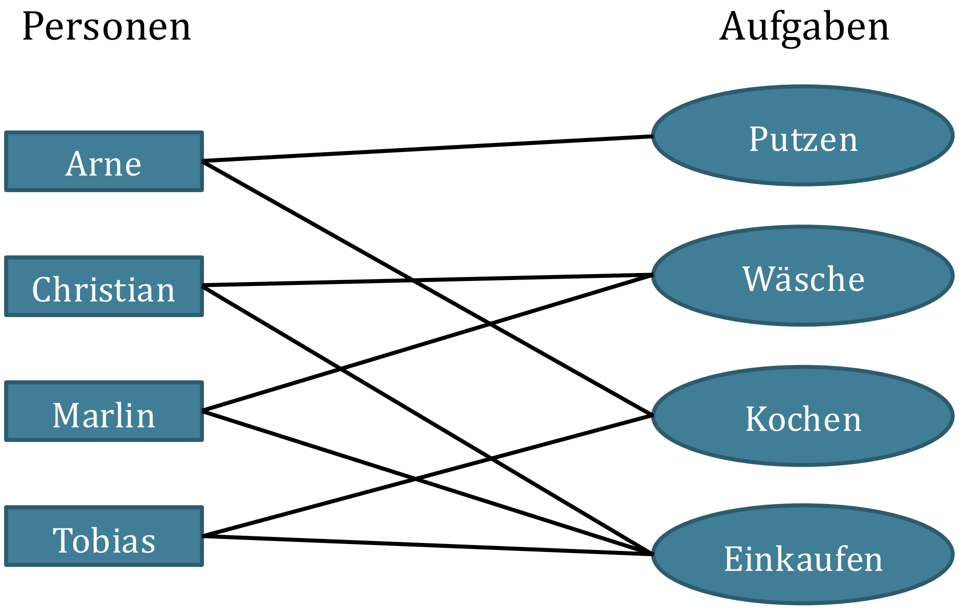
Technische Universität Braunschweig

Mathematische Methoden der Algorithmik

Arne Schmidt

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.

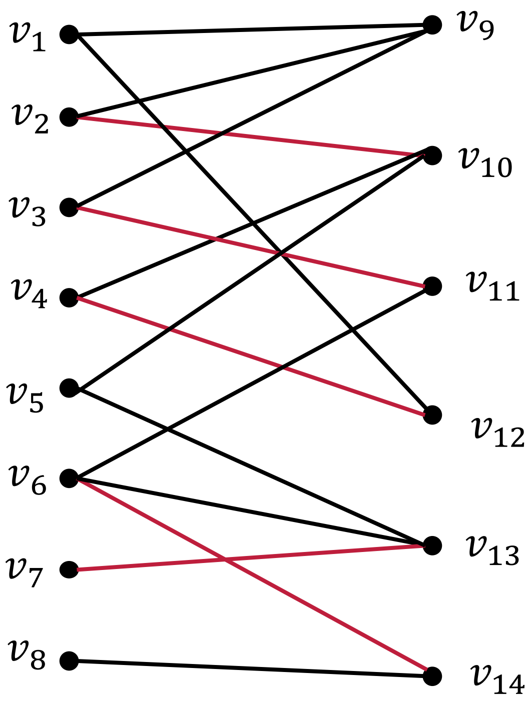
(2) Kanten in M heißen **unabhängig**.

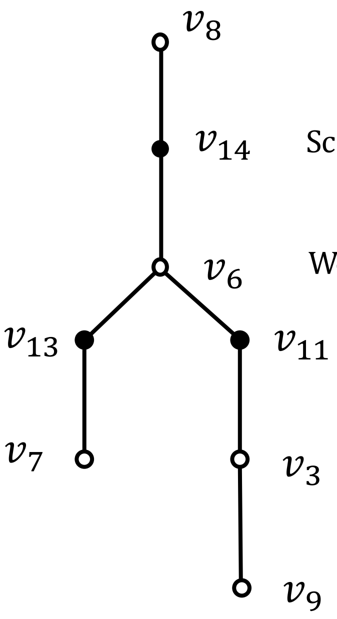
(3) Ein Matching M heißt **perfektes Matching**, wenn $2|M| = |V|$ gilt.

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

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

Algorithmus – Bipartite Graphen





Alternierender Baum

Weiß: v_8 muss gematcht werden.

Schwarz: v_{14} hat dann neuen Partner.

Weiß: v_6 muss neu gematcht werden.

