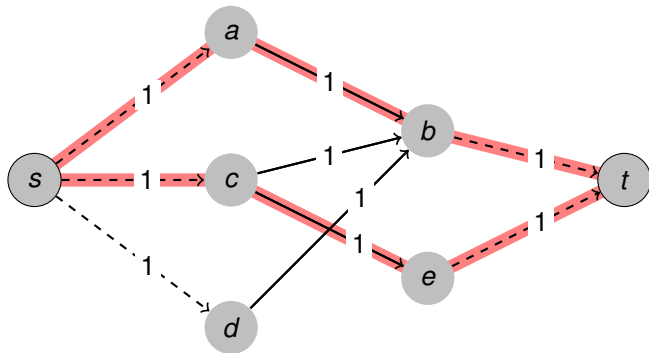
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



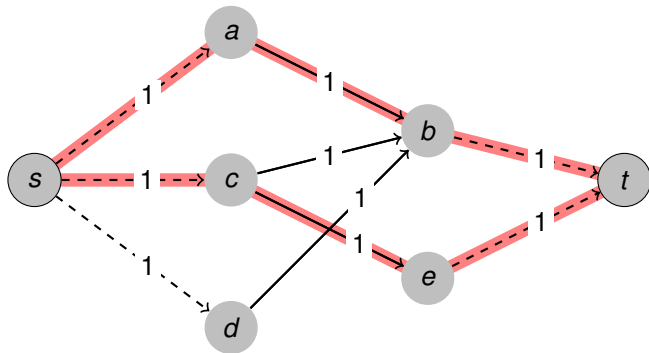
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

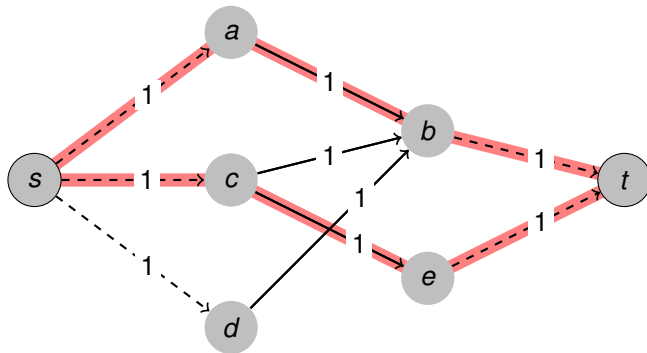
Bipartite Graphen



- Edmond-Karp:  $\mathcal{O}(|V| \cdot |E|^2)$

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

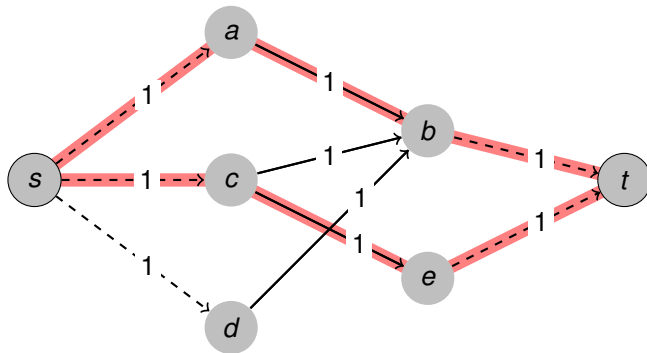


- Edmond-Karp:  $\mathcal{O}(|V| \cdot |E|^2)$
- Ford-Fulkerson:  $\mathcal{O}(f^* \cdot |E|)$



