

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wie ist ein ungerichteter Graph formal definiert?

Ein *ungerichteter Graph*  $G$  ist ein Tripel  $(V, E, \Psi)$ , für das

1.  $V$  und  $E$  endliche Mengen sind und
2.  $\Psi : E \rightarrow \{X \subseteq V \mid 1 \leq |X| \leq 2\}$  ist.

$V$  ist die Knotenmenge,  $E$  ist die Kantenmenge.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein induzierter/aufspannender Teilgraph?

Ein *Teilgraph*  $H = (V(H), E(H))$  eines Graphen  $G = (V(G), E(G))$  ist ein Graph mit  $V(H) \subseteq V(G)$  und  $E(H) \subseteq E(G)$ .

$H$  ist ein *induzierter Teilgraph* von  $G$ , wenn  $E(H) = \{\{v, w\} \in E(G) \mid v, w \in V(H)\}$ .

$H$  ist *aufspannend*, wenn  $V(H) = V(G)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wann sind zwei Graphen isomorph?

Zwei Graphen  $G$  und  $H$  sind *isomorph*, wenn es Bijektionen

$$\Phi_V : V(G) \rightarrow V(H) \text{ und } \Phi_E : E(G) \rightarrow E(H)$$

gibt, mit

$$\Phi_E(\{v, w\}) = \{\Phi_V(v), \Phi_V(w)\}.$$

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

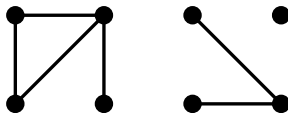
---

Was ist ein regulärer/vollständiger Graph?

Was ist das Komplement eines Graphen?

Ein Graph heißt *regulär*, wenn alle Knoten gleichen Grad haben und *vollständig*, wenn alle Knoten paarweise adjazent sind.

Das *Komplement* eines Graphen  $G = (V, E)$  mit  $|V| = n$  ist der Graph  $\bar{G} = (V, \bar{E})$  mit  $\bar{E} = E(K_n) \setminus E$ .



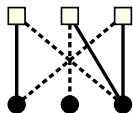
# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was versteht man unter einem Matching, einem Vertex Cover, einer unabhängige Menge und einer Clique?

Ein *Matching* ist eine Menge knotendisjunkter Kanten. Ein *Vertex Cover* ( $\square$ ) ist eine kantenüberdeckende Knotenmenge.



Eine *unabhängige (stabile) Menge* ist eine Menge paarweise nicht adjazenter Kanten. Ein *Clique* ist eine Menge paarweise adjazenter Knoten.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche besonderen und wichtigen Kantenfolgen kann es in einem Graphen geben?

- Eine *Kantenfolge*  $W$  in  $G$  ist eine Folge  $v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$  mit  $k \geq 0$  und  $e_i = \{v_i, v_{i+1}\}$ .
- Ein *Weg* ist eine Kantenfolge, in der sich keine Kante wiederholt.
- Ein *Pfad* ist eine Kantenfolge, in der sich kein Knoten wiederholt.
- Ein *s-t-Pfad* ist ein Pfad mit Anfangsknoten  $s$  und Endknoten  $t$ .
- Ein *Kreis* ist ein geschlossener Weg, in dem sich außer Anfangs- und Endknoten kein Knoten wiederholt.
- Die *Länge* eines Pfades oder Kreises ist die Anzahl der Kanten.
- Ein *Hamiltonpfad* (-kreis) ist ein aufspannender Pfad (Kreis), also der Länge  $n - 1$  ( $n$ ).

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was kann man über den Zusammenhang von ungerichteten Graphen sagen?