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

$$\max \sum_{e \in E} x_e$$
$$\text{s.t.}$$
$$\sum_{e \in \delta(v)} x_e \leq 1, \qquad \forall v \in V$$
$$x_e \geq 0, \qquad \forall e \in E$$

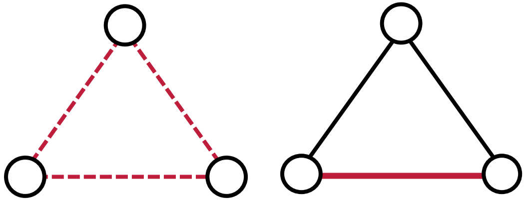
$\delta(v)$ ist die Menge an Kanten inzident zu v.

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



Wert: 1.5 Wert: 1

Es existieren fraktionale Basislösungen!

Allgemeine Graphen mit perfekten Matchings

Dieser Graph hatte kein perfektes Matching. Kann man unter der Annahme das perfekte Matchings existieren, mehr garantieren?

Perfekte Matchings

$$\max \sum_{e \in E} x_e$$

s. t.

$$\sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \qquad \forall v \in V$$

$$\sum_{e \in E(B, V \setminus B)} x_e \geq 1, \qquad \forall B \in \text{odd}(V)$$

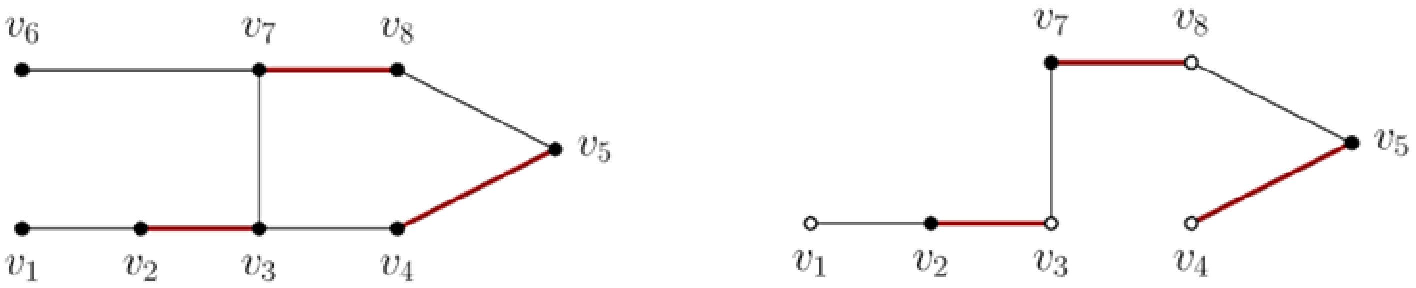
$$x_e \geq 0, \qquad \forall e \in E$$

Jeder Knoten muss gematcht sein.

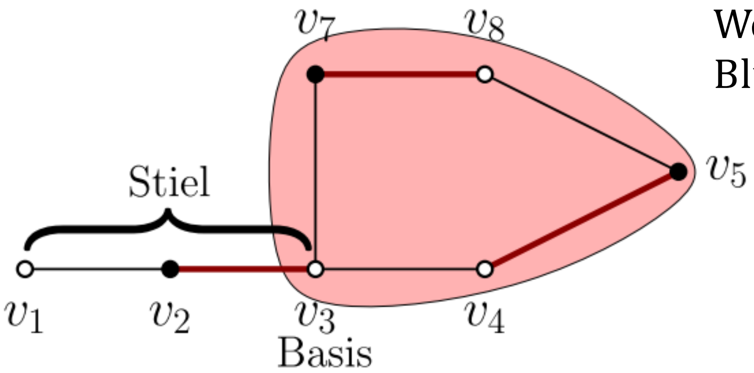
Aus ungeraden Mengen muss eine Kante herausführen.

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

Problem: Ungerade Kreise



Shrink it!

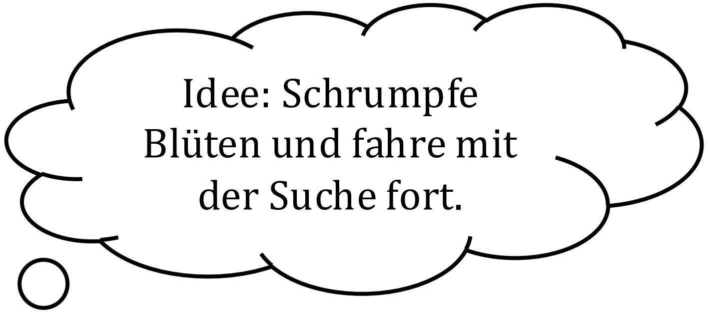


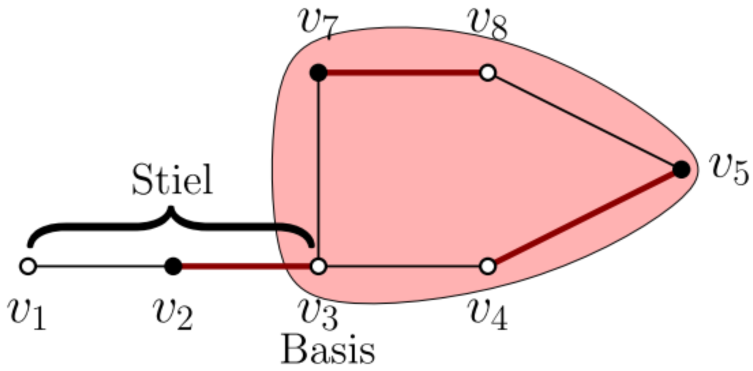
Weg: $(v_1, v_2, v_3, v_7, v_8, v_5, v_4, v_3)$
Blüte: $(v_3, v_7, v_8, v_5, v_4, v_3)$

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.