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

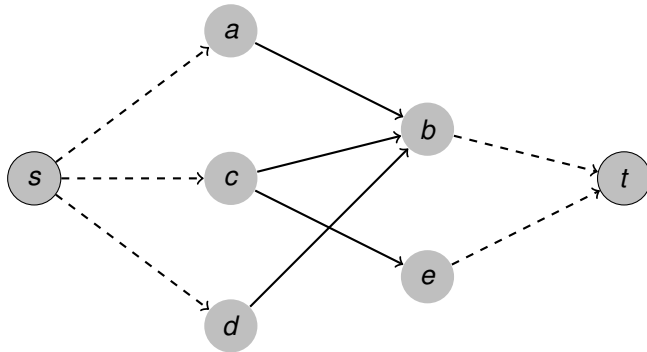
Bipartite Graphen



- Edmond-Karp:  $\mathcal{O}(|V| \cdot |E|^2)$
- Ford-Fulkerson:  $\mathcal{O}(f^* \cdot |E|) = \mathcal{O}(|V| \cdot |E|)$

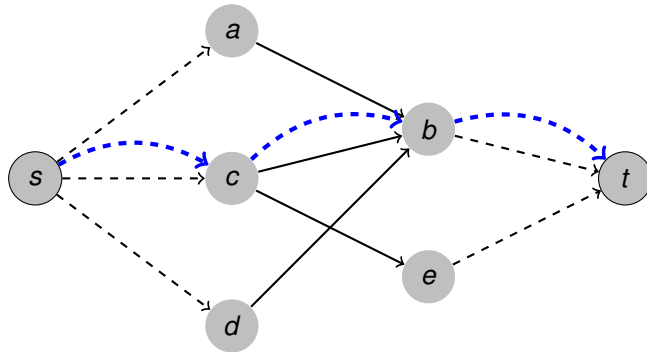
Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



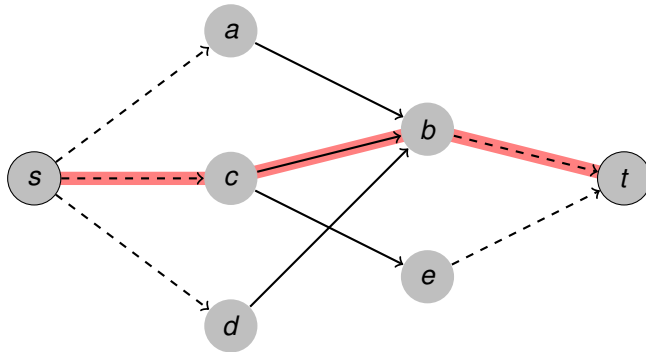
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



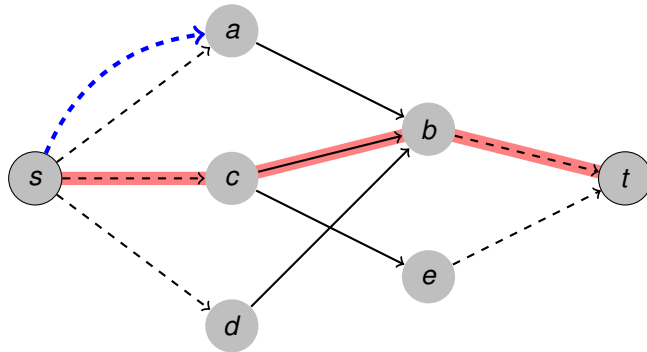
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



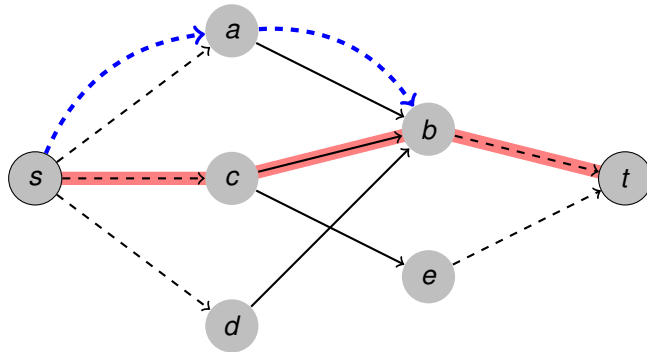
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



