

# Algorithmische Graphentheorie

Julian Schubert

26. Mai 2021

## Inhaltsverzeichnis

<b>1</b>	<b>Wichtige Begriffe</b>	<b>2</b>
<b>2</b>	<b>Eulerkreise</b>	<b>2</b>
2.1	Eulerkreis finden . . . . .	2
<b>3</b>	<b>Hamiltonkreise</b>	<b>3</b>
<b>4</b>	<b>Handlungsreisen (TSP)</b>	<b>3</b>
<b>5</b>	<b>Lineare Programmierung</b>	<b>3</b>
<b>6</b>	<b>Flussalgorithmen</b>	<b>4</b>
6.1	Flussvergrößernde Wege . . . . .	4
6.2	Algorithmen . . . . .	5
<b>7</b>	<b>Matchings</b>	<b>5</b>
<b>8</b>	<b>Alternierende und augmentierende Wege</b>	<b>7</b>
<b>9</b>	<b>Wurzelspannbäume</b>	<b>8</b>

## 1 Wichtige Begriffe

### Definition 1

Ein gerichteter Graph  $G$  ist **schwach** zusammenhängend wenn der darunterliegende ungerichtete Graph zusammenhängend ist  
 Ein gerichteter Graph  $G$  ist **stark** zusammenhängend wenn es für jedes Knotenpaar  $(u, v)$  einen gerichteten Weg von  $u$  nach  $v$  gibt

### Definition 2: bipartiter Graph

Ein Graph  $G$  wird als bipartit bezeichnet, wenn sich seine Knoten in zwei disjunkte Teilmengen  $A$  und  $B$  aufteilen lassen. Zwischen den Knoten innerhalb dieser Teilmengen dürfen also keine Kanten existieren.

## 2 Eulerkreise

### Definition 3: Eulerkreis

Sei  $G$  ein (un-)gerichteter Graph.  
 Ein Eulerkreis (-weg) in  $G$  ist ein Kreis (Weg), der jede **Kante** genau einmal durchläuft.  
 Ein Graph heißt **eulersch**, falls er einen Eulerkreis enthält

Ein Graph der nur einen Eulerweg aber keinen Eulerkreis enthält, ist nicht eulersch!

### Eigenschaft 1: Satz von Euler

Sei  $G$  ein ungerichteter und zsh. Graph.  
 Dann gilt:  $G$  eulersch  $\Leftrightarrow$  alle Knoten haben geraden Grad

Bei gerichteten Graphen:  $\text{indeg}(v) = \text{outdeg}(v)$

### 2.1 Eulerkreis finden

Man kann in  $O(E)$  testen ob  $G$  eulersch ist (Knotengrade zählen)

Eulerkreis finden:

Verwalte in jedem Knoten  $v$  einen Zeiger  $\text{curr}[v]$ , der auf den ersten unbenutzten Nachbarn  $w$  zeigt

### 3 Hamiltonkreise

**Definition 4: Hamiltonkreis NP-schwer**

Sei  $G$  ein (un-)gerichteter Graph. Ein Hamiltonkreis (-weg) in  $G$  ist ein Kreis (Weg), der jeden **Knoten** genau einmal durchläuft.

**Eigenschaft 2: Satz von Bondy und Chvátal**

Sei  $G = (V, E)$  ein ungerichteter Graph mit  $|V| \geq 3$   
 Seien  $u$  und  $v$  nicht-adjazente Knoten von  $G$  mit  $\deg(u) + \deg(v) \geq n := |V|$ . Dann gilt:  
 $G$  hamiltons  $\Leftrightarrow G + uv$  hamiltons

**Eigenschaft 3: Satz von Dirac**

Sei  $G = (V, E)$  ein ungerichteter Graph mit  $|V| \geq 3$ . Falls jeder Knoten von  $G$   $\text{Grad} \geq |V| / 2$  hat, so ist  $G$  hamiltons