- Ein ungerichteter Graph  $G$  ist *zusammenhängend*, wenn es für je zwei Knoten  $v, w \in V(G)$  immer einen  $v$ - $w$ -Pfad gibt.
- Die maximalen zusammenhängenden Teilgraphen von  $G$  nennt man *Zusammenhangskomponenten*.
- Ein Knoten mit der Eigenschaft, dass  $G - v$  mehr Zusammenhangskomponenten als  $G$  hat, nennt man *Schnittknoten* (*Artikulationsknoten*).
- Ein Kante ist eine *Schnittkante* (*Brücke*), wenn Entfernen der Kante die Zahl der Zusammenhangskomponenten erhöht.
- Ein ungerichteter Graph ohne Kreis ist ein *Wald*.
- Ein zusammenhängender Wald ist ein *Baum*.
- Ein Knoten vom Grad 1 ist ein *Blatt*.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was kann man über den Zusammenhang von gerichteten Graphen sagen?

- Ein gerichteter Graph  $G$  ist *zusammenhängend*, wenn der zugrundeliegende ungerichtete Graph zusammenhängend ist.
- Ein gerichteter Graph  $G$  ist *stark zusammenhängend*, wenn für je zwei Knoten  $v, w$  sowohl ein (gerichteter)  $v$ - $w$ -Pfad, als auch ein  $w$ - $v$ -Pfad existiert.
- Die *starken Zusammenhangskomponenten* sind die maximalen stark zusammenhängenden Teilgraphen.
- Ein gerichteter Graph ist ein *Branching*, wenn der zugrundeliegende ungerichtete Graph ein Wald ist, und jeder ungerichtete Knoten nur eine eingehende Kante hat.
- Ein zusammenhängendes Branching ist eine *Arboreszenz*.
- Der eindeutige Knoten in einer Arboreszenz mit Eingrad 0 ist die Wurzel, die Knoten mit Ausgrad 0 sind die *Blätter*.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein (gerichteter) Schnitt?

Ein *Schnitt* in einem ungerichteten Graphen ist eine Kantenmenge vom Typ  $\delta(X)$  für  $\emptyset \neq X \subset V(G)$ , die also  $X$  und  $V \setminus X$  trennt.

Ein *gerichteter Schnitt* ist eine Kantenmenge vom Typ  $\delta^+(X)$  für  $\emptyset \neq X \subset V(G)$  mit  $\delta^-(X) = \emptyset$ .

Eine Kantenmenge  $F \subseteq E(G)$  *trennt* zwei Knoten  $s$  und  $t$ , wenn  $t$  von  $s$  in  $G$  erreichbar ist, aber nicht in  $(V(G), E(G) \setminus F)$ .

In einem Digraphen ist eine Kantenmenge vom Typ  $\delta^+(X)$  mit  $s \in X$  und  $t \notin X$  ein (gerichteter) *s-t-Schnitt*.

Analog ist in ungerichteten Graphen ein *s-t-Schnitt* eine trennende Kantenmenge der Form  $\delta(X)$ , die  $s$  und  $t$  trennt.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist eine topologische Ordnung und wann existiert sie?

Eine *topologische Ordnung* von einem Digraphen  $G$  ist eine Ordnung  $\{v_1, \dots, v_n\}$  auf den Knoten von  $G$  so, dass für jede Kante  $(v_i, v_j) \in E(G)$  gilt  $i < j$ .

Ein Digraph  $G$  hat eine topologische Ordnung  $\iff G$  ist azyklisch.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein (Ko-)Kreisraum? Was ist eine (Ko-)Kreisbasis?

Der Untervektorraum des  $\mathbb{R}^{E(G)}$ , der von Kreisen (Schnitten) aufgespannt wird, heißt *(Ko-)Kreisraum*.

Eine Menge von ungerichteten Kreisen (Schnitten) heißt *(Ko-)Kreisbasis*, wenn die zugehörigen Vektoren eine Basis des *(Ko-)Kreisraums* bilden.

Sei  $G$  ein Digraph und  $T$  ein maximaler Teilgraph ohne ungerichteten Kreis.