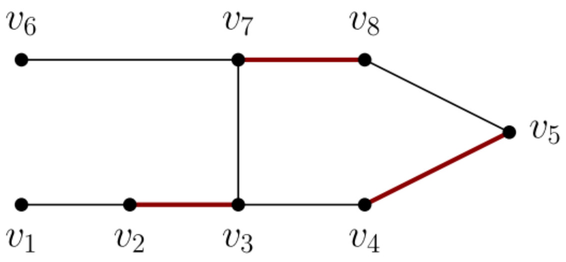
Shrink it!


Idee: Schrumpfe Blüten und fahre mit der Suche fort.


v_7
v_8
v_5
Stiel
v_1
v_2
v_3
v_4
Basis

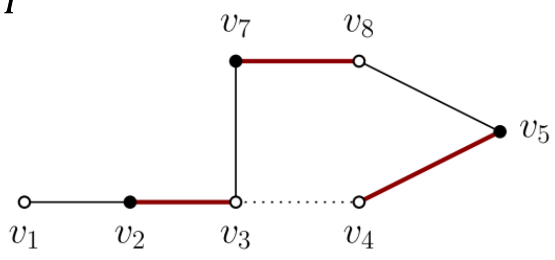
Beispiel

G'

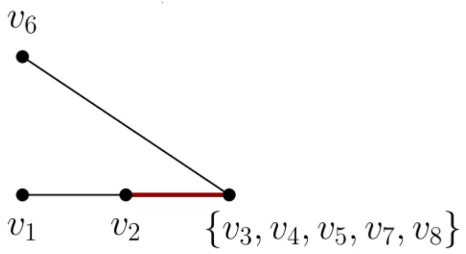


v_6
v_7
v_8
v_5
v_1
v_2
v_3
v_4

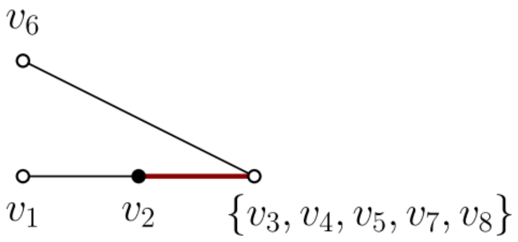
T



v_7
v_8
v_5
v_1
v_2
v_3
v_4



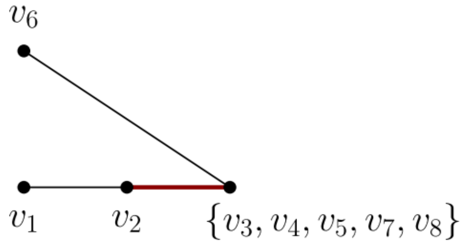
v_6
v_1
v_2
$\{v_3, v_4, v_5, v_7, v_8\}$



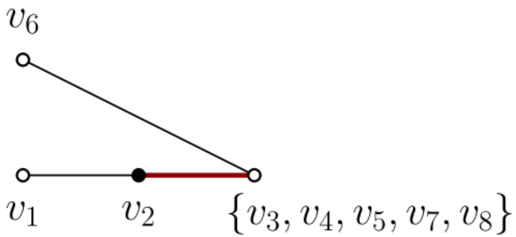
v_6
v_1
v_2
$\{v_3, v_4, v_5, v_7, v_8\}$

Beispiel

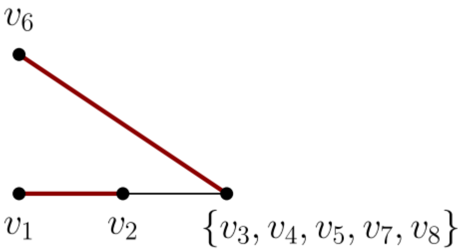
G'



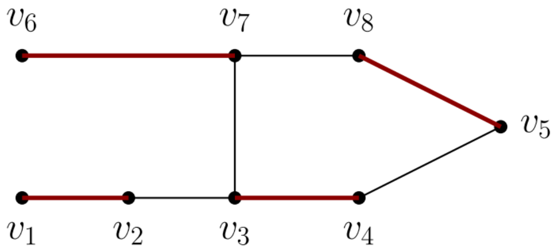
T



Augmentierender Pfad gefunden!