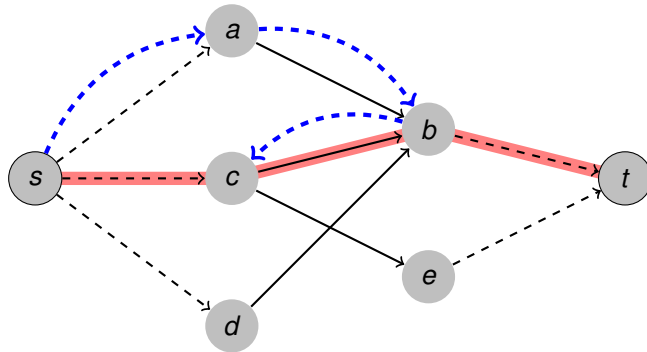
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



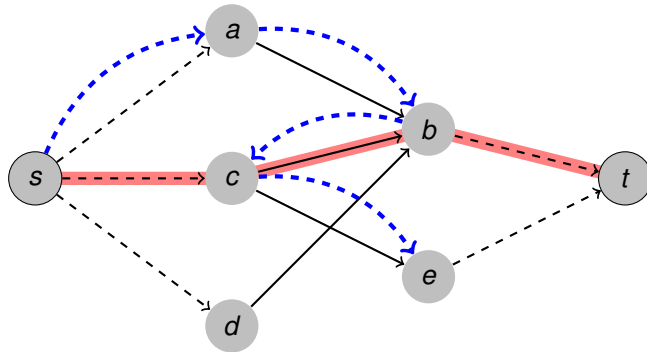
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

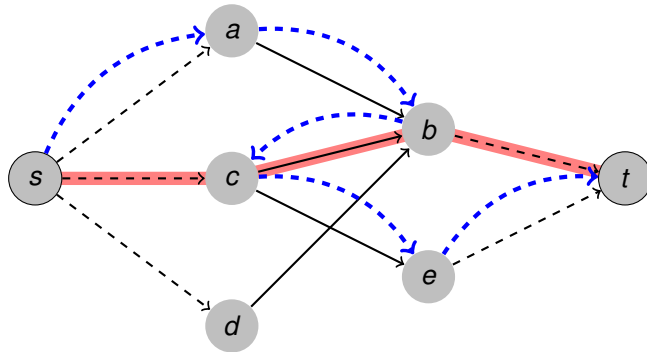
Bipartite Graphen





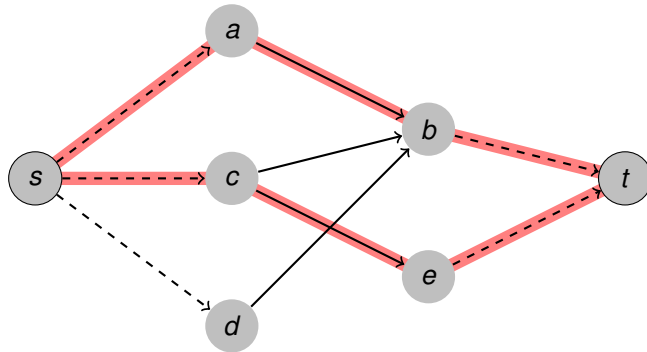
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



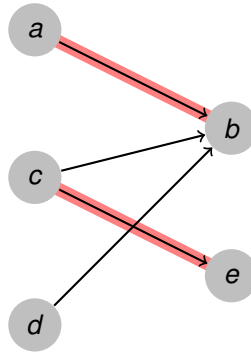
Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen



## Augmenting Paths

Sei  $G = (V, E)$ ,  $V = V_1 \dot{\cup} V_2$  bipartit und  $M \subseteq E$  ein Matching.  
Ein Pfad  $(v_1, \dots, v_n)$  in  $G$  heißt **Augmenting Path** (in  $G$  bzgl.  $M$ )

## Augmenting Paths

Sei  $G = (V, E)$ ,  $V = V_1 \dot{\cup} V_2$  bipartit und  $M \subseteq E$  ein Matching.