- Für jedes  $e \in E(G) \setminus E(T)$  ist der eindeutig ungerichtete Kreis in  $T + e$  der *Fundamentalkreis von  $e$  bzgl.  $T$* .
- Für jedes  $e \in E(T)$  gibt es eine Menge  $X \subseteq V(G)$  mit  $\delta_G(X) \cap E(T) = \{e\}$ . Wir nennen  $\delta_G(X)$  einen *Fundamentalschnitt von  $e$  bzgl.  $T$* .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Graphen-Scan-Algorithmus“?

**Input:** Graph  $G$ . Knoten  $S$ .

**Output:** Knotenmenge  $R \subseteq V(G)$ , die von  $s$  aus erreichbar sind. Kantenmenge  $T \subseteq E(G)$ , die die Erreichbarkeit von  $R$  sicherstellen.

Im gerichteten Fall ist  $(R, T)$  eine in  $s$  verwurzelte Arboreszenz, im ungerichteten Fall ein aufspannender Baum.

**Komplexität:**  $\mathcal{O}(n + m)$ , also linear.

**Varianten:** Breitensuche (BFS), Tiefensuch (DFS).

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Worin liegt der Unterschied zwischen Breiten- und Tiefensuche?

**Breitensuche (BFS):** Die zuerst aufgenommenen Knoten werden zuerst abgearbeitet (Warteschlange, Queue, „FIFO“-Regel), d.h. die Suche geht erst einmal in die Breite, bevor weitere Knoten betrachtet werden.

Breitensuche liefert kürzeste Wege.

**Tiefensuche (DFS):** Die zuletzt aufgenommenen Knoten werden zuerst abgearbeitet (Stapel, Stack, „LIFO“-Regel), d.h. die Suche geht erst einmal in die Tiefe, bevor weitere Knoten betrachtet werden.

Breitensuche garantiert, ein verstecktes Objekt in bester Worst-Case-Zeit zu finden.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche Datenstrukturen für Graphen unterscheidet man?

**Inzidenzmatrix:** Sei  $A \in \{0, 1\}^{n \times m}$  mit  $a_{v,e} = \begin{cases} 1 & \text{für } v \in e \\ 0 & \text{sonst} \end{cases}$ ,

bzw.  $A \in \{0, 1, -1\}^{n \times m}$  mit  $a_{v,(x,y)} = \begin{cases} 1 & \text{für } v = x \\ -1 & \text{für } v = y \\ 0 & \text{sonst} \end{cases}$ . Größe:  $\mathcal{O}(mn)$ .

**Adjazenzmatrix:** Sei  $A \in \{0, 1\}^{n \times n}$  mit  $a_{v,w} = \begin{cases} 1 & \text{für } \{v, w\} \in E \\ 0 & \text{sonst} \end{cases}$ ,

bzw.  $A \in \{0, 1, -1\}^{n \times n}$  mit  $a_{v,w} = \begin{cases} 1 & \text{für } (v, w) \in E \\ -1 & \text{für } (w, v) \in E \\ 0 & \text{sonst} \end{cases}$ .

Größe:  $\mathcal{O}(n^2)$ .

**Kantenliste** Größe:  $\mathcal{O}(m \log n)$ .

**Adjazenzliste:** Größe:  $\mathcal{O}(n \log m + m \log n)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Aufspannende Bäume und Arboreszenzen“ durchgenommen und was liefern sie genau?

**Kruskal-Algorithmus:** Aufspannender Baum  $T$  minimalen Gewichts.

**Prim-Algorithmus:** Aufspannender Baum  $T$  minimalen Gewichts.

**Edmonds Branching-Algorithmus:** Ein gewichtsmaximales Branching.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Kruskal-Algorithmus“?

**Input:** Zusammenhängender (ungerichteter) Graph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}$ .

**Output:** Aufspannender Baum  $T$  minimalen Gewichts (MST).

**Komplexität:**  $\mathcal{O}(m \cdot n)$ .

**Verbesserung:** Buchführung über Zusammenhangskomponenten. Kante nur zwischen unterschiedlichen Zusammenhangskomponenten zulässig. Dann  $\mathcal{O}(m \log n)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Prim-Algorithmus“?

