# 1 Matching

## 1.1 Matching-Lemma

G Graph, $w : E \to \mathbb{R}$, $v \in V$, $M'$ maximales Matching für $G' = G - v$; dann kann mit einer Berechnung eines erhöhenden Weges Matching $M$ maximalen Gewichts von $G$ berechnet werden.

## 1.2 Matching-Algorithmus für planaren Graphen $G$

1. Zerlege $G$ in $G_1, G_2$ dank Separator $S$ entsprechend Planar-Separator-Theorem und berechne rekursiv in $G_1$ und $G_2$ Matching $M_1, M_2$ maximalen Gewichts; definiere $M := M_1 \cup M_2$, $G' = G_1 \cup G_2$
2. Solange $S \neq \emptyset$:
   - wähle $v \in S$, $S := S - v$, und berechne mit Lemma aus $M'$ Matching maximalen Gewichts in $G' + v$

**Laufzeit** $t$ Laufzeit von Matching, $t'$ von Lemma, $c_1, c_2 \leq \frac{2}{3}$,$c_3 \in \mathbb{N}$, $c_1 + c_2 \leq 1$

$$t(n) = t(c_1 \cdot n) + t(c_2 \cdot n) + c_3\sqrt{n}t'(n)$$

Mit Mastertheorem kann $t(n)$ abgeschätzt werden durch

$$t(n) \in \mathcal{O}\left(n^{\frac{3}{2}}\right), \text{ falls } t'(n) \in \mathcal{O}(n), \text{ falls ungewichtet}$$

$$t(n) \in \mathcal{O}\left(n^{\frac{3}{2}} \log n\right), \text{ falls } t'(n) \in \mathcal{O}(n \log n), \text{ falls gewichtet}$$

# 2 Mixed-Max-Cut in planaren Graphen

## 2.1 Definition: Schnitt

$G = (V, E)$ Graph, $S \subset E$ heißt **Schnitt** von $G$, falls der durch $E - S$ induzierter Subgraph unzusammenhängend ist und für alle $(u, v) \in S$, $u$ und $v$ in verschiedenen Zusammenhangkomponenten liegen.

## 2.2 Definition: Mixed-Max Cut

Kantengewichte $w : E \to \mathbb{R}$
**Mixed-Max Cut**: Finde Schnitt $S$ mit $w(S) = \sum_{s \in S} w(S)$ maximal.
ist in bel. Graphen NP-Schwer.

**Beobachtung:** Mixed-Max Cut und Mixed-Min Cut sind äquivalent. (Vorzeichen der Gewichte umdrehen.)

**Spezialfall:** Min Cut Problem mit $w : E \to \mathbb{R}_{\geq 0}$ ist auch für beliebige Graphen in P.

## 2.3 Definition:

Matching M in G mit $|V|$ gerade heißt perfekt, falls $2|M| = |V|$

## 2.4 Polynomialer Algorithmus für Mixed-Max Cut in planaren Graphen

Verwende

- Dualität von Schnitten und Kreisen
- maximales Matching bzw. Planar Separator Theorem

Laufzeit in $\mathcal{O}\left(n^{3/2} \log n\right)$

Es gilt: $G$ enthält Euler-Kreis g.d.w. E kandendisjunkte Vereinigung einfacher Kreise g.d.w. $\forall v \in V$ ist $d(v) \in 2\mathbb{Z}$

Dualität von Schnitt in G und Menge von einfachen Kreisen in Dualgraph $G^*$ (bzgl. bel. pl. Einbettung) Menge von einfachen Kreisen = Kantenmenge, in der für alle Knoten der Knotengrad gerade ist =: gerade Menge

(maximaler Schnitt in G induziert maximalen Kreis in $G^*$ und umgekehrt)

1. Trianguliere G in $\mathcal{O}(n)$; zusätzliche Kanten erhalten Gewicht 0
2. berechne in $\mathcal{O}(n)$ Dualgraph $G^*$ bzgl. bel. pl. Einbettung; $G^*$ ist dann 3-regulär, d.h. $\forall v \in V^* : d(v) = 3$
3. Konstruiere zu $G^*$ Graph $G'$, so dass perfektes Matching min. Gewichts in $G'$ eine gerade Menge (bzw. Menge von Kreisen) max. Gewichts in $G^*$ induziert
4. berechne in $\mathcal{O}\left(n^{3/2} \log n\right)$ solch ein Matching bzw. gerade Menge
5. falls diese gerade Menge nichtleer, berechne daraus den entsprechenden Schnitt, ansonsten Sonderfall

### 2.4.1 Schritt 3

beachte, dass $G^*$ 3-regulär; ersetze jeden Punkt in $G^*$ durch ein Dreieck, erhalte $G'$; Matching ergibt zwei Fälle: (1)

Sei $m$ die Anzahl der Kanten in $G^*$ und $n$ die Anzahl der Knoten

$$\Rightarrow 3n = 2m \Rightarrow n \text{ gerade}$$

ergo hat $G'$ eine gerade Anzahl an Knoten. Wir sehen, dass mindestens ein perfektes Matching für $G'$ existiert.