Ein Pfad  $(v_1, \dots, v_n)$  in  $G$  heißt **Augmenting Path** (in  $G$  bzgl.  $M$ ), falls

- $v_1 \in V_1 \setminus \bigcup M$  (freier Knoten links)

## Augmenting Paths

Sei  $G = (V, E)$ ,  $V = V_1 \dot{\cup} V_2$  bipartit und  $M \subseteq E$  ein Matching.

Ein Pfad  $(v_1, \dots, v_n)$  in  $G$  heißt **Augmenting Path** (in  $G$  bzgl.  $M$ ), falls

- $v_1 \in V_1 \setminus \bigcup M$  (freier Knoten links)
- $v_n \in V_2 \setminus \bigcup M$  (freier Knoten rechts)

## Augmenting Paths

Sei  $G = (V, E)$ ,  $V = V_1 \dot{\cup} V_2$  bipartit und  $M \subseteq E$  ein Matching.

Ein Pfad  $(v_1, \dots, v_n)$  in  $G$  heißt **Augmenting Path** (in  $G$  bzgl.  $M$ ), falls

- $v_1 \in V_1 \setminus \bigcup M$  (freier Knoten links)
- $v_n \in V_2 \setminus \bigcup M$  (freier Knoten rechts)
- $\{v_i, v_{i+1}\}$  ist abwechselnd  $\in E \setminus M$  (frei) und  $\in M$  (gematcht)

Peter Koepernik, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Augmenting Path Algorithmus

Gegeben: bipartiter Graph  $G = (V, E)$ .



## Augmenting Path Algorithmus

Gegeben: bipartiter Graph  $G = (V, E)$ .

(1) Initialisiere  $M := \emptyset$ .

## Augmenting Path Algorithmus

Gegeben: bipartiter Graph  $G = (V, E)$ .

- (1) Initialisiere  $M := \emptyset$ .
- (2) Suche einen Augmenting Path. Gebe  $M$  aus, falls keinen gefunden.

## Augmenting Path Algorithmus

Gegeben: bipartiter Graph  $G = (V, E)$ .

- (1) Initialisiere  $M := \emptyset$ .
- (2) Suche einen Augmenting Path. Gebe  $M$  aus, falls keinen gefunden.
- (3) Flippe die Kanten entlang des gefundenen Pfades. Gehe zu (2).

### Augmenting Path Algorithmus

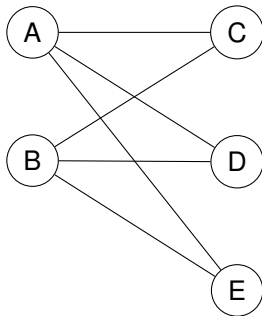
Gegeben: bipartiter Graph  $G = (V, E)$ .

- (1) Initialisiere  $M := \emptyset$ .
- (2) Suche einen Augmenting Path. Gebe  $M$  aus, falls keinen gefunden.
- (3) Flippe die Kanten entlang des gefundenen Pfades. Gehe zu (2).

Findet MCBM in Laufzeit  $\mathcal{O}(|V| \cdot |E|)$ .

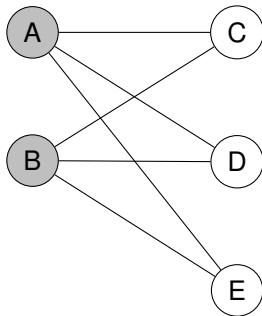
## Definition

Gegeben einen Graphen  $G$ . Ein Independent Set  $IS$  ist eine Menge von Knoten, sodass keine zwei Knoten in  $IS$  über eine Kante in  $G$  verbunden sind.



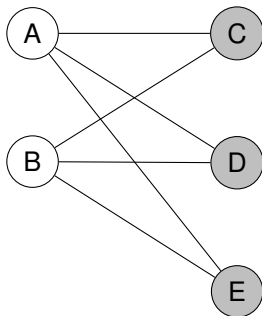
## Definition

Gegeben einen Graphen  $G$ . Ein Independent Set  $IS$  ist eine Menge von Knoten, sodass keine zwei Knoten in  $IS$  über eine Kante in  $G$  verbunden sind.