**Input:** Zusammenhängender (ungerichteter) Graph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}$ .

**Output:** Aufspannender Baum  $T$  minimalen Gewichts (MST).

**Komplexität:**  $\mathcal{O}(n^2)$ .

**Verbesserung:**  $\mathcal{O}(n^2)$  ist bestmöglich für dichte Graphen.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert „Edmonds Branching-Algorithmus“?

**Input:** Digraph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}$ .

**Output:** Gewichtsmaximales Branching.

**Komplexität:**  $\mathcal{O}(nm)$ .

**Verbesserung:** Es existiert Algorithmus mit  $\mathcal{O}(m + n \log n)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Kürzeste Wege“ durchgenommen und was liefern sie genau?

**Dijkstra-Algorithmus:** Für jeden Knoten  $v \in V(G)$  die Länge eines kürzesten  $s$ - $v$ -Pfad und den Vorgänger von  $v$  in einem kürzesten  $s$ - $v$ -Pfad.

**Algorithmus von Moore-Bellman-Ford:** Für jeden Knoten  $v \in V(G)$  die Länge eines kürzesten  $s$ - $v$ -Pfad und den Vorgänger von  $v$  in einem kürzesten  $s$ - $v$ -Pfad.

**Algorithmus von Floyd, Warshall:** Für jedes Knotenpaar  $i, j \in V(G)$  die Länge eines kürzesten  $i$ - $j$ -Pfad und den Vorgänger von  $j$  in einem kürzesten  $i$ - $j$ -Pfad.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Dijkstra-Algorithmus“?

**Input:** Digraph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}_+$ .

Startknoten  $s \in V(G)$ .

**Output:** Für jeden Knoten  $v \in V(G)$  die Angaben

- $l(v)$ : Länge eines kürzesten  $s$ - $v$ -Pfades.
- $p(v)$ : Vorgänger von  $v$  in einem kürzesten  $s$ - $v$ -Pfad.

**Komplexität:**  $\mathcal{O}(n^2)$ .

**Verbesserung:** Wahl von geschickteren Datenstrukturen, z.B. Fibonacci-Heaps.  
Algorithmus von Moore-Bellman-Ford.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus von „Moore-Bellman-Ford“?

**Input:** Digraph  $G$ .

Konservative Kantengewichte  $c : E(G) \rightarrow \mathbb{R}$ .

Startknoten  $s \in V(G)$ .

**Output:** Für jeden Knoten  $v \in V(G)$  die Angaben

- $l(v)$ : Länge eines kürzesten  $s$ - $v$ -Pfades.
- $p(v)$ : Vorgänger von  $v$  in einem kürzesten  $s$ - $v$ -Pfad.

**Komplexität:**  $\mathcal{O}(nm)$ .

**Prinzip:** Dynamische Programmierung.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus von „Floyd, Warshall“?

**Input:** Digraph  $G$ .

Konservative Kantengewichte  $c : E(G) \rightarrow \mathbb{R}$ .

**Output:** Für jedes Knotenpaar  $i, j \in V(G)$  die Angaben

- $l_{ij}$ : Länge eines kürzesten  $i$ - $j$ -Pfad.
- $p_{ij}$ : Vorgänger von  $j$  in einem kürzesten  $i$ - $j$ -Pfad.

**Komplexität:**  $\mathcal{O}(n^3)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was versteht man unter einem Fluss und welche Definitionen gibt es dazu?

Gegeben sei ein Digraph  $G$  mit Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$ .

