



Technische  
Universität  
Braunschweig

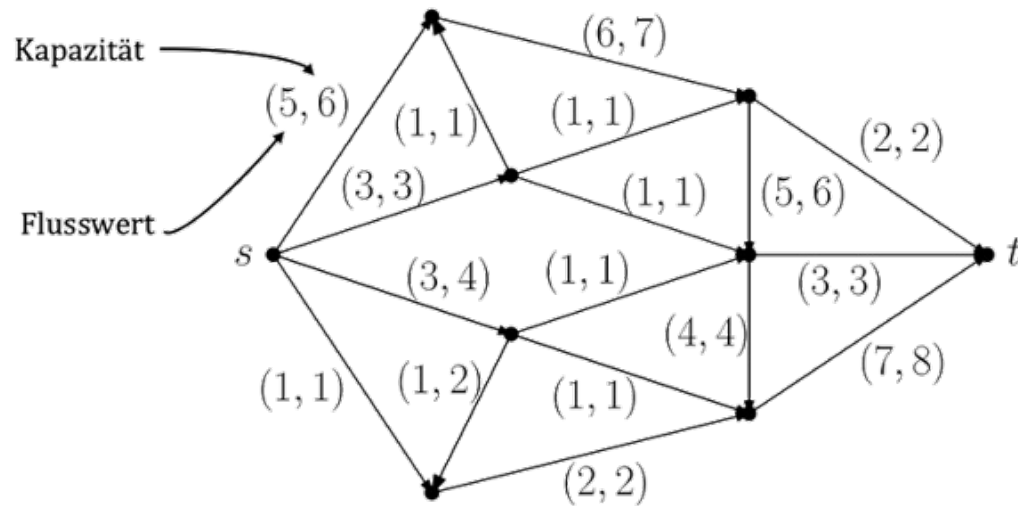


# Mathematische Methoden der Algorithmik – Vorlesung #10

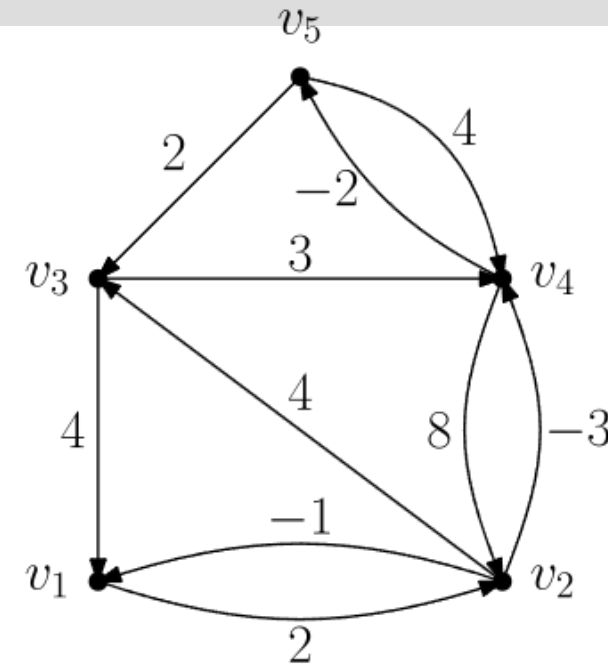
Arne Schmidt

# Letzte Woche

# Letze Woche - Graphenprobleme



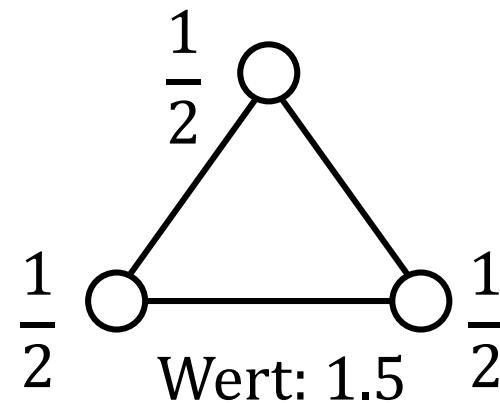
Maximum Flow -> Flow Constraints



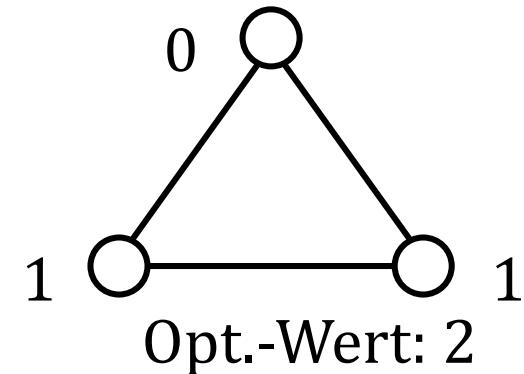
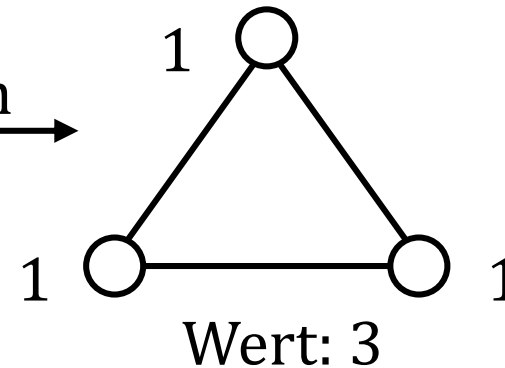
	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
$v_1$	0	2	-1	-1	-3
$v_2$	-1	0	-3	-3	-5
$v_3$	4	6	0	3	1
$v_4$	4	6	0	0	-2
$v_5$	6	8	2	4	0

Shortest Path -> Integral LP

Minimum Vertex Cover

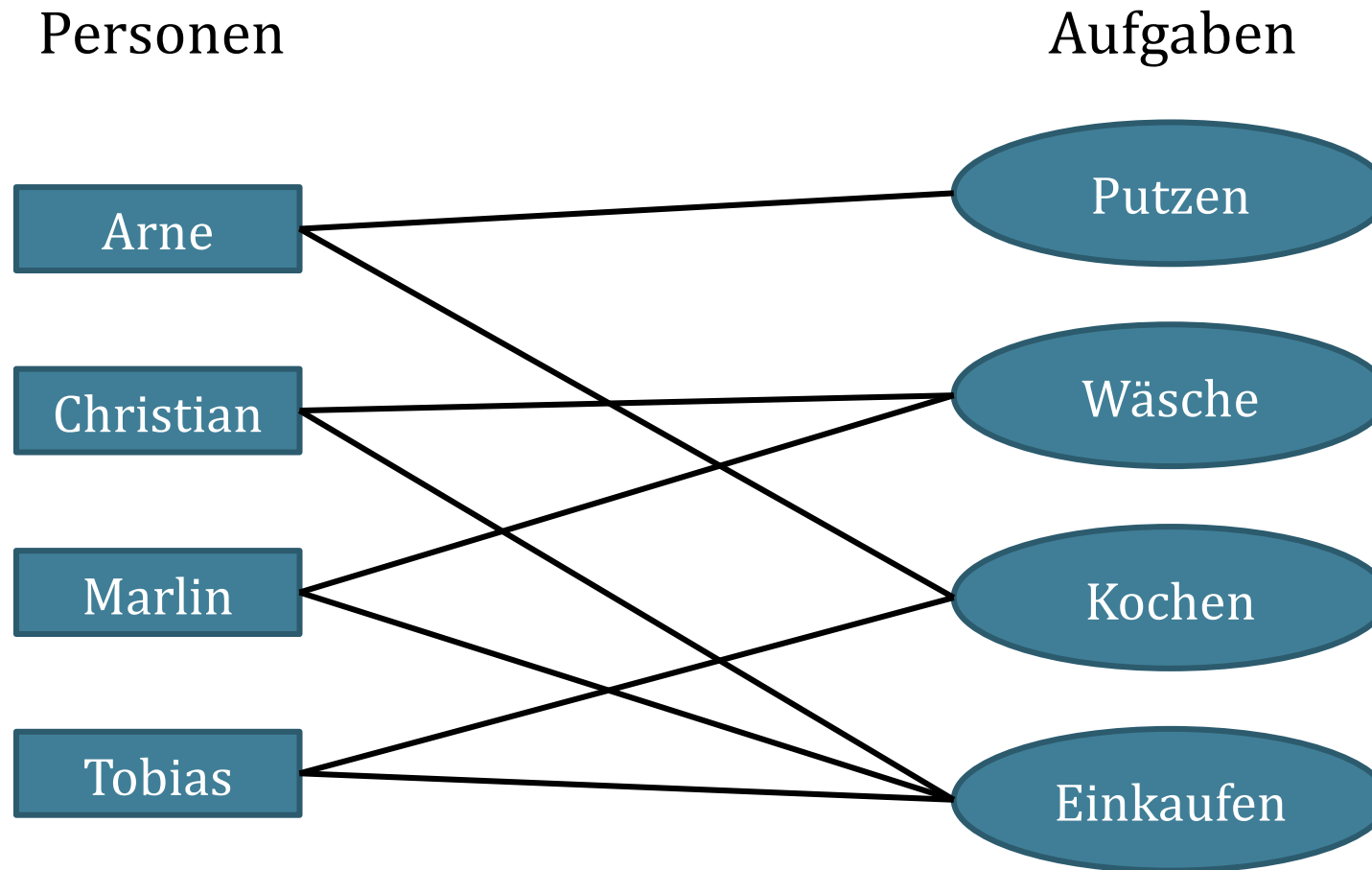


runden



# Graphenprobleme: Matchings

# Aufgabenverteilung



Möchten sich einer Aufgabe zuweisen.

Möchten einer Person zugewiesen werden.



# Matchings

## Definition (Matchings)

Sei  $G = (V, E)$  ein Graph

(1) Eine Kantenmenge  $M \subseteq E$  heißt **Matching**, wenn  $e \cap f = \emptyset$  für je zwei Kanten  $e, f \in M$  gilt.