TODO: Beweisen

### 4 Handlungsreisen (TSP)

Lösbar mit Algorithmus von Bellman & Held-Karp

### 5 Lineare Programmierung

**Definition 5: Knotenüberdeckung**

**Gegeben:** Graph  $G = (V, E)$   
**Gesucht:** Knotenüberdeckung, d.h.  $V' \subseteq V$ , so dass jede Kante mindestens einen Endpunkt in  $V'$  hat.  
**Ziel:**  $|V'|$  minimal

**Definition 6: Clique**

**Gegeben:** ungerichteter, ungewichteter Graph  $G = (V, E)$   
**Gesucht:** Clique in  $G$   
 d.h.  $V' \subseteq V$ , so dass der von  $V'$  induzierte Graph  $G[V']$  vollständig ist

(also jeder Knoten eine Verbindung zu jedem anderen Knoten hat)  
 Mit anderen Worten:  $V' \subseteq V$ , so dass für alle  $\{u', v'\} \in \binom{V'}{2}$  gilt  $u'v' \in E$

### Definition 7: Fluss

Sei  $G = (V, E)$  ein gerichteter Graph mit  $s, t \in V$ . Eine Funktion  $f : E \rightarrow \mathbb{R}_{\geq 0}$  heißt s-t-Fluss (Fluss), wenn für jeden Knoten  $v \in V \setminus \{s, t\}$  gilt:

$$\sum_{u \in V \mid uv \in E} f(uv) - \sum_{w \in V \mid vw \in E} f(vw) = 0$$

Zufluss zum Knoten  $v$  = Abfluss vom Knoten  $v$ , also der Nettozufluss muss gleich Null sein.

### Definition 8

Sei  $G = (V, E)$  ein gerichteter Graph mit  $s, t \in V$ .  
 Seien durch  $c : E \rightarrow \mathbb{R}_{\geq 0}$  Kantenkapazitäten gegeben. Ein Fluss  $f$  ist zulässig, wenn für jede Kante  $e \in E$  gilt:

$$0 \leq f(e) \leq c(e)$$

Der **Wert**  $|f|$  eines Flusses  $f$  ist der Nettozufluss zum Knoten  $t$ .

## 6 Flussalgorithmen

### Definition 9: Kapazität eines Schnittes

$G$  Graph mit Kap.  $c : E \rightarrow \mathbb{R}_{> 0}$ ,  $(S, T)$  s-t-Schnitt.  
 Dann ist  $c(S) := c(\text{Raus}(S))$  die Kapazität von  $(S, T)$

### 6.1 Flussvergrößernde Wege

1. Residualgraph  $G'$  bilden:

- Hinrichtung: Benutzte Kapazität in  $G$
- Rückrichtung: Übrige Kapazität der Kante

**Definition 10**

Eins s-t-Weg  $W$  in  $G_f$  heißt flussvergrößernder Weg für  $f$ .  
Die Residualkapazität von  $W$  ist

$$\Delta_W := \min_{e \in W} c_f(e)$$

Ein zulässiger s-t-Fluss in  $G$  ist maximal  $\Leftrightarrow$  es gibt keinen Flussvergrößernden Weg in  $G_f$

**Definition 11: Max-Flow-Min-Cut-Theorem**

Sei  $f$  ein zulässiger s-t-Fluss in einem gerichteten Graphen  $G$  mit Kapazitäten  $c : E \rightarrow \mathbb{R}_{\geq 0}$

Dann sind folgende Bedingungen äquivalent:

1.  $f$  ist ein maximaler Fluss in  $G$
2.  $G_f$  enthält keine augmentierenden Wege
3. Es gibt einen s-t-Schnitt  $(S, T)$  mit  $|f| = c(S)$

Kurz

$$\max_{f \text{ zulässiger s-t-Fluss}} |f| = \min_{(S, T) \text{ s-t-Schnitt}} c(S)$$