- Ein *Fluss* ist eine Funktion  $f : E(G) \rightarrow \mathbb{R}_+$  mit  $f(e) \leq u(e)$  für alle  $e \in E(G)$ .
- *Flusserhaltung* an einem Knoten  $v \in V(G)$  bedeutet, dass gilt

$$\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e).$$

- Eine *Zirkulation* ist ein Fluss, für den überall Flusserhaltung gilt.
- Für ein Netzwerk  $(G, u, s, t)$  mit  $s, t \in V(G)$  ist ein *s-t-Fluss* ein Fluss, für den überall außer in  $s$  und  $t$  Flusserhaltung gilt. Man nennt

$$\text{Wert}(f) = \sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e).$$

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wie sieht das LP zum „Maximaler Fluss“-Problem aus?

Ist  $F$  der Wert des Flusses, so lautet das LP

$$\max F$$

$$\sum_{e \in \delta^+(s)} f_e - \sum_{e \in \delta^-(s)} f_e = F$$

$$\sum_{e \in \delta^+(v)} f_e - \sum_{e \in \delta^-(v)} f_e = 0 \quad \forall v \in V(G) \setminus \{s, t\}$$

$$0 \leq f_e \leq u_e \quad \forall e \in E(G).$$

Da das LP zulässig ( $f_e = 0$ ) und durch  $u_e$  beschränkt ist, gibt es immer eine Lösung für das „Maximaler Fluss“-Problem.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein Residualgraph?

- Für einen Fluss  $f$  in einem Digraphen  $G$  mit Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$  definieren wir die *Residualkapazitäten*

$$u_f := \begin{cases} u(e) - f(e) & \text{für Vorwärtskanten} \\ f(e) & \text{für Rückwärtskanten} \end{cases} .$$

- Der *Residualgraph* ist der Graph

$$G_f := \left( V(G), \{e \in E(\overleftrightarrow{G}) \mid u_f(e) > 0\} \right) .$$

- Interpretation:
  - $u_f$  auf Vorwärtskanten beschreibt, um wieviel man  $f$  noch erhöhen kann.
  - $u_f$  auf Rückwärtskanten beschreibt, um wieviel man  $f$  noch verringern kann.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein  $f$ -augmentierender Pfad?

- Um für einen Fluss  $f$  und einen Pfad (oder Kreis)  $P$  in  $G_f$  den Fluss entlang  $P$  um  $\gamma$  zu *augmentieren* (verbessern), muss man den Fluss für Vorwärtskanten um  $\gamma$  erhöhen, für Rückwärtskanten um  $\gamma$  reduzieren.
- Ein *f*-*augmentierender Pfad* in einem Netzwerk  $(G, u, s, t)$  und einem Fluss  $f$  ist ein *s-t*-Pfad im Residualgraphen.
- Ein *s-t*-Fluss ist optimal  $\iff$  Es gibt keinen *f*-*augmentierenden Pfad*.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Netzwerkflüsse“ durchgenommen und was liefern sie genau?

**Algorithmus von Ford-Fulkerson:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Algorithmus von Edmonds-Karp:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Push-Relabel-Algorithmus:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus von „Ford-Fulkerson“?

**Input:** Netzwerk  $(G, u, s, t)$ .

**Output:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Komplexität:** Schlechte Laufzeit (abhängig von den Kapazitäten) und terminiert für irrationale Kapazitäten mitunter nicht, bzw. konvergiert dabei nicht unbedingt gegen den richtigen Wert.

**Verbesserungen:** Algorithmus von Edmonds-Karp durch geschickte Wahl der augmentierenden Pfade.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was besagt das „Max-Flow-Min-Cut“-Theorem?

In jedem Netzwerk  $(G, u, s, t,)$  ist der maximale Wert eines  $s$ - $t$ -Flusses gleich der minimalen Kapazität eines  $s$ - $t$ -Schnittes.

Will man also die Größe eines maximalen Flusses positiv oder negativ beeinflussen, dann sollte man an einem minimalen Schnitt die Kapazitäten verändern.

Wenn die Kapazitäten der Kanten ganzzahlig sind, dann existiert ein optimaler Fluss von ganzzahligem Wert.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus von „Edmonds-Karp“?