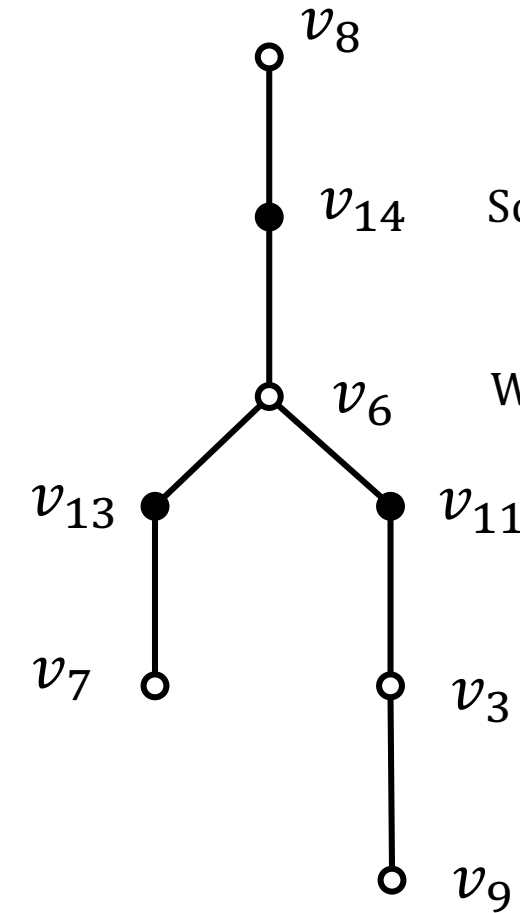
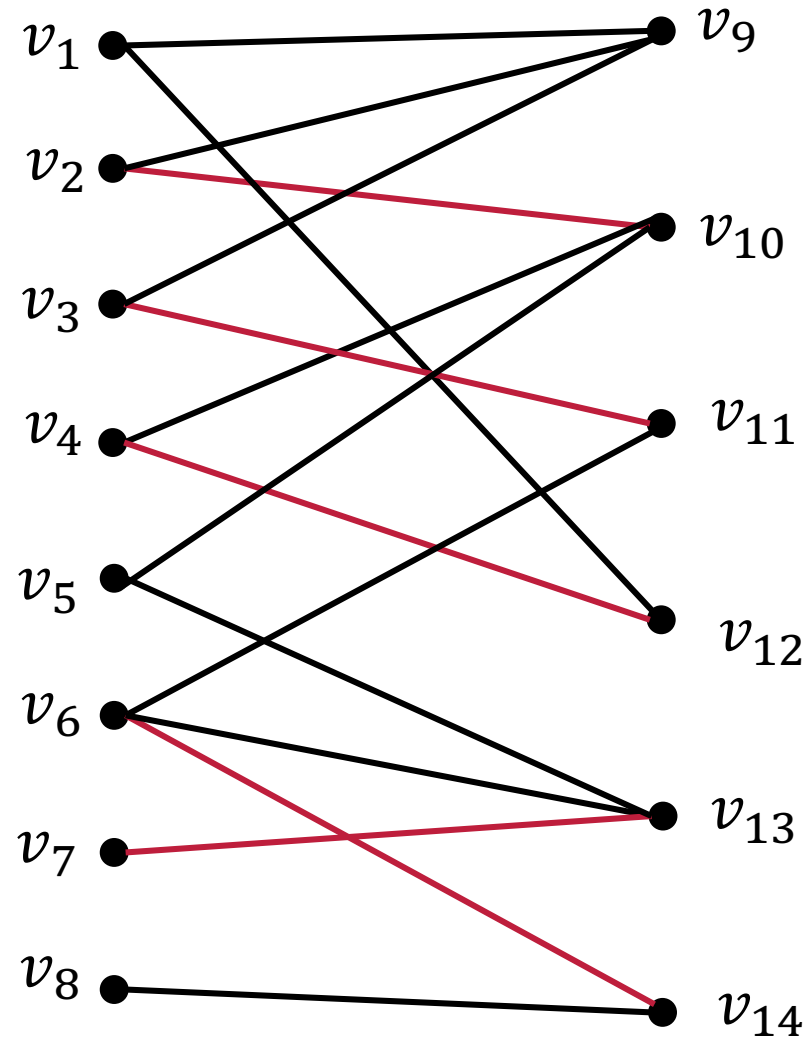
(2) Kanten in  $M$  heißen **unabhängig**.

(3) Ein Matching  $M$  heißt **perfektes Matching**, wenn  $2|M| = |V|$  gilt.

(4) Ein Matching heißt **inklusionsmaximal** (engl. **maximal**), wenn  $M \cup \{e\}$  für jede Kante  $e \in E \setminus M$  kein Matching ist.

(5) Ein Matching heißt **(kardinalitäts-)maximal** (engl. **maximum**), wenn kein Matching  $M'$  mit  $|M| < |M'|$  existiert.

# Algorithmus – Bipartite Graphen



Weiß:  $v_8$  muss gematcht werden.

Schwarz:  $v_{14}$  hat dann neuen Partner.

Weiß:  $v_6$  muss neu gematcht werden.

Alternierender Baum

# Matching für bipartite Graphen

## Algorithmus (Bipartite Matchings)

Eingabe:

Bipartiter Graph  $G = (V, E)$

Ausgabe:

Maximales Matching  $M$

```
1.  Function BIPARTITEMATCHING( $G$ )
2.      Set  $M := \emptyset$ 
3.      for  $r \in V_1$  mit  $r$  ungematcht do
4.          Setze  $T := (\{r\}, \emptyset)$  und  $W(T) := \{r\}$ 
5.          While (es ex. Kante  $\{v, w\} \in E$  mit  $v \in W(T)$  und  $w \notin V(T)$ )
6.              If  $w$  ist ungematcht then
7.                  Benutze  $w$ , um augmentierenden Pfad zu bilden.
8.                  Augmentiere  $M$ 
9.                  If es ex. Kein ungematchter Knoten mehr then
10.                     Return perfektes Matching  $M$ 
11.                  else
12.                      Gehe zu Zeile 3.
13.              else
14.                  Sei  $\{w, z\}$  Matchingkante an  $w$ .
15.                  Füge  $w, z$  zu  $V(T)$  hinzu und  $z$  zu  $W(T)$ 
16.                  Füge  $\{v, w\}, \{w, z\}$  zu  $E(T)$  hinzu.
17.      Return  $M$ 
```



# Matching Polytop

$$\begin{array}{ll} \max & \sum_{e \in E} x_e \\ \text{s. t.} & \\ & \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V \\ & x_e \geq 0, \quad \forall e \in E \end{array}$$

$\delta(v)$  ist die Menge an Kanten inzident zu  $v$ .

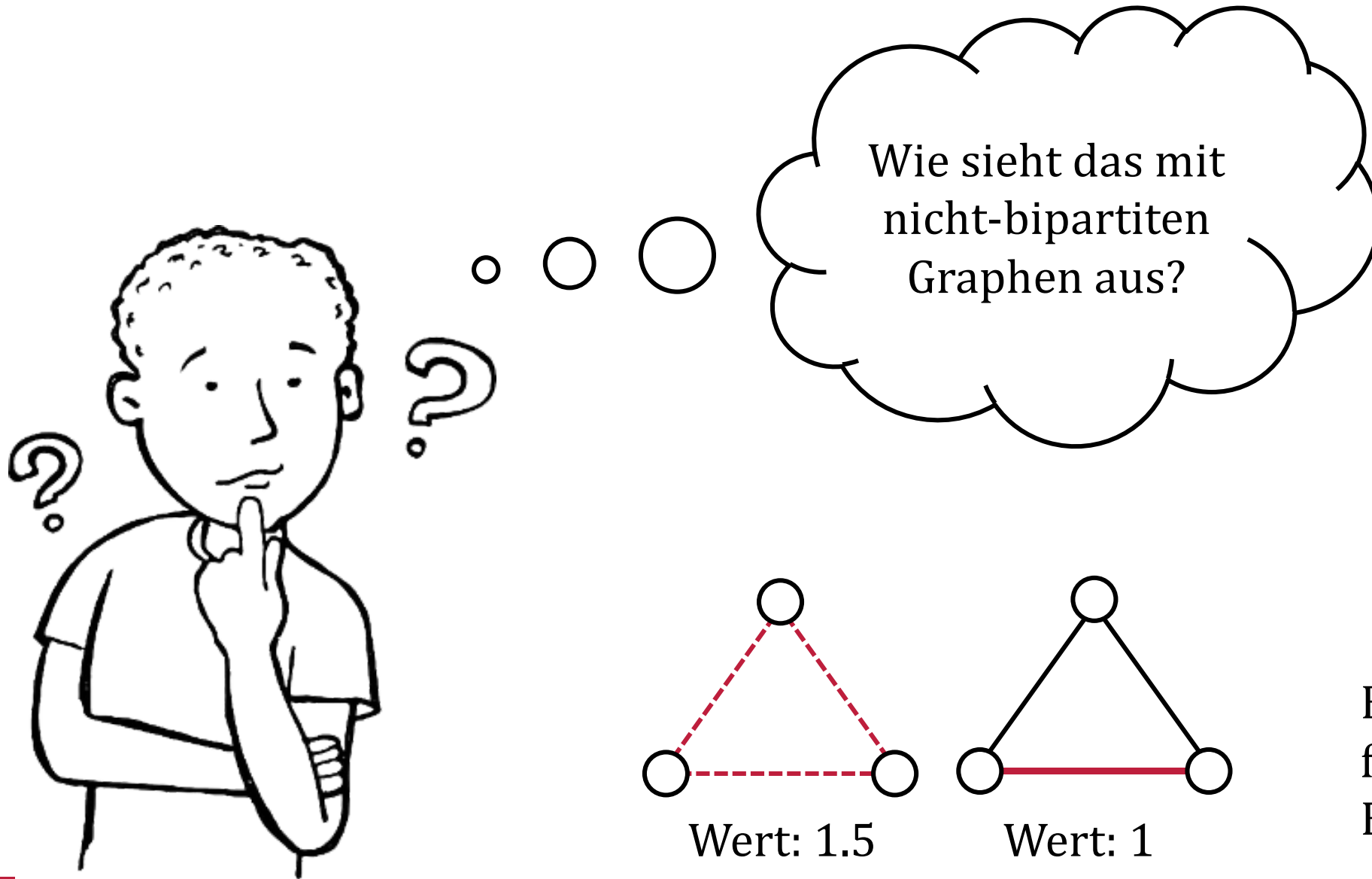
## Definition:

Das **Fraktionale Matching Polytop** (FMP) ist das durch das LP beschriebene Polytop (Lösungsraum).