### 2.4.2 Schritt 4

Konstruiere perfektes Matching minimalen Gewichts in $G'$

**Beobachtung** $M$ perfektes Matching minimalen Gewichts in $G = (V, E)$ mit $w : E \to \mathbb{R}$, g.d.w. $M$ perfektes Matching max. Gewichts in $G$ bzgl. $y(e) = W - w(e)$, für $W$ geeignet
erzwinge, dass Matching max. Gewichts perfekt ist:

- zu $M$ perfekt, betrachte $y(M) = \sum_{e \in M} y(e) = nW/2 - \sum_{e \in M} w(e) \geq n/2(W - w_{max})$
- Zu $N$ nicht perfekt, gilt $v(N) \leq (n/2 - 1)(W - w_{min})$
- Wähle $W$ so, dass

$$v(N) \leq (n/2 - 1)(W - w_{min}) < n/2(W - w_{max}) \leq y(M)$$

in $\mathcal{O}\left(n^{\frac{3}{2}} \log n\right)$

### 2.4.3 Schritt 5

Komplementmenge von perfektem Matching min. Gewichts in $G'$ induziert gerade Menge max. Gewichts in $G^*$ und damit max. Schnitt in $G$.
Es kann sein, dass resultierende Menge leer ist. Passiert, wenn max. Schnitt negatives Gewicht hat.

$\rightsquigarrow$ **Sonderfall:** Wollen nichttrivialen Schnitt erzwingen;
betrachte wieder Schritt 3, erzwinge, dass in perfektem Matching minimalen Gewichts für mindestens einen Knoten $v \in G^*$, Fall 2 eintritt.

**Vorgehensweise** betrachte alle Knoten $v \in G^*$ und $G^* - v$ sowie durch perfektes Matching in $G'$ induziertes Matching in $G^* - v$ und berechne mit Matching-Lemma ein Matching in $G^*$.
Wähle $M$ mit $w(M) = \min_{v \in V^*} w(M_v)$

**Frage** Wie kann man dabei Fall 2 bei v erzwingen? (2)

Folien: Maximale s-t-Flüsse in Planaren Graphen

## 2.5 Lemma

Für jeden Kozykel $C$ gilt: $\pi(C) \in \{-1, 0, 1\}$.
Ferner: $\pi(C) = 1 \iff C$ ist $(s,t)$-Schnitt

**Beweis**

Fall 1: $s, t$ liegen auf derselben Seite von $C^*$

$$\begin{aligned}
&\iff P \text{ kreuzt } C^* \text{ gleich oft in jeder Richtung} \\
&\iff C \text{ enthält dieselbe Zahl von Kanten in } P \text{ und } \overline{P} \\
&\iff \pi(C) = 0
\end{aligned}$$

Fall 2: $s$ liegt innen und $t$ außen von $C^*$

$$\begin{aligned}
&\iff P \text{ kreuzt } C^* \text{ einmal mehr in die Richtung von } s \to t \\
&\iff C \text{ enthält eine Kante mehr in } P \text{ als in } \overline{P} \\
&\iff \pi(C) = 1
\end{aligned}$$

Fall 2: $s$ liegt innen und $t$ außen von $C^*$
analog wie Fall 2

## 2.6 Lemma

$G$ besitzt einen gültigen $s-t$-Fluss mit Wert $\lambda$ genau dann, wenn $G^*_\lambda$ keinen negativen Kreis enthält.

**Beweis** Zeige '$\Longrightarrow$': Annahme: $G^*_\lambda$ enthält negativen Kreis $C^*$, d.h. $0 > c(\lambda, C^*) = \sum_{e \in C} c(\lambda, e) = \sum_{e \in C} c(e) - \lambda \sum_{e \in C} \pi(e) = c(C) - \lambda\pi(C) \Longrightarrow \pi(C) > c(C)/\lambda \leq 0 \Longrightarrow \pi(C) = 1$

$