**Input:** Netzwerk  $(G, u, s, t)$ .

**Output:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Komplexität:**  $\mathcal{O}(m^2n)$ . Unabhängig von den Kapazitäten stoppt er nach  $\frac{mn}{2}$  Augmentierungen.

**Anmerkung:** Sucht *kürzeste*  $f$ -augmentierende Pfade und benutzt dafür BFS in  $G_f$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was versteht man unter einem Präfluss?

- In einem Netzwerk  $(G, u, s, t)$  heißt eine Funktion  $f : E(G) \rightarrow \mathbb{R}_+$  *Präfluss* (*Preflow*), wenn sie folgendes erfüllt:

1.  $f(e) \leq u(e)$ .

2. In jeden Knoten fließt mindestens so viel hinein wie heraus. Der *Exzess* (*Überschuss*) ist

$$\text{ex}_f(v) := \sum_{e \in \delta^-(v)} f(e) - \sum_{e \in \delta^+(v)} f(e) \geq 0.$$

- $v \in V(G) \setminus \{s, t\}$  ist *aktiv*, wenn  $\text{ex}_f(v) > 0$  ist.

Ein Präfluss ist ein Fluss, wenn er keine aktiven Knoten hat.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was versteht man unter einem Distanzlabeling?

- Sei  $(G, u, s, t)$  ein Netzwerk und  $f$  ein  $s$ - $t$ -Präfluss. Ein *Distanzlabeling* ist eine Funktion  $\Psi : V(G) \rightarrow \mathbb{Z}_+$  so, dass

$$\Psi(t) = 0, \quad \Psi(s) = n$$

und  $\Psi(v) \leq \Psi(w) + 1$  für alle  $(v, w) \in E(G_f)$

- Eine Kante  $e = (v, w) \in E(\overleftrightarrow{G})$  ist *zulässig*, falls  $e \in E(G_f)$  und  $\Psi(v) = \Psi(w) + 1$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Push-Relabel-Algorithmus“?

**Input:** Netzwerk  $(G, u, s, t)$ .

**Output:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Komplexität:** ?

**Anmerkung:** Während des Algorithmus ist  $f$  immer ein  $s$ - $t$ -Präfluss und  $\Psi$  immer ein Distanzlabeling bzgl.  $f$ .

**Erweiterung:** Algorithmus von Goldberg-Tarjan.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus von „Goldberg-Tarjan“?

**Input:** Netzwerk  $(G, u, s, t)$ .

**Output:** Ein  $s$ - $t$ -Fluss mit maximalem Wert.

**Komplexität:**  $\mathcal{O}(n^3)$

**Anmerkung:** Während des Algorithmus ist  $f$  immer ein  $s$ - $t$ -Präfluss und  $\Psi$  immer ein Distanzlabeling bzgl.  $f$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was ist ein „Gomory-Hu-Baum“?

- Sei  $G$  ein ungerichteter Graph und  $u : E(G) \rightarrow \mathbb{R}_+$  eine Kapazitätsfunktion. Für zwei Knoten  $s, t \in V(G)$  ist  $\lambda_{st}$  die *Kapazität* eines minimalen  $s$ - $t$ -Schnittes.
- Ein Baum  $T$  auf der Knotenmenge  $V$  eines ungerichteten Graphen  $G$  mit Kapazitäten ist ein *Gomory-Hu-Baum*, wenn

$$\lambda_{st} = \min_{e \in E(P_{st})} u(\delta_G(C_e))$$

für alle  $s, t \in V(G)$  ist, wobei  $P_{st}$  der eindeutige  $s$ - $t$ -Pfad in  $T$  und  $C_e$  und  $V(G) \setminus C_e$  die Zusammenhangskomponenten von  $(T - e)$  für  $e \in E(T)$  sind.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Kostenminimale Flüsse“ durchgenommen und was liefern sie genau?