Sind die Variablen aus  $\{0,1\}$ , beschreibt die konvexe Hülle der ganzzahligen Lösungspunkte das **(Integrale) Matching Polytop** (MP).

**Theorem:** Das FMP für bipartite Graphen ist ganzzahlig.

# Allgemeine Graphen



Es existieren  
fraktionale  
Basislösungen!

# Allgemeine Graphen mit perfekten Matchings



# Perfekte Matchings

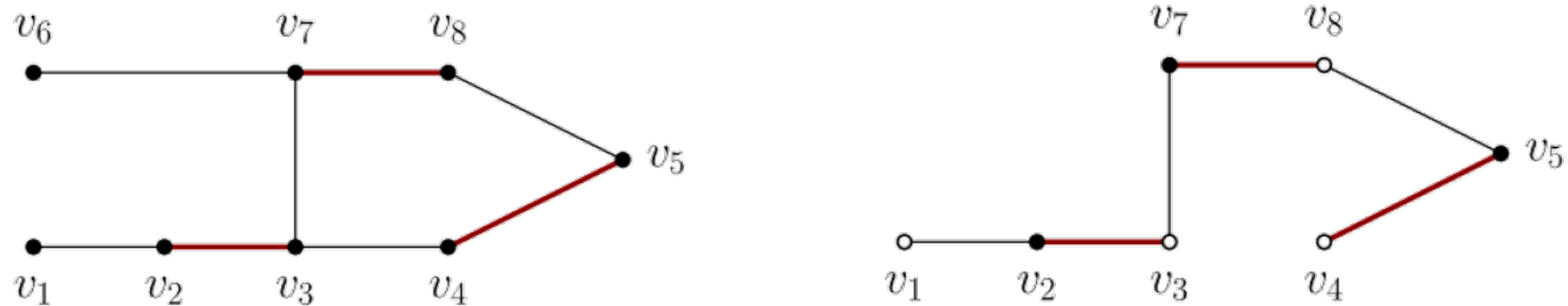
$$\begin{array}{ll} \max & \sum_{e \in E} x_e \\ \text{s. t.} & \\ & \sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \quad \forall v \in V \\ & \sum_{e \in E(B, V \setminus B)} x_e \geq 1, \quad \forall B \in \text{odd}(V) \\ & x_e \geq 0, \quad \forall e \in E \end{array}$$

Jeder Knoten muss gematcht sein.

Aus ungeraden Mengen muss eine Kante herausführen.

**Theorem:** Das PMP (perfekte Matching Polytop) für beliebige Graphen ist ganzzahlig.

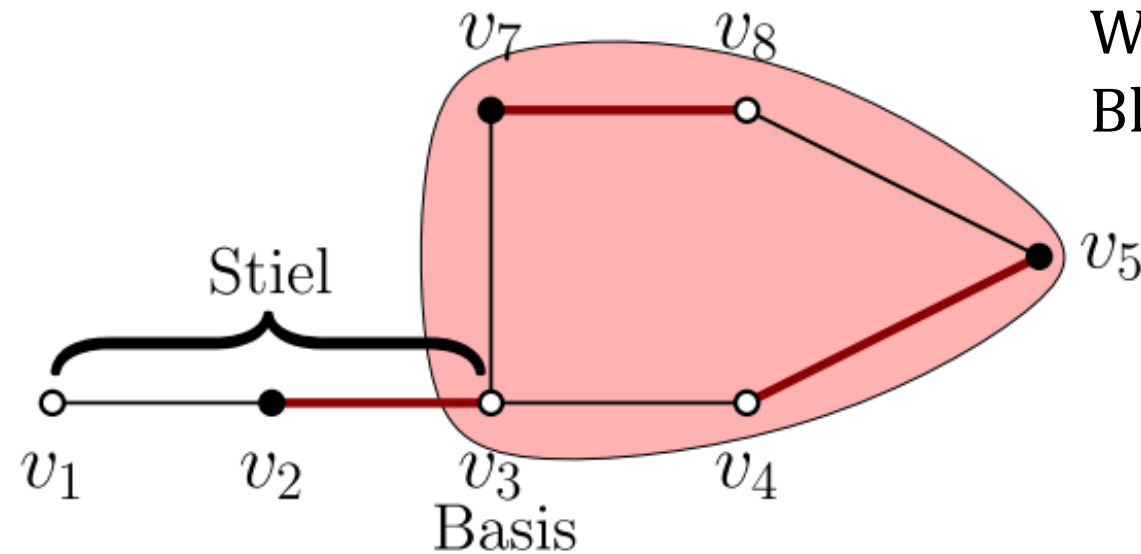
# Problem: Ungerade Kreise



Wie vermeide ich Kreise  
falsch herum abzulaufen?

Ungerader Kreis minus ein  
Knoten lässt sich perfekt  
matchen!

# Shrink it!



Weg:  $(v_1, v_2, v_3, v_7, v_8, v_5, v_4, v_3)$   
Blüte:  $(v_3, v_7, v_8, v_5, v_4, v_3)$

## Definition

Ein  $M$ -alternierender Weg  $(v_1, \dots, v_\ell)$  heißt  $M$ -Blume, wenn

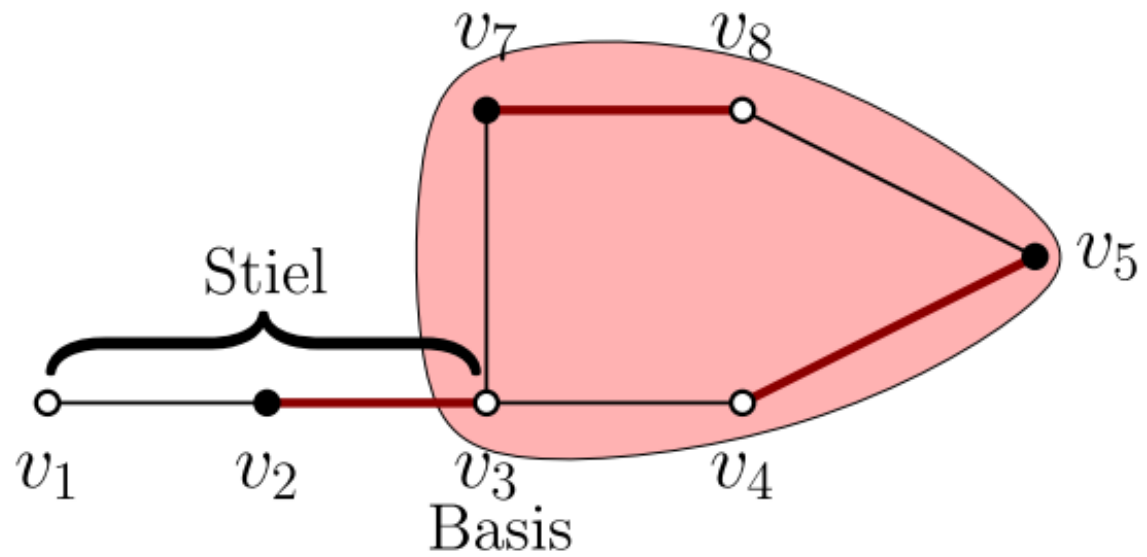
- $v_1$  nicht von  $M$  überdeckt ist.
- $v_1, \dots, v_{\ell-1}$  paarweise verschieden sind.
- $v_i = v_\ell$  für ein ungerades  $i \in \{1, \dots, \ell - 1\}$  gilt.

Dabei heißt  $(v_1, \dots, v_i)$   $M$ -Stiel,  $(v_i, \dots, v_\ell)$   $M$ -Blüte und  $v_i$  Basis.



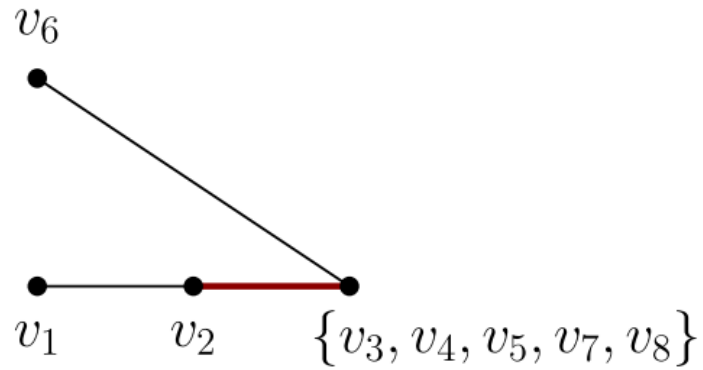
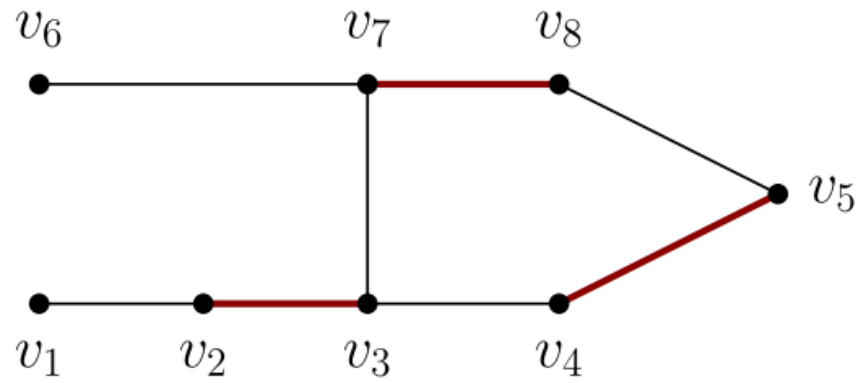
# Shrink it!

Idee: Schrumpfe  
Blüten und fahre mit  
der Suche fort.