**6.2 Algorithmen****Definition 12: FordFulkerson / EdmondsKarp**

Suche s-t-weg in  $G_f$  und füge das dann den Kanten hinzu.  
Änderung von EdmondsKarp: Muss der Kürzeste s-t-Weg sein

EdmondsKarp führt  $O(VE)$  Flussvergrößerungen durch  
EdmondsKarp läuft in  $O(VE^2)$

**7 Matchings****Definition 13: Matchings**

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

$M \subseteq E$  ist eine **Paarung** (engl. matching), wenn je zwei Kanten in  $M$  keinen gleichen Endpunkt haben  
 Falls für jede Kante  $e \in E$  gilt, dass  $M \cup \{e\}$  keine Paarung ist, so ist  $M$  **nicht erweiterbar** (engl. maximal)  
 Falls für alle Paarungen  $M'$  in  $G$  gilt, dass  $|M'| \leq |M|$ , so ist  $M$  eine **größte Paarung** (engl. maximum)  
 Falls jeder Knoten in  $G$  durch  $M$  gepaart ist, so ist  $M$  eine **perfekte Paarung** (engl. perfect matching)

#### Definition 14: Ganzzahligkeitssatz

Sind alle Kapazitäten ganzzahlig, d.h.  $c : E \rightarrow \mathbb{N}$ , so existiert ein maximaler Fluss, der ganzzahlig ist.

#### Eigenschaft 4: Satz von Menger

Sei  $G = (V, E)$  ein gerichteter Graph und  $s, t \in V$ . Dann ist die maximale Anzahl kantendisjunkter  $s$ - $t$ -Wege gleich der minimalen Kardinalität eines  $s$ - $t$ -Schnittes

**Kardinalität eines  $s$ - $t$ -Schnittes:** Anzahl an Kanten die von  $s$  nach  $t$  laufen.

$\Rightarrow$  minimale Kardinalität eines  $s$ - $t$ -Schnitts = maximale Anzahl an kantendisjunkter  $s$ - $t$ -Wege (die Kapazität aller möglichen  $s$ - $t$ -Schnitte ist genau so groß wie die Anzahl an möglichen  $s$ - $t$ -Wegen)

#### Eigenschaft 5: Auch von Menger

Sei  $G = (V, E)$  ein gerichteter Graph,  $s, t \in V, st \notin E$ . Dann ist die maximale Anzahl **knotendisjunkter**  $s$ - $t$ -Wege gleich der Kardinalität einer kleinsten Knotenmenge, die  $s$  und  $t$  trennt.

#### Definition 15: Nachbarschaft

Nachbarschaft von  $v \in V$  ist

$$N(v) := \{u \in V \mid uv \in E\}$$

Nachbarschaft von  $V' \subseteq V$  ist

$$N(V') := \bigcup_{v' \in V'} N(v')$$

**Definition 16: Heiratssatz (bewiesen von Philip Hall)**

Es existiert ein perfektes Matching  $\Leftrightarrow$  Für jedes  $D' \subseteq D$  gilt:  $|D'| \leq |N(D')|$

**Eigenschaft 6**

Sei  $G = (V, E)$  ein bipartiter Graph  
Dann lässt sich eine größte Parrung in  $G$  in  $O(VE^2)$  Zeit bestimmen

In  $G'$  können wir  $|V|$  s-t-wege in je  $O(E)$  zeit berechnen

## 8 Alternierende und augmentierende Wege

**Definition 17: Augmentierender Weg**

Ein Weg ist **augmentierend**, wenn die Kanten immer Abwechselnd im Matching und nicht im Matching liegen. Starten und Enden mit einer Kante die nicht im Matching liegt.

**Alternierend:** Wechselt zwischen im Matching und nicht im Matching

**Definition 18: Satz von Berge**

Sei  $G = (V, E)$  Graph,  $M \subseteq E$  Matching in  $G$ .  
 $M$  ist ein größtes Matching in  $G \Leftrightarrow$  es gibt keinen  $M$ -augmentierenden Weg.