**Cycle-Cancelling Algorithmus:** Ein kostenminimaler  $b$ -Fluss.

**Minimum-Mean-Cycle-Cancelling Algorithmus:** Ein kostenminimaler  $b$ -Fluss.

**Minimum-Mean-Cycle Algorithmus:** Ein Kreis  $C$  minimalen Durchschnittsgewichts.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was versteht man unter einem  $b$ -Fluss und welche Definitionen gibt es dazu?

- Gegeben sei ein Digraph  $G$  mit den Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$ . Weiterhin gibt es Zahlen  $b : V(G) \rightarrow \mathbb{R}$  mit  $\sum_{v \in V(G)} b(v) = 0$ , die das Angebot und die Nachfrage beschreiben.

Ein  $b$ -Fluss ist eine Funktion  $f : E(G) \rightarrow \mathbb{R}_+$  mit  $0 \leq f(e) \leq u(e)$  und

$$b(v) = \sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e)$$

für alle  $v \in V(G)$ .

- Ein  $b$ -Fluss mit  $b \equiv 0$  ist eine *Zirkulation* und  $b(v)$  ist die Bilanz vom Knoten  $v$ . Von  $v$  ist

$$|b(v)| \text{ auch } \begin{cases} \text{Angebot (Überschuss)} & \text{für } b(v) \geq 0 \\ \text{Nachfrage (Defizit)} & \text{für } b(v) \leq 0 \end{cases}$$

- Knoten mit  $b(v) > 0$  nennt man auch *Quellen*, mit  $b(v) < 0$  *Senken*.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wann ist ein  $b$ -Fluss optimal?

Sei  $(G, u, b, c)$  eine Instanz des „Kostenminimaler-Fluss“-Problems. Ein  $b$ -Fluss  $f$  ist genau dann optimal, wenn es keinen  $f$ -augmentierenden Kreis von negativem Gesamtgewicht gibt.

Ein  $b$ -Fluss  $f$  ist genau dann optimal für eine Instanz  $(G, u, b, c)$ , wenn es eine Potentialfunktion  $\pi : V(G) \rightarrow \mathbb{R}$  gibt, die für alle  $x, y \in V(G)$  mit  $(x, y) \in G_f$

$$\pi(x) + c((x, y)) \geq \pi(y)$$

ist.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Cycle-Cancelling-Algorithmus“?

**Input:** Digraph  $G$ .

Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$ .

Bilanzen  $b : V(G) \rightarrow \mathbb{R}$  mit  $\sum_{v \in V(G)} b(v) = 0$ .

**Output:** Ein kostenminimaler  $b$ -Fluss  $f$ .

**Komplexität:** Laufzeit kann schlecht sein.

**Verbesserung:** Kreis minimalen Durchschnittsgewicht (Minimum-Mean-Cycle-Cancelling Algorithmus).

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Minimum-Mean-Cycle-Cancelling-Algorithmus“?

**Input:** Digraph  $G$ .

Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$ .

Bilanzen  $b : V(G) \rightarrow \mathbb{R}$  mit  $\sum_{v \in V(G)} b(v) = 0$ .

**Output:** Ein kostenminimaler  $b$ -Fluss  $f$ .

**Komplexität:**  $\mathcal{O}(m^3 n^2 \log n)$ .

**Bemerkung:** Beinhaltet Minimum-Mean-Cycle Algorithmus.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Minimum-Mean-Cycle-Algorithmus“?

**Input:** Digraph  $G$ .

Kapazitäten  $u : E(G) \rightarrow \mathbb{R}_+$ .

**Output:** Ein Kreis  $C$  minimalen Durchschnittsgewichts.

**Komplexität:**  $\mathcal{O}(n(m + n))$ .

**Bemerkung:** Wird benötigt im Minimum-Mean-Cycle-Cancelling Algorithmus.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Maximale Matchings“ durchgenommen und was liefern sie genau?