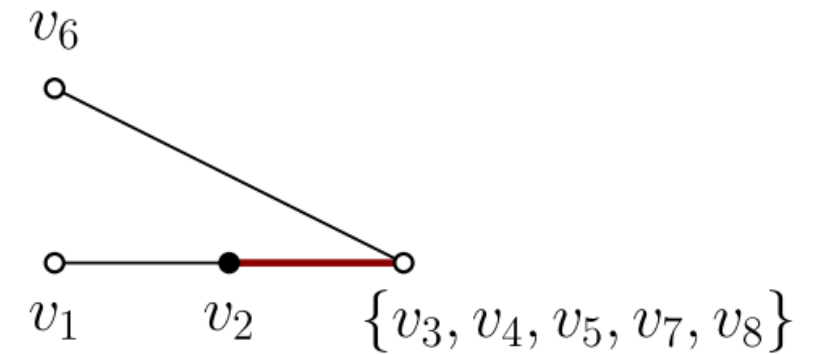
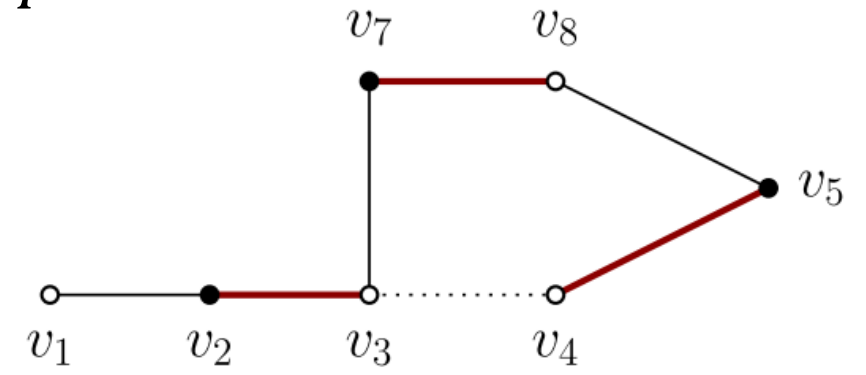


# Beispiel

$G'$

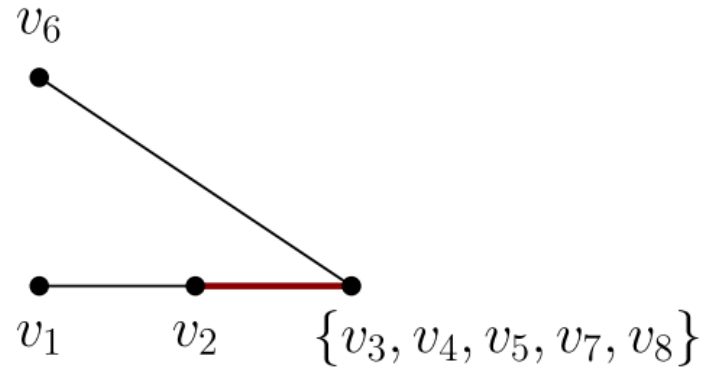


$T$

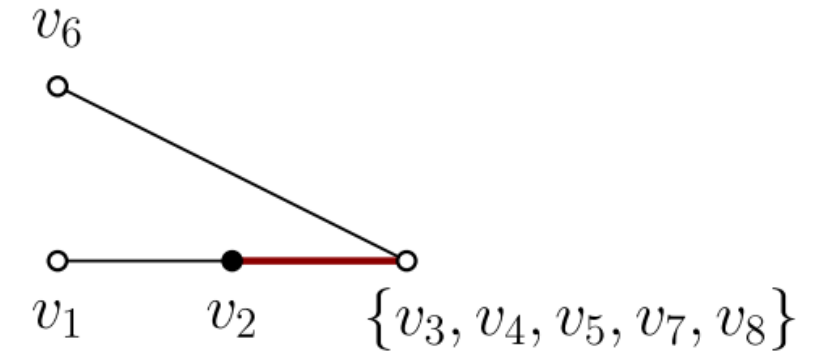


# Beispiel

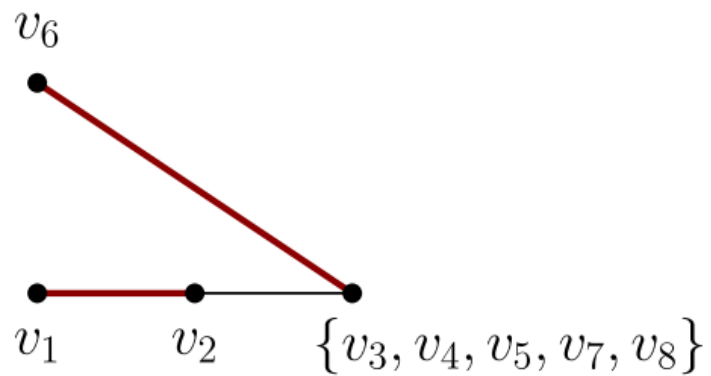
$G'$



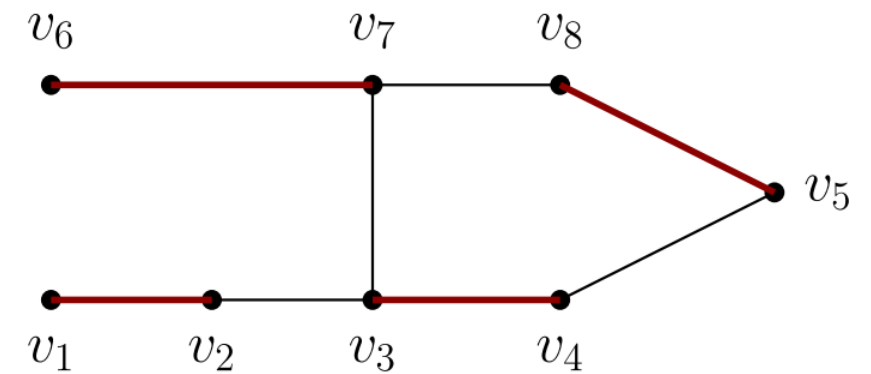
$T$



Augmentierender Pfad gefunden!



Blüte auflösen



# Algorithmus von Edmonds

## Algorithmus 5.22

Eingabe:

Bipartiter Graph  $G = (V, E)$

Ausgabe:

Maximales Matching  $M$

```
1.  Function BLOSSOM( $G$ )
2.      Setze  $M' = M = \emptyset$  und  $G' = G$ 
3.      for  $r \in V$  mit  $r$  ungematcht do
4.          Setze  $T := (\{r\}, \emptyset)$ ,  $W(T) := \{r\}$  und  $S(T) := \emptyset$ 
5.          While (es ex. Kante  $\{v, w\} \in E'$  mit  $v \in W(T)$  und  $w \notin S(T)$ )
6.              If  $w$  ist ungematcht then
7.                  Benutze  $\{v, w\}$ , um  $M'$  zu augmentieren.
8.                  Erweitere  $M'$  zu einem Matching  $M$  von  $G$ 
9.                  Ersetze  $M'$  durch  $M$  und  $G'$  durch  $G$ .
10.                 Gehe zu Zeile 3.
11.             else if  $w \notin V(T)$ , aber  $w$  ist in  $M'$  gematcht then
12.                 Benutze  $\{v, w\}$ , um  $T$  zu erweitern.
13.             else if  $w \in W(T)$  then
14.                 Benutze  $\{v, w\}$  zum Schrumpfen einer Blüte.
15.                 Aktualisiere  $M'$ ,  $G'$  und  $T$  entsprechend.
16.      Return  $M$ 
```

# Gewichtete Matchings

# Gewichtete Matchings

## Problem 5.24: Minimum Cost Perfect Matchings

Gegeben:

Graph  $G = (V, E)$  und Kantenkosten  $c: E \rightarrow \mathbb{R}^+$

Gesucht:

Perfektes Matching  $M \subseteq E$  mit  $\sum_{e \in M} c(e)$  minimal.

Wichtig:

- Annahme  $G$  enthält ein perfektes Matching
- $\Rightarrow$  Für jeden Knoten existiert ein augmentierender Pfad



# IP und LP zu MinCost Perfect Matchings

„Relaxierung“

$$\begin{array}{ll} \min & \sum_{e \in E} x_e c(e) \\ \text{s. t.} & \\ & \sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \quad \forall v \in V \\ & \sum_{e \in E(B, V \setminus B)} x_e \geq 1, \quad \forall B \in \text{odd}(V) \\ & x_e \in \{0, 1\}, \quad \forall e \in E \end{array}$$

$$\begin{array}{ll} \min & \sum_{e \in E} x_e c(e) \\ \text{s. t.} & \\ & \sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \quad \forall v \in V \\ & \sum_{e \in E(B, V \setminus B)} x_e \geq 1, \quad \forall B \in \text{odd}(V) \\ & x_e \geq 0, \quad \forall e \in E \end{array}$$

Das rechte LP ist ganzzahlig!

# Zwei lineare Programme

Optimale Lösungswerte stimmen überein!

$$\begin{aligned} \min & \sum_{e \in E} x_e c(e) \\ \text{s. t.} & \\ & \sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \quad \forall v \in V \\ & \sum_{e \in E(B, V \setminus B)} x_e \geq 1, \quad \forall B \in \text{odd}(V) \\ & x_e \geq 0, \quad \forall e \in E \end{aligned}$$

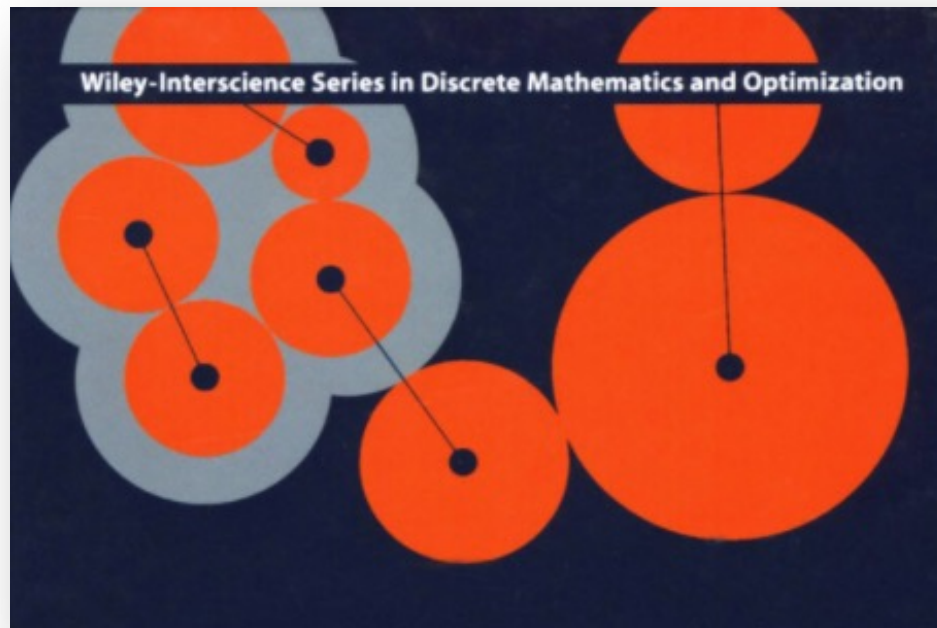
Primal

$$\begin{aligned} \max & \sum_{v \in V} y_v + \sum_{B \in \text{odd}(V)} y_B \\ \text{s. t.} & \\ & y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \text{odd}(V)}} y_B \leq c(e), \quad \forall e \in E \\ & y_B \geq 0, \quad \forall B \in \text{odd}(V) \end{aligned}$$

Dual

Was ist das für ein Problem?