Blüte auflösen



Algorithmus von Edmonds

Algorithmus
Eingabe:
Bipartiter Graph $G = (V, E)$
Ausgabe:
Maximales Matching M

```
1.  Function BLOSSOM(G)
2.      Setze M′ = M = ∅ und G′ = G
3.      for r ∈ V mit r ungematcht do
4.          Setze T ≔ ({r}, ∅), W(T) ≔ {r} und S(T) ≔ ∅
5.          While (es ex. Kante {v, w} ∈ E′ mit v ∈ W(T) und w ∉ 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 ∉ V(T), aber w ist in M′ gematcht then
12.                 Benutze {v, w}, um T zu erweitern.
13.             else if w ∈ 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“

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

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

Das rechte LP ist ganzzahlig!

Zwei lineare Programme

Optimale Lösungswerte stimmen überein!

$$\min \sum_{e \in E} x_e c(e)$$

s.t.

$$\sum_{e \in E(\{v\}, V \setminus \{v\})} x_e = 1, \qquad \forall v \in V$$

$$\sum_{e \in E(B, V \setminus B)} x_e \geq 1, \qquad \forall B \in \mathrm{odd}(V)$$

$$x_e \geq 0, \qquad \forall e \in E$$

Primal

$$\max \sum_{v \in V} y_v + \sum_{B \in \mathrm{odd}(V)} y_B$$

s.t.

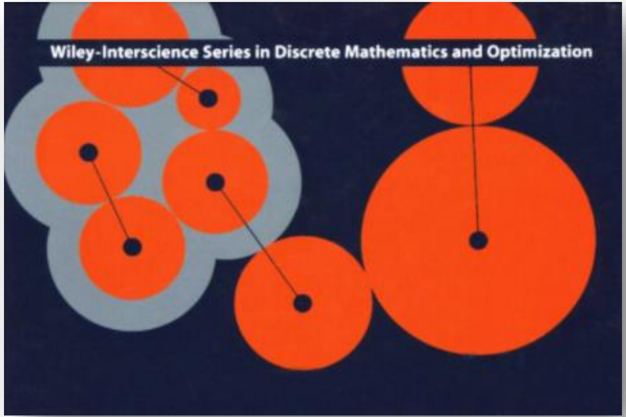
$$y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \mathrm{odd}(V)}} y_B \leq c(e), \qquad \forall e \in E$$

$$y_B \geq 0, \qquad \forall B \in \mathrm{odd}(V)$$

Dual

Was ist das für ein Problem?

Zwei lineare Programme



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

$$\max \sum_{v \in V} y_v + \sum_{B \in \mathrm{odd}(V)} y_B$$

s. t.

$$y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \mathrm{odd}(V)}} y_B \leq c(e), \qquad \forall e \in E$$

$$y_B \geq 0, \qquad \forall B \in \mathrm{odd}(V)$$

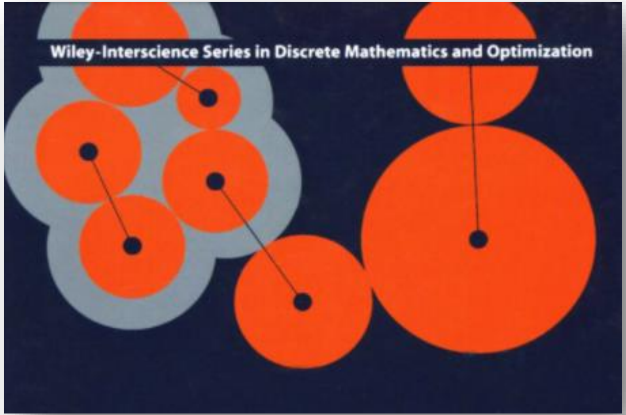
Dual

Was ist das für ein Problem?

Dualität

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

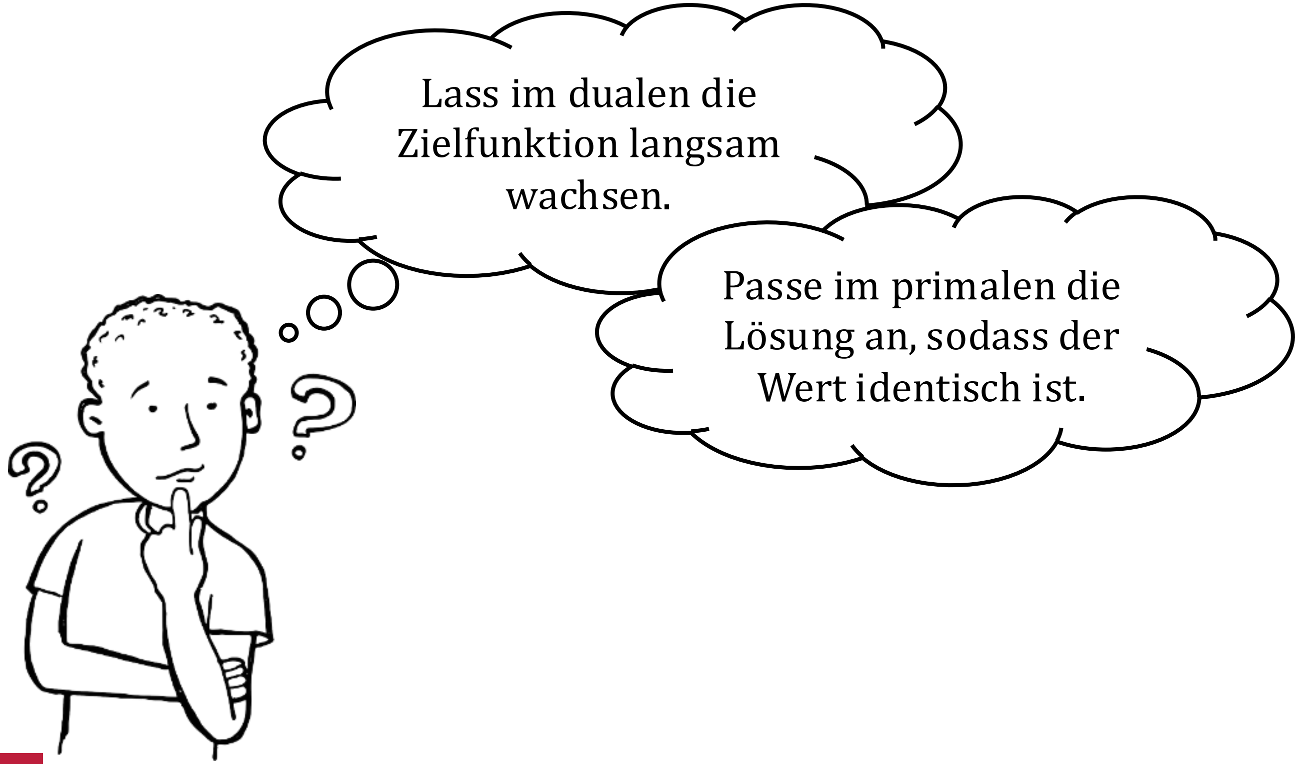
$$x_e = 1 \;\Rightarrow\; y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \mathrm{odd}(V)}} y_B = c(e)$$



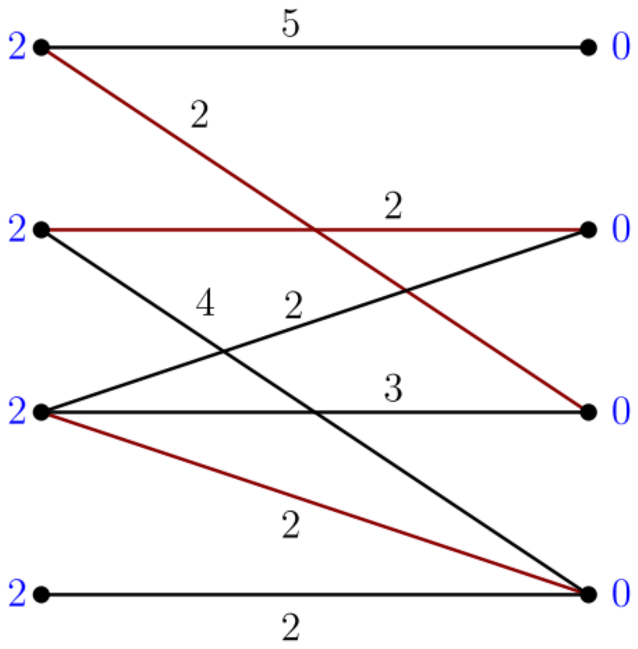
Kante kann nur eine Matchingkante sein, wenn sie durch Kreise überdeckt ist!

Primal-Dual-Algorithmen

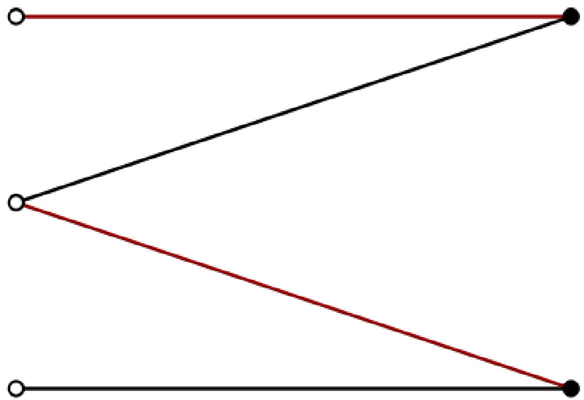
Idee


Lass im dualen die Zielfunktion langsam wachsen.
Passe im primalen die Lösung an, sodass der Wert identisch ist.

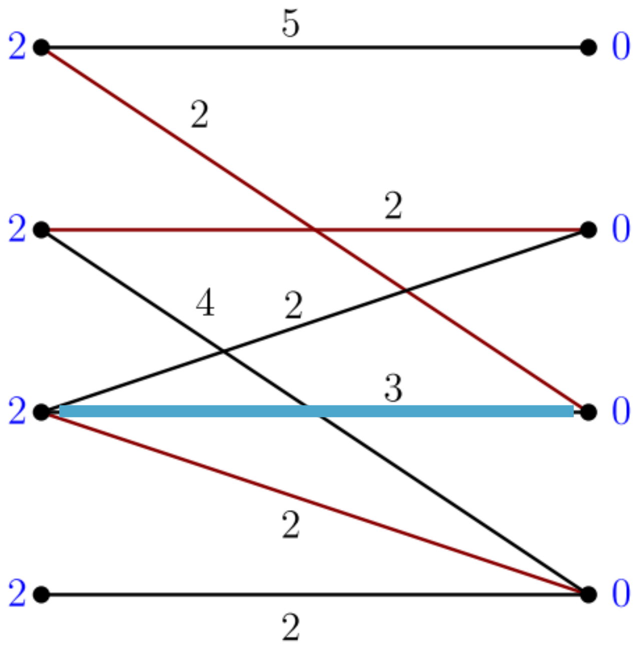
Finden von augmentierenden Pfaden



Konstruiere alternierenden Baum nur über überdeckte Kanten.



Aktualisieren der Dual-Werte



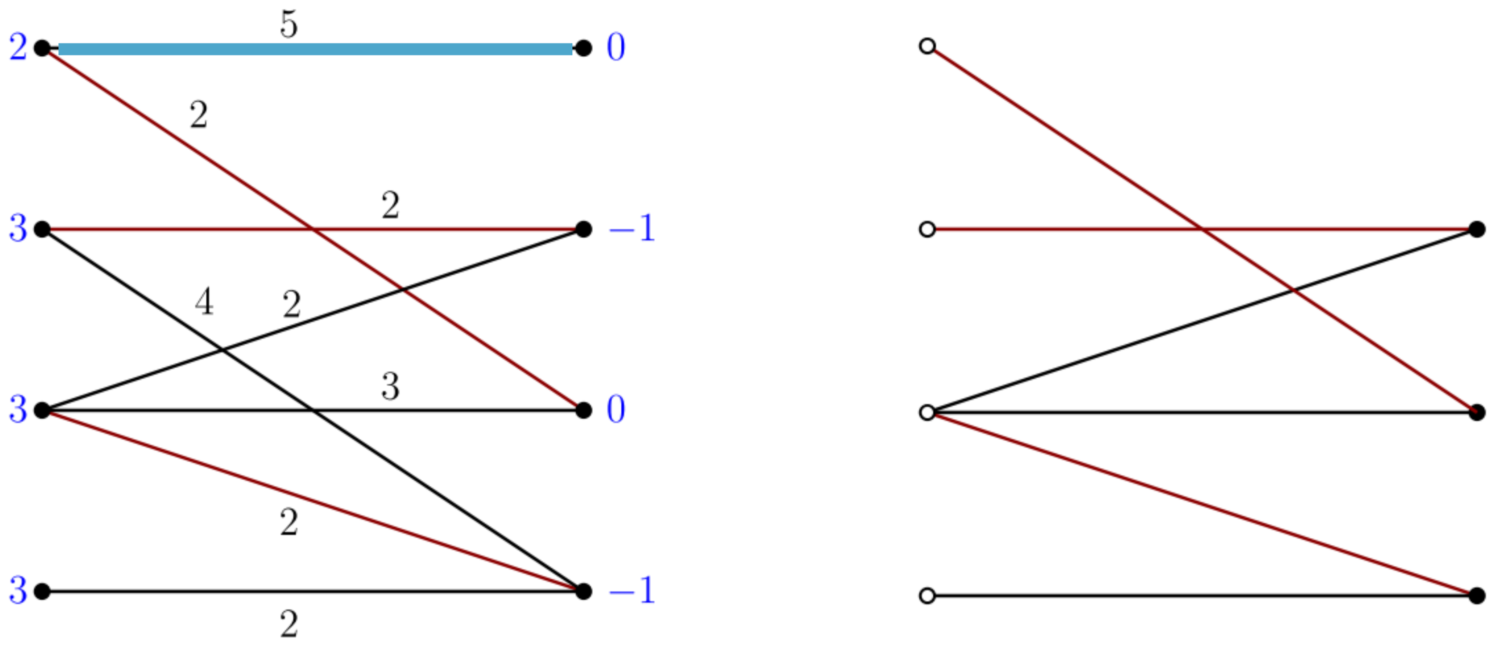
Suche kleinste reduzierte Kosten von Kanten, die herausführen.

$\Rightarrow 1$

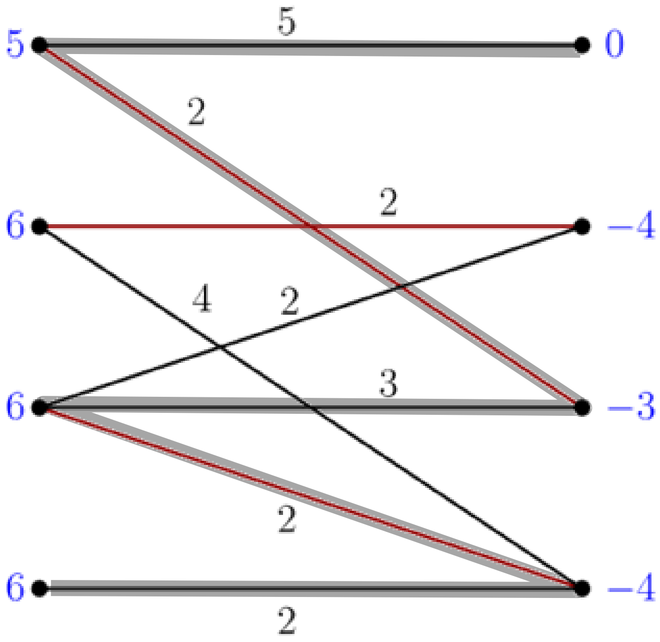
Erhöhe/Verringere Werte von Knoten um 1.

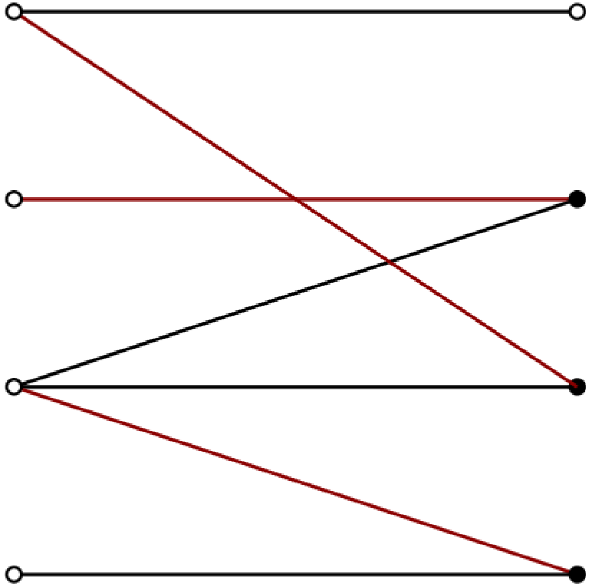
Aber wie?

Nächste Iteration



Nächste Iteration





Ungarische Methode

Algorithmus

Eingabe:

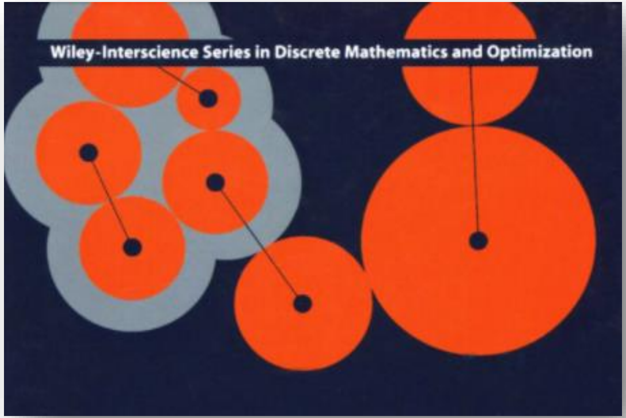
Bipartiter Graph $G = (V = V_1 \dot\cup V_2, E)$, Kantenkosten $c: E \to \mathbb{R}^+$

Ausgabe:

Min Cost Perfect Matching M

Function $\textsc{UngarischeMethode}(G, c)$
  Setze $y_v \coloneqq \min_{\{v,w\} \in E} c(\{v, w\})$ für alle $v \in V_1$.
  Finde maximales Matching M auf überdeckten Kanten.
  for $r \in V_1$ mit r ungematcht **do**
    Konstruiere alternierenden Baum T mit r als Start.
    while T enthält keinen augmentierenden Pfad **do**
      Sei $e \in E\big(W(T), V \setminus W(T)\big)$ eine Kante mit $\bar{c}(e)$ minimal.
      Setze $y_v \coloneqq y_v + \bar{c}(e)$ für alle $v \in W(T)$.
      Setze $y_w \coloneqq y_w - \bar{c}(e)$ für alle $w \in V(T) \setminus W(T)$.
      Erweitere T
    Augmentiere M
  return M

Ungerade Kreise



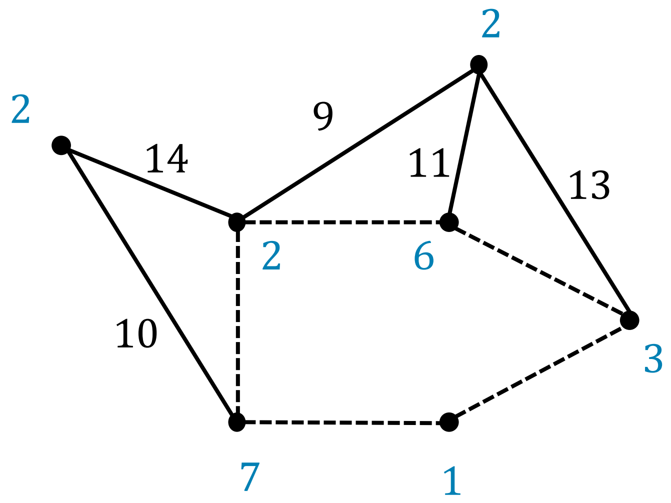
$$\max \sum_{v \in V} y_v + \sum_{B \in \mathrm{odd}(V)} y_B$$

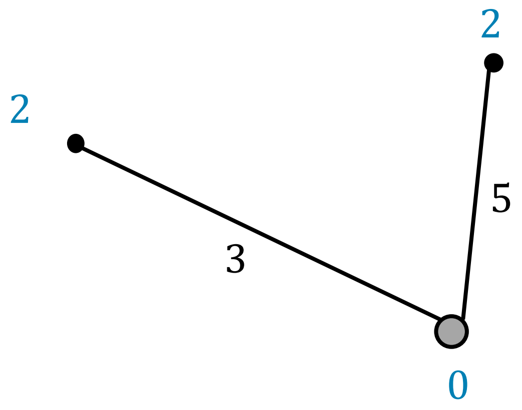
s. t.

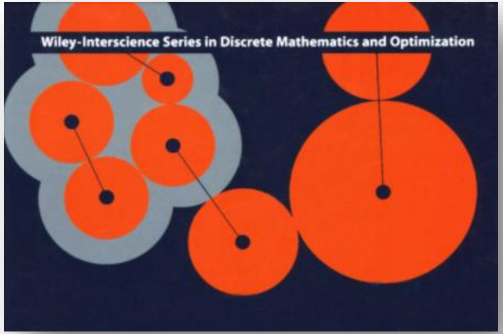
$$y_v + y_w + \sum_{\substack{e \in E(B, V \setminus B) \\ B \in \mathrm{odd}(V)}} y_B \leq c(e), \qquad \forall e \in E$$

$$y_B \geq 0, \qquad \forall B \in \mathrm{odd}(V)$$

Schrumpfen einer Blüte

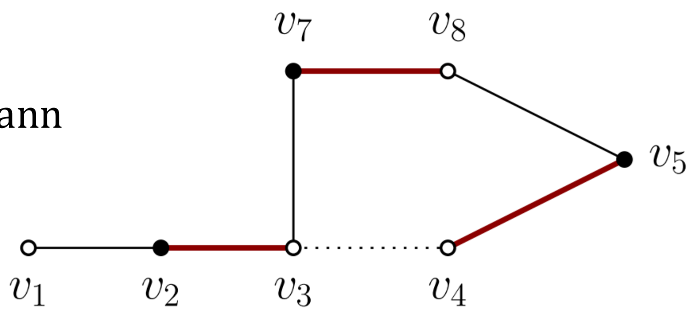






Kanten teilweise über Knoten im Kreis überdeckt.

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.

$$y_B \geq 0, \qquad \forall B \in \mathrm{odd}(V)$$

Updateregeln

Betrachte:

1. $\varepsilon_1 := \min\left\{\bar{c}(e) \mid e \in E\big(W(T), V \setminus W(T)\big)\right\}$
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 ε und verringere Knoten aus $S(T)$ um ε.

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 $c: E \to \mathbb{R}^+$

Ausgabe:

Min Cost Perfect Matching M

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

Primal-Duale-Algorithmen

Generell:
- Halte eine gültige primale Lösung mit einer unzulässigen dual-Lösung gleichen Wertes (oder andersherum)
- Passe primale Lösung an, sodass die zugehörige, unzulässige dual-Lösung näher an eine gültige Lösung kommt.

Weitere Beispiele:
- "Ford-Fulkerson" (Maximale Flüsse).
- "Dijkstra" (Kürzeste Wege).

Primal-Duale-Algorithmen eignen sich auch für Approximationen.