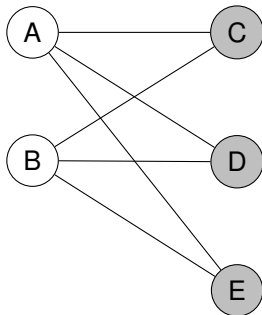
## Definition

Gegeben einen Graphen  $G$ . Ein Independent Set  $IS$  ist eine Menge von Knoten, sodass keine zwei Knoten in  $IS$  über eine Kante in  $G$  verbunden sind.



## Definition

Gegeben einen Graphen  $G$ . Ein Independent Set  $IS$  ist eine Menge von Knoten, sodass keine zwei Knoten in  $IS$  über eine Kante in  $G$  verbunden sind.

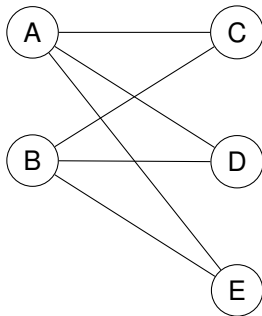


In der Regel wird nach einem möglichst großen Independent Set gesucht.



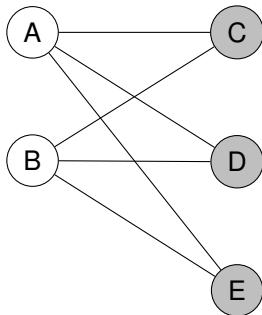
## Definition

Gegeben einen Graphen  $G$ . Ein Vertex Cover  $VC$  ist eine Menge von Knoten, sodass jede Kante in  $G$  mit mindestens einem Knoten aus  $VC$  verbunden ist.



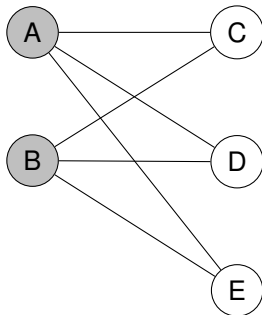
## Definition

Gegeben einen Graphen  $G$ . Ein Vertex Cover  $VC$  ist eine Menge von Knoten, sodass jede Kante in  $G$  mit mindestens einem Knoten aus  $VC$  verbunden ist.



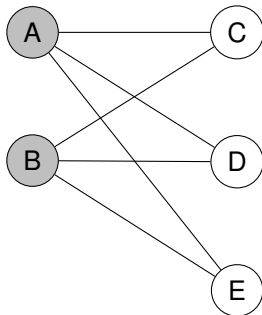
## Definition

Gegeben einen Graphen  $G$ . Ein Vertex Cover  $VC$  ist eine Menge von Knoten, sodass jede Kante in  $G$  mit mindestens einem Knoten aus  $VC$  verbunden ist.



## Definition

Gegeben einen Graphen  $G$ . Ein Vertex Cover  $VC$  ist eine Menge von Knoten, sodass jede Kante in  $G$  mit mindestens einem Knoten aus  $VC$  verbunden ist.



In der Regel wird nach einem möglichst kleinen Vertex Cover gesucht.

## Satz

Sei  $G = (V, E)$  ein Graph und  $X \subseteq V$  eine Menge von Knoten. Dann gilt:

$$X \text{ ist ein VC von } G \iff V \setminus X \text{ ist ein IS von } G$$

## Satz

Sei  $G = (V, E)$  ein Graph und  $X \subseteq V$  eine Menge von Knoten. Dann gilt:

$$X \text{ ist ein VC von } G \iff V \setminus X \text{ ist ein IS von } G$$

## Beweis:

- Sei  $X$  ein beliebiges VC. Wir behaupten, dass  $V \setminus X$  ein IS ist.
- Nehmen wir also das Gegenteil an und führen dies zum Widerspruch:

## Satz

Sei  $G = (V, E)$  ein Graph und  $X \subseteq V$  eine Menge von Knoten. Dann gilt:

$$X \text{ ist ein VC von } G \iff V \setminus X \text{ ist ein IS von } G$$

## Beweis:

- Sei  $X$  ein beliebiges VC. Wir behaupten, dass  $V \setminus X$  ein IS ist.
- Nehmen wir also das Gegenteil an und führen dies zum Widerspruch:
  - Angenommen es würde  $\{u, v\} \subseteq V \setminus X, u \neq v$  existieren mit  $(u, v) \in E$
  - Dann wäre aber  $u, v \notin X$  und die Kante  $(u, v)$  wäre vom VC  $X$  nicht abgedeckt  $\Rightarrow$  Widerspruch!

## Satz

Sei  $G = (V, E)$  ein Graph und  $X \subseteq V$  eine Menge von Knoten. Dann gilt:

$$X \text{ ist ein VC von } G \iff V \setminus X \text{ ist ein IS von } G$$

## Beweis:

- Sei  $X$  ein beliebiges VC. Wir behaupten, dass  $V \setminus X$  ein IS ist.
- Nehmen wir also das Gegenteil an und führen dies zum Widerspruch:
  - Angenommen es würde  $\{u, v\} \subseteq V \setminus X$ ,  $u \neq v$  existieren mit  $(u, v) \in E$
  - Dann wäre aber  $u, v \notin X$  und die Kante  $(u, v)$  wäre vom VC  $X$  nicht abgedeckt  $\Rightarrow$  Widerspruch!
- Die andere Richtung folgt ähnlich



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

Es ist trivial beliebige IS oder VC anzugeben. Deswegen suchen wir in der Regel nach möglichst großen IS und möglichst kleinen VC. Das wollen wir formalisieren:

Es ist trivial beliebige IS oder VC anzugeben. Deswegen suchen wir in der Regel nach möglichst großen IS und möglichst kleinen VC. Das wollen wir formalisieren:

### Definition

Ein IS/VC ist **inklusions maximal/minimal**, wenn kein Knoten hinzugefügt/entfernt werden kann ohne die Eigenschaft des IS/VC zu behalten.

Es ist trivial beliebige IS oder VC anzugeben. Deswegen suchen wir in der Regel nach möglichst großen IS und möglichst kleinen VC. Das wollen wir formalisieren:

### Definition

Ein IS/VC ist **inklusions maximal/minimal**, wenn kein Knoten hinzugefügt/entfernt werden kann ohne die Eigenschaft des IS/VC zu behalten. Ein IS/VC ist **kardinalitäts maximal/minimal**, wenn kein größeres/kleineres IS/VC existiert.

Es ist trivial beliebige IS oder VC anzugeben. Deswegen suchen wir in der Regel nach möglichst großen IS und möglichst kleinen VC. Das wollen wir formalisieren:

### Definition

Ein IS/VC ist **inklusions maximal/minimal**, wenn kein Knoten hinzugefügt/entfernt werden kann ohne die Eigenschaft des IS/VC zu behalten. Ein IS/VC ist **kardinalitäts maximal/minimal**, wenn kein größeres/kleineres IS/VC existiert.

### Bemerkung

Ein kardinalitätsmaximales IS oder ein kardinalitätsminimales VC auszurechnen ist *NP*-schwer.

### Satz (von Dénes König)

In einem bipartiten Graphen ist die Größe eines kardinalitätsminimalem Vertex Cover (VC) gleich der Größe eines Max Cardinality Bipartite Matching (MCBM).

### Satz (von Dénes König)

In einem bipartiten Graphen ist die Größe eines kardinalitätsminimalem Vertex Cover (VC) gleich der Größe eines Max Cardinality Bipartite Matching (MCBM).

Etwas informeller aufgeschrieben erhalten wir damit  $|VC| = |MCBM|$ .