**Eigenschaft 7**

In einem bipartiten Graphen  $G = (V, E)$  lässt sich in  $O(VE)$  ein größtes Matching bestimmen

**Ansatz:** Knoten  $S$  erstellen mit Kante zu allen Knoten im einen Teil, dann BFS  $|V|/2$  mal ausführen (oder bis kein freier Knoten in  $B$  mehr gefunden wird).

**Definition 19: Christofides Algorithmus**

- Ermittle einen minimalen Spannbaum  $B$  für  $G$
  - Sei  $U$  die Menge der Knoten ungeraden Grades in  $B$
  - Ermittle kostenminimales perfektes Matching  $M$  für  $G[U]$ 
    - $G[U]$  ist der von  $U$  induzierte Graph
    - $(U, \{vw \in E(G) : v \in U, w \in U\})$
  - Berechne im eulerschen Graphen  $B \cup M$  erst Eulertour und dann Rundtour  $T$  wie beim Tree-Doubling
- $\Rightarrow$  liefert eine  $3/2$ -Approximation für  $\Delta$ -TSP

**Definition 20: Kostenminimales perfektes Matching**

Gegeben: vollständiger Graph  $G = (V, E)$ , mit Kantenkosten  $c : E \rightarrow \mathbb{R}_{\geq 0}$   
 Gesucht: Perfektes Matching  $M$  mit minimalen Kosten  $c(M) = \sum_{e \in M} c(e)$   
 $\Rightarrow$  kann in  $O(V^3)$  berechnet werden (ist aber ziemlich kompliziert :)

## 9 Wurzelspannbäume

**Definition 21: Wurzelbaum**

Ein gerichteter Graph  $T = (V, E)$  mit Knoten  $s \in V$  heißt **s-Wurzelbaum**, wenn

- $T$  azyklisch
- $\text{indeg}(s) = 0$
- $\text{indeg}(v) = 1$  für jeden Knoten  $v \in V \setminus \{s\}$

**Definition 22: Wurzelspannbaum**

Sei  $G = (V, E)$  ein gerichteter (Multi-) Graph mit Knoten  $s \in V$ . Ein Teilgraph  $T$  von  $G$  mit Knotenmenge  $V$  heißt **s-Wurzelspannbaum** von  $G$ , wenn  $T$  ein  $s$ -Wurzelbaum ist.



**Eigenschaft 8**

Sei  $G$  ein gerichteter (Multi-) Graph mit Knoten  $s$   
 $G$  besitzt einen  $s$ -Wurzelspannbaum  $\Leftrightarrow$  jeder Knoten  $v \in V$  ist von  $s$  in  $G$  erreichbar.  
 $\text{DFS}(s)$  liefert  $s$ -Wurzelspannbaum (falls es einen gibt)

**Eigenschaft 9**

Sei  $K$  Kreis in  $F$  und  $\tilde{T}$   $s$ -Wurzelspannbaum von  $G/K$ .  
 Dann gibt es einen  $s$ -Wurzelspannbaum  $T$  von  $G$  mit

$$c'(T) \leq c'(\tilde{T})$$

$G/K$ :  $K$  sei Teilmenge von  $G$ . Alle Knoten in  $K$  werden durch einen einzigen Ersetzt.

Algorithmus zur berechnung von  $s$ -Wurzelspannbäumen:

- Berechne modifizierte Kantenkosten  $c'$
- Bestimme Teilgraph  $F$
- Falls  $F$  azyklisch, gib  $F$  zurück
- Ansonsten ermittle Kreis  $K$  in  $F$
- Kontrahiere  $G$  zu  $G / K$
- Wende Algo rekursiv auf  $(G/K, c')$  and
  - $s$ -Wurzelspannbaum für  $\tilde{T}$  für  $G/K$
- Expandiere  $\tilde{T}$  zu  $s$ -Wurzelspannbaum  $T$  von  $G$
- Gib  $T$  zurück