# Zwei lineare Programme



Lass Kreise um Knoten  
möglichst groß wachsen,  
sodass sie sich nicht schneiden.

$$\begin{aligned} \max \quad & \sum_{v \in V} y_v + \sum_{B \in \text{odd}(V)} y_B \\ \text{s. t.} \quad & \\ y_v + y_w + & \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \text{odd}(V)}} y_B \leq c(e), \quad \forall e \in E \\ & y_B \geq 0, \quad \forall B \in \text{odd}(V) \end{aligned}$$

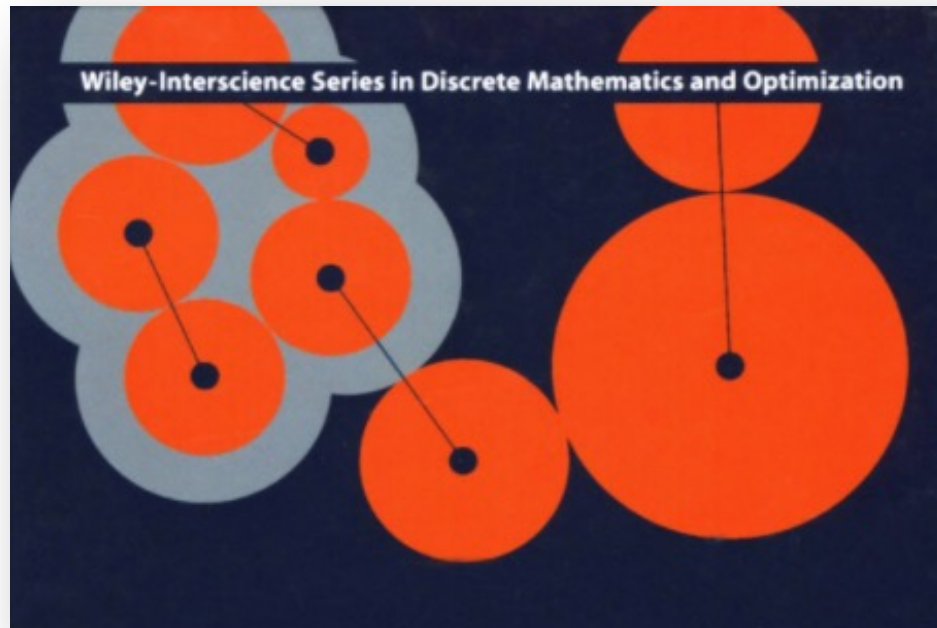
Dual

Was ist das für ein Problem?

# Dualität

Aus komplementären Schlupf folgt:  
Sei  $G$  ein Graph und  $e = \{v, w\} \in E$ . Dann gilt:

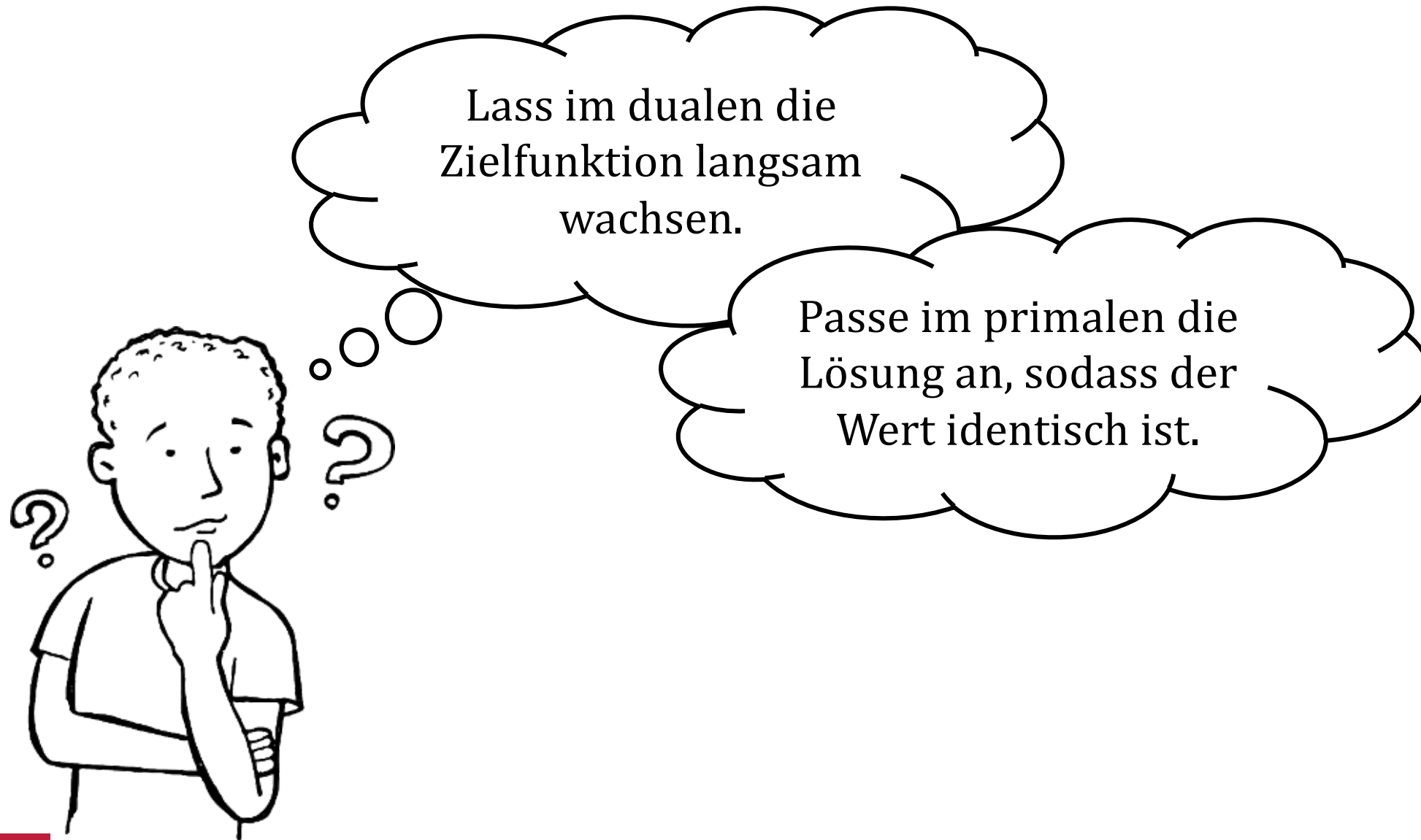
$$x_e = 1 \Rightarrow y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \text{odd}(V)}} y_B = c(e)$$



Kante kann nur eine Matchingkante sein, wenn sie durch Kreise überdeckt ist!