**Algorithmus Maximales Matching:** Ein Matching  $M$  maximaler Kardinalität in bipartiten Graphen.

**Blossom-Algorithmus für perfektes Matching:** Ein perfektes Matching  $M$  oder Beleg, dass keines existiert.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wie lauten der Satz von König-Égerváry und der Heiratssatz von Frobenius?

**König-Égerváry:** In bipartiten Graphen gilt  $\nu(G) = \tau(G)$ , d.h. minimales Vertex-Cover und maximales Matching sind gleich.

**Heiratssatz:** Sei  $G$  ein bipartiter Graph mit Knotenpartitionen  $A$  und  $B$  mit  $V(G) = A \dot{\cup} B$ . Dann hat  $G$  ein perfektes Matching genau dann, wenn

$$|A| = |B|$$
$$\text{und } |\Gamma(X)| \geq |X| \quad \forall X \subseteq A$$

ist. Dabei ist  $\Gamma(X)$  die Menge der Nachbarn von  $X$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus für „Maximales Matching in bipartiten Graphen“?

**Input:** Ein zusammenhängender bipartiter Graph  $G$ .

**Output:** Ein Matching  $M$  maximaler Kardinalität (falls möglich, perfekt).

**Komplexität:** ?

**Bemerkung:** Geht nur in bipartiten Graphen.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Wann hat ein Graph ein perfektes Matching?

Ein Graph  $G = (V, E)$  hat genau dann ein perfektes Matching, wenn für jede Menge  $A \subseteq V$

$$\text{oc}(G \setminus A) \leq |A|$$

gilt.  $\text{oc}(X)$  gibt hierbei die Zahl der ungeraden Komponenten von  $X$  an.

Für  $G = (V, E)$  gilt

$$\max\{|M| \mid M \text{ ist Matching}\} = \min\left\{\frac{1}{2}|V| - \text{oc}(G \setminus A) + |A| \mid A \subseteq V\right\}.$$

Demnach liefert ein Menge  $A$  eine obere Schranke für die Kardinalität eines minimalen Matchings.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Blossom-Algorithmus für perfektes Matching“?

**Input:** Ein zusammenhängender Graph  $G$ .

**Output:** Ein perfektes Matching  $M$  oder ein Beleg, dass keines existiert.

**Komplexität:**  $\mathcal{O}(nm \log n)$ .

**Erweiterung:** Lässt sich leicht anpassen, um ein maximales Matching zu finden.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Welche unterschiedlichen Algorithmen wurden für das Problem „Gewichtsminimales Matchings“ durchgenommen und was liefern sie genau?

**Algorithmus für gewichtsminales perfektes Matching:** Ein perfektes Matching  $M$  minimalen Gewichts in bipartiten Graphen oder Beleg, dass keines existiert.

**Blossom-Algorithmus für gewichtsminales perfektes Matching:** Ein gewichtsminales perfektes Matching  $M$  oder Beleg, dass keines existiert.

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der Algorithmus für „Gewichtsm minimales perfektes Matching in bipartiten Graphen“?

**Input:** Ein zusammenhängender bipartiter Graph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}_+$ .

**Output:** Ein perfektes Matching  $M$  minimalen Gewichts samt Optimalitätsbeleg durch passende Dualvariable,  
oder ein Beleg, dass  $G$  kein perfektes Matching hat.

**Komplexität:**  $\mathcal{O}(n^2m)$ .

# DISKRETE OPTIMIERUNG

*Prof. Dr. Fekete (SS 2003)*

---

Was liefert der „Blossom-Algorithmus für gewichtsminimales Matching“?

**Input:** Ein zusammenhängender Graph  $G$ .

Kantengewichte  $c : E(G) \rightarrow \mathbb{R}_+$ .

**Output:** Ein perfektes Matching  $M$  minimalen Gewichts samt Optimalitätsbeleg durch passende Dualvariable,  
oder ein Beleg, dass  $G$  kein perfektes Matching hat.

**Komplexität:** Polynomiell.