Und mit unserem Wissen aus dem vorangegangenen Satz folgt:

$$|V| = |VC| + |IS| = |MCBM| + |IS|$$

### Satz (von Dénes König)

In einem bipartiten Graphen ist die Größe eines kardinalitätsminimalem Vertex Cover (VC) gleich der Größe eines Max Cardinality Bipartite Matching (MCBM).

Etwas informeller aufgeschrieben erhalten wir damit  $|VC| = |MCBM|$ .

Und mit unserem Wissen aus dem vorangegangenen Satz folgt:

$$|V| = |VC| + |IS| = |MCBM| + |IS|$$

Mit diesem Satz und den uns bekannten Verfahren erhalten wir nur die Größen der Mengen nicht aber deren Elemente. Um auch an die Elemente der Mengen ran zu kommen, braucht es noch mehr Verfahren. Die Mengen sind allerdings nicht eindeutig.

Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Aufgabe

Gegeben sind  $N \leq 500$  Schüler, beschrieben durch Größe, Geschlecht und Musikgeschmack. Der Lehrer möchte wissen wie viele Schüler maximal auf Klassenfahrt kommen können, ohne dass die Gefahr besteht, dass zwei Schüler ein Paar werden. Zwei Schüler laufen Gefahr ein Paar zu werden, wenn sie ein unterschiedliches Geschlecht, maximal 40cm Größendifferenz und einen gleichen Musikgeschmack haben.



Peter Koepf, Robert  
Brede, Serge Thilges,  
Jean-Pierre von der Heydt

Bipartite Graphen

## Aufgabe

Gegeben sind  $N \leq 500$  Schüler, beschrieben durch Größe, Geschlecht und Musikgeschmack. Der Lehrer möchte wissen wie viele Schüler maximal auf Klassenfahrt kommen können, ohne dass die Gefahr besteht, dass zwei Schüler ein Paar werden. Zwei Schüler laufen Gefahr ein Paar zu werden, wenn sie ein unterschiedliches Geschlecht, maximal 40cm Größendifferenz und einen gleichen Musikgeschmack haben.

## Lösungsansatz:

- Modelliere das Problem als Graphen mit den Schülern als Knoten
- Verbinde Schüler, wenn sie ein Paar werden könnten

## Aufgabe

Gegeben sind  $N \leq 500$  Schüler, beschrieben durch Größe, Geschlecht und Musikgeschmack. Der Lehrer möchte wissen wie viele Schüler maximal auf Klassenfahrt kommen können, ohne dass die Gefahr besteht, dass zwei Schüler ein Paar werden. Zwei Schüler laufen Gefahr ein Paar zu werden, wenn sie ein unterschiedliches Geschlecht, maximal 40cm Größendifferenz und einen gleichen Musikgeschmack haben.

## Lösungsansatz:

- Modelliere das Problem als Graphen mit den Schülern als Knoten
- Verbinde Schüler, wenn sie ein Paar werden könnten
- Suche nach einem maximalem IS

## Aufgabe

Gegeben sind  $N \leq 500$  Schüler, beschrieben durch Größe, Geschlecht und Musikgeschmack. Der Lehrer möchte wissen wie viele Schüler maximal auf Klassenfahrt kommen können, ohne dass die Gefahr besteht, dass zwei Schüler ein Paar werden. Zwei Schüler laufen Gefahr ein Paar zu werden, wenn sie ein unterschiedliches Geschlecht, maximal 40cm Größendifferenz und einen gleichen Musikgeschmack haben.

## Lösungsansatz:

- Modelliere das Problem als Graphen mit den Schülern als Knoten
- Verbinde Schüler, wenn sie ein Paar werden könnten
- Suche nach einem maximalem IS
- nutze dafür aus dass der Graph bipartit ist, indem Männchen und Weibchen voneinander getrennt werden
- Berechne mittels Flow ein MCBM und daraus die Größe von IS