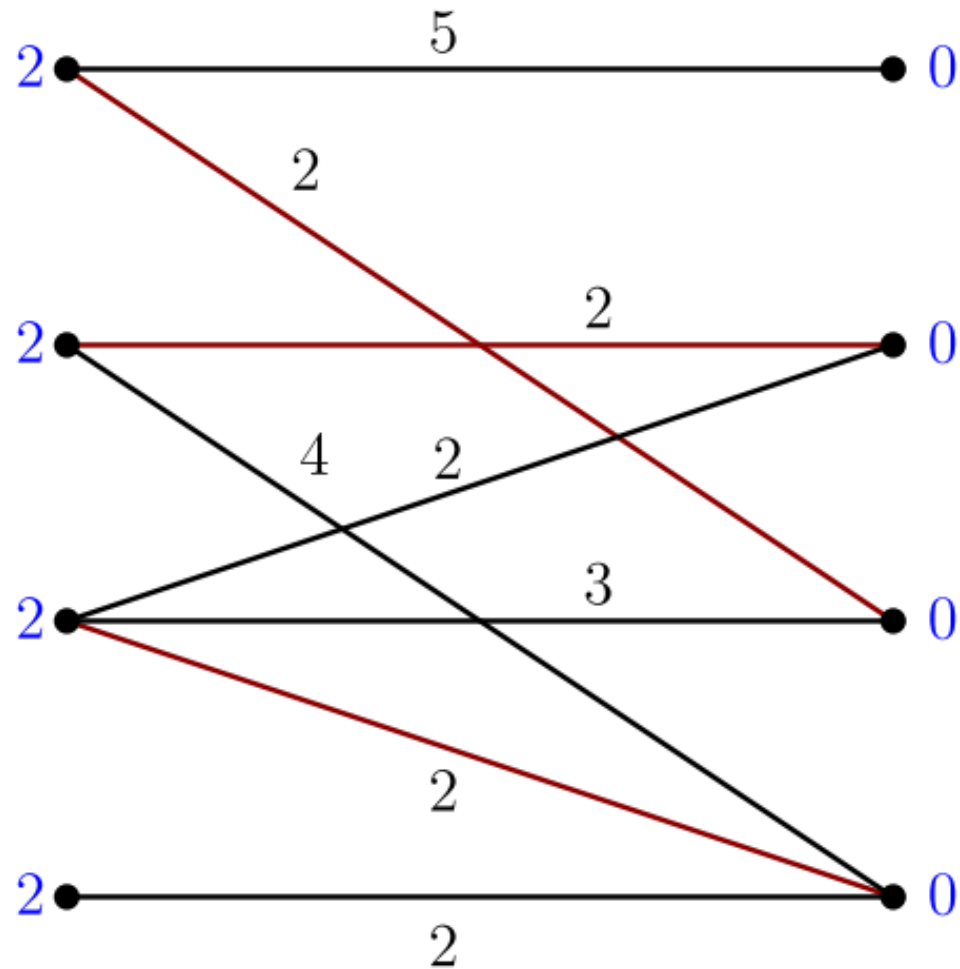
# Primal-Dual-Algorithmen

# Idee

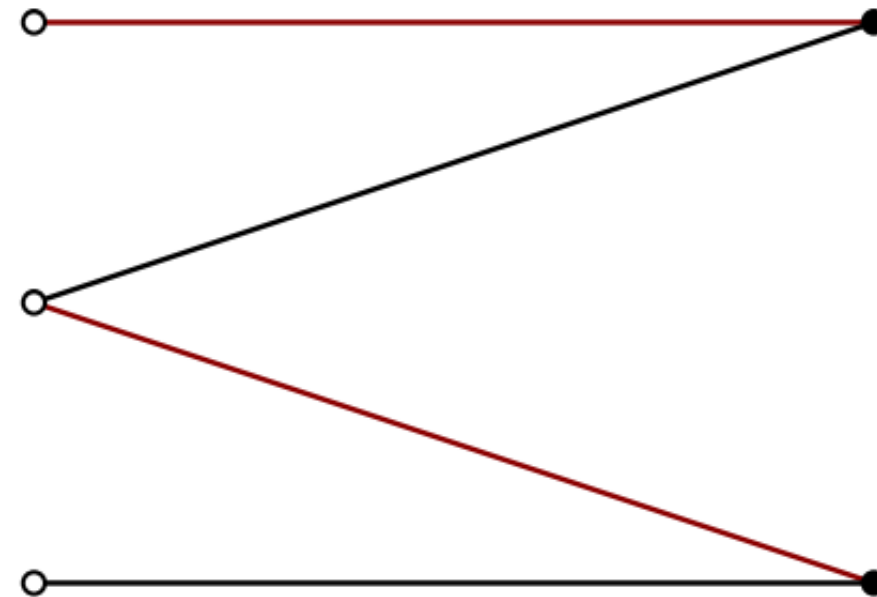




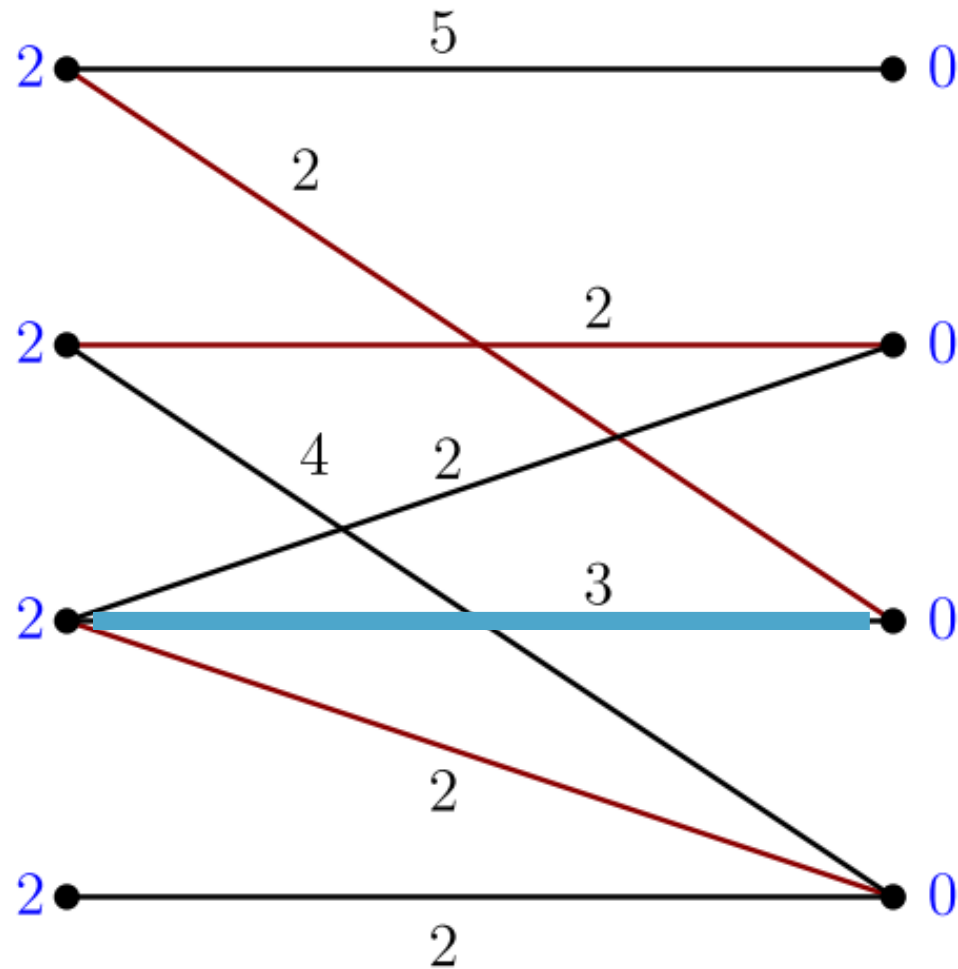
# Finden von augmentierenden Pfaden



Konstruiere alternierenden Baum  
nur über überdeckte Kanten.



# Aktualisieren der Dual-Werte



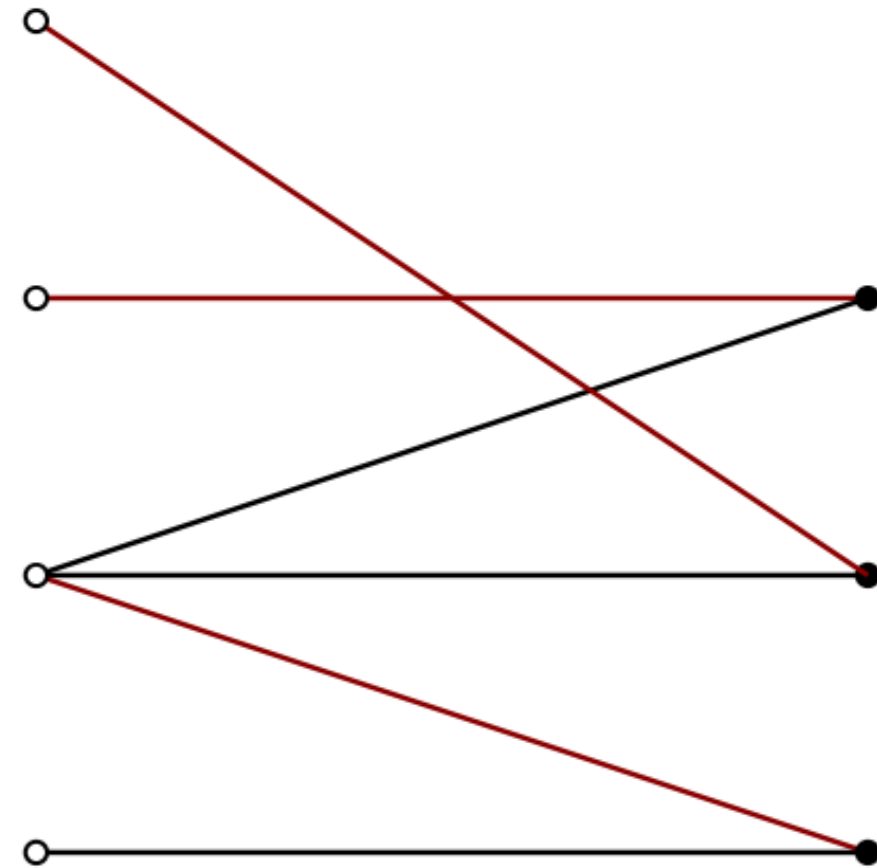
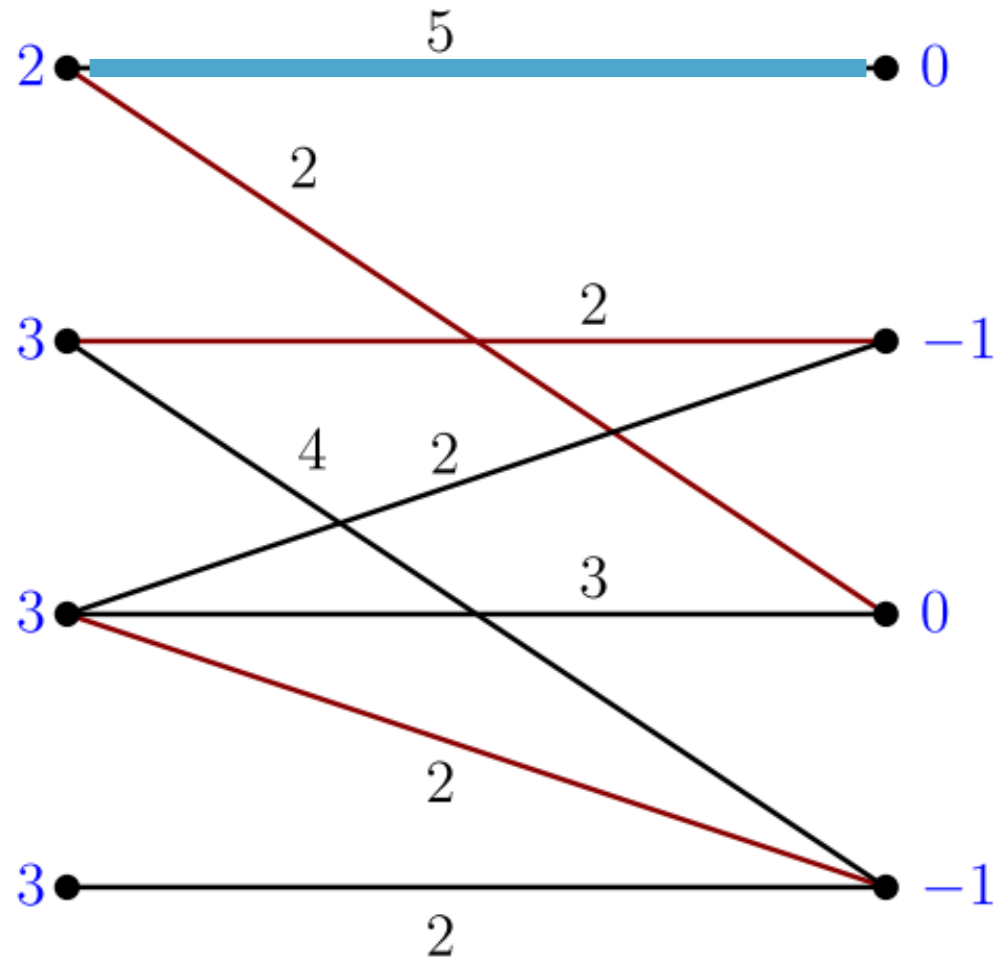
Suche kleinste reduzierte Kosten  
von Kanten, die herausführen.

$\Rightarrow 1$

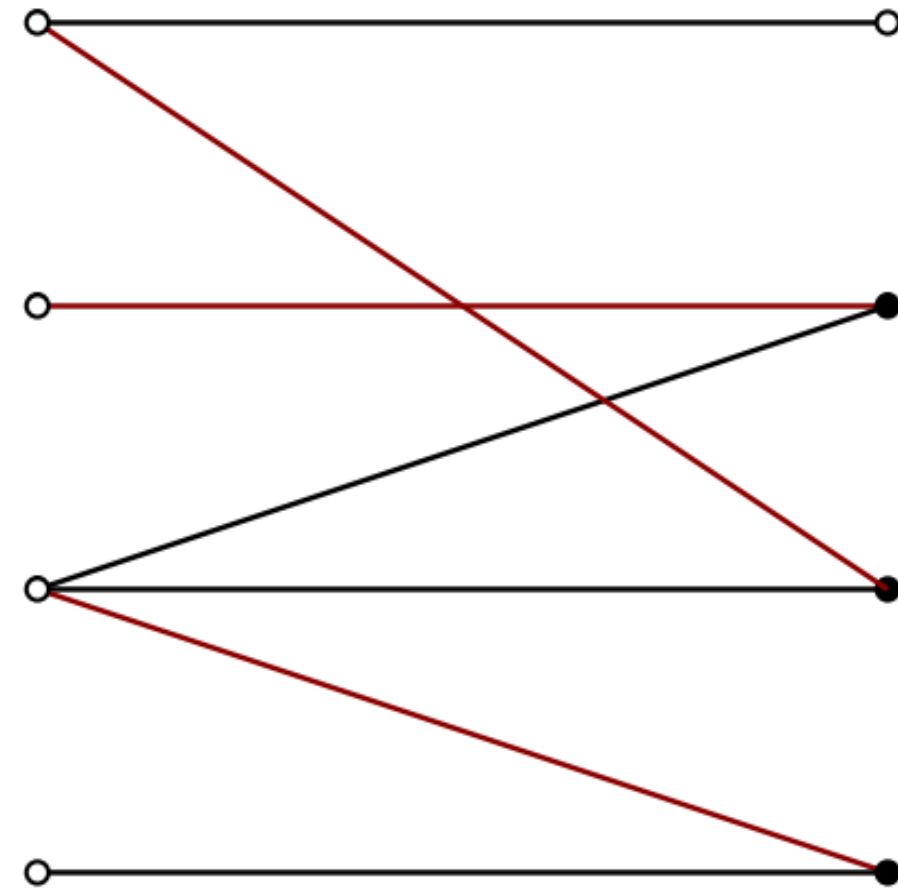
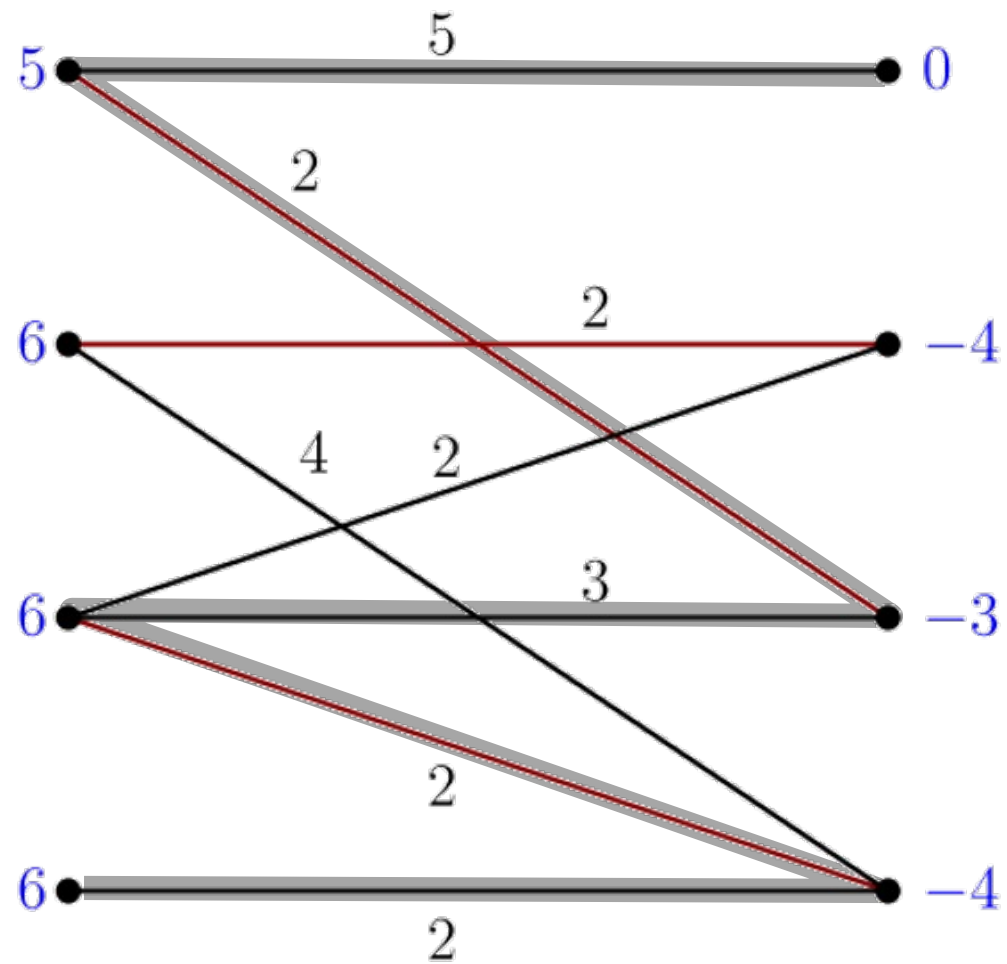
Erhöhe/Verringere Werte von  
Knoten um 1.

Aber wie?

# Nächste Iteration



# Nächste Iteration



# Ungarische Methode

## Algorithmus

Eingabe:

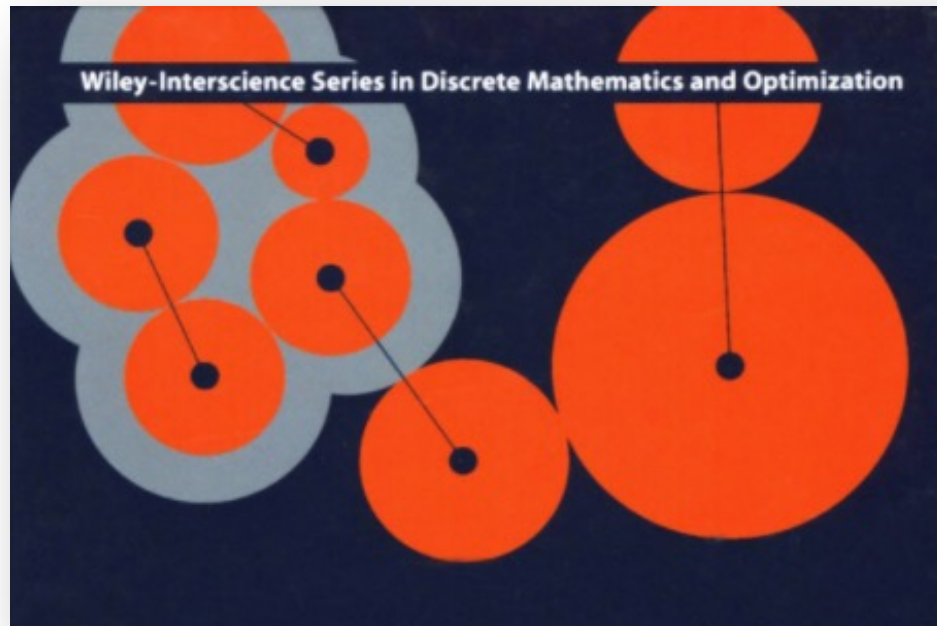
Bipartiter Graph  $G = (V = V_1 \dot{\cup} V_2, E)$ , Kantenkosten  $c: E \rightarrow \mathbb{R}^+$

Ausgabe:

Min Cost Perfect Matching  $M$

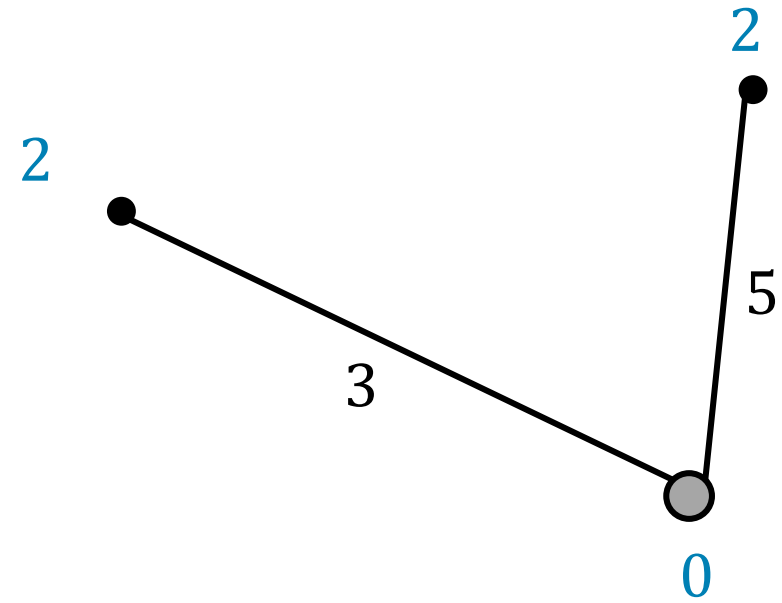
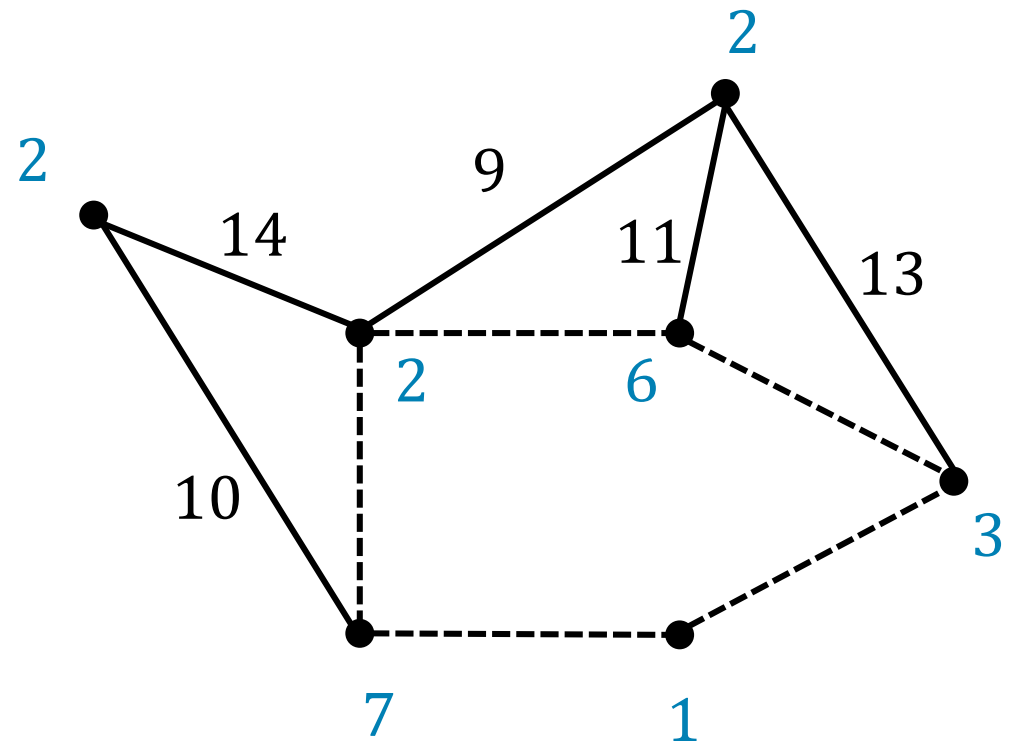
1. **Function** UNGARISCHEMETHODE( $G, c$ )
2. Setze  $y_v := \min_{\{v,w\} \in E} c(\{v,w\})$  für alle  $v \in V_1$ .
3. Finde maximales Matching  $M$  auf überdeckten Kanten.
4. **for**  $r \in V_1$  mit  $r$  ungematcht **do**
5.     Konstruiere alternierenden Baum  $T$  mit  $r$  als Start.
6.     **while**  $T$  enthält keinen augmentierenden Pfad **do**
7.         Sei  $e \in E(W(T), V \setminus W(T))$  eine Kante mit  $\bar{c}(e)$  minimal.
8.         Setze  $y_v := y_v + \bar{c}(e)$  für alle  $v \in W(T)$ .
9.         Setze  $y_w := y_w - \bar{c}(e)$  für alle  $w \in V(T) \setminus W(T)$ .
10.        Erweitere  $T$
11.     Augmentiere  $M$
12. **return**  $M$

# Ungerade Kreise

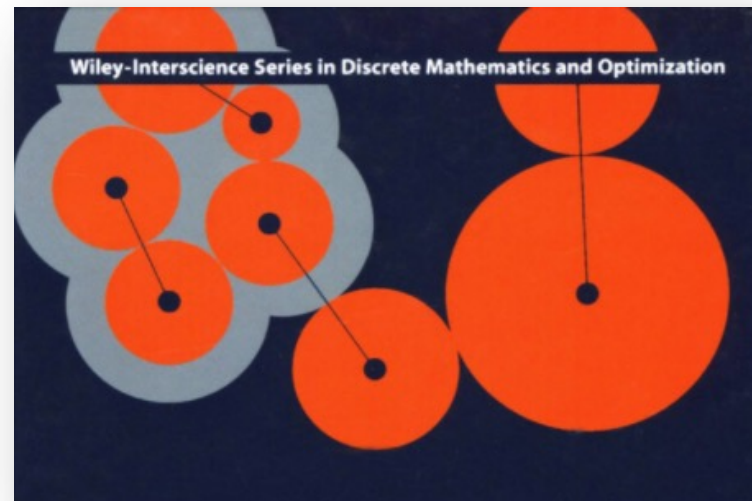


$$\begin{aligned} \max \quad & \sum_{v \in V} y_v + \sum_{B \in \text{odd}(V)} y_B \\ \text{s. t.} \quad & \\ & y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \text{odd}(V)}} y_B \leq c(e), \quad \forall e \in E \\ & y_B \geq 0, \quad \forall B \in \text{odd}(V) \end{aligned}$$

# Schrumpfen einer Blüte



Kanten teilweise über Knoten im Kreis überdeckt.

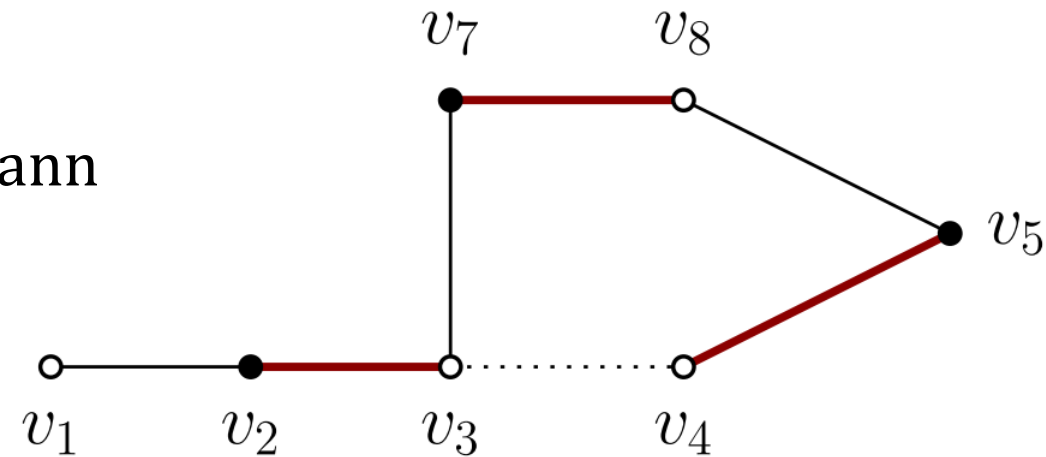


# Die Probleme

## Problem 1:

Erhöhen von Dualwerten von weißen Knoten kann Kanten zwischen weißen Knoten überdecken.

⇒ maximal um  $\frac{\bar{c}(e)}{2}$  erhöhen.



## Problem 2:

Nach Augmentieren: Eine Blüte B kann nur aufgelöst werden, wenn  $y_B = 0$ .

⇒ B kann später als schwarzer Knoten existieren.

⇒ Schwarze Knoten dürfen um maximal  $y_B$  verringert werden.

$$y_B \geq 0, \quad \forall B \in \text{odd}(V)$$



# Updaterregeln

Betrachte:

1.  $\varepsilon_1 := \min\{\bar{c}(e) \mid e \in E(W(T), V \setminus W(T))\}$
2.  $\varepsilon_2 := \frac{1}{2} \min\{\bar{c}(e) \mid e = \{v, w\} \text{ mit } v, w \in W(T)\}$
3.  $\varepsilon_3 := \min\{y_B \mid \text{Blüte } B \in S(T)\}$

Sei  $\varepsilon := \min(\varepsilon_1, \varepsilon_2, \varepsilon_3)$

Erhöhe Knoten aus  $W(T)$  um  $\varepsilon$  und verringere Knoten aus  $S(T)$  um  $\varepsilon$ .

Auch hier:

Nach Aktualisieren sind bereits überdeckte Kanten immer noch überdeckt und reduzierte Kosten sind nicht-negativ.

# Primal-Dual Methode

## Algorithmus

Eingabe:

Graph  $G = (V, E)$ , Kantenkosten  $c: E \rightarrow \mathbb{R}^+$

Ausgabe:

Min Cost Perfect Matching  $M$

1. **Function** PRIMALDUAL( $G, c$ )
2. Setze  $G' := G$  und  $M' = M = \emptyset$
3. **for**  $r \in V$  mit  $r$  ungematcht **do**
4.     Konstruiere alternierenden Baum  $T$  mit  $r$  als Start.
5.     **while**  $T$  enthält keinen augmentierenden Pfad **do**
6.         Bestimme  $\varepsilon := \min(\varepsilon_1, \varepsilon_2, \varepsilon_3)$
7.         Setze  $y_v := y_v + \varepsilon$  für alle  $v \in W(T)$ .
8.         Setze  $y_w := y_w - \varepsilon$  für alle  $w \in S(T)$ .
9.         ***Erweitere**  $T$*  (solange möglich)
10.     Augmentiere  $M'$
11. Konstruiere Matching  $M$  für  $G$  aus  $M'$  in  $G'$ .
12. **return**  $M$

Erweitere  $T$ :

**Case**  $\exists$  Blüte  $B \in S(T)$  mit  $y_B = 0$ :

Löse Blüte auf.

Passe  $G', M'$  und  $T$  entsprechend an.

**Case** Blüte  $B$  gefunden:

Schrumpfe  $B$ .

Passe  $G', M'$  und  $T$  entsprechend an.

**Case** Sonst:

Erweitere  $T$  über überdeckte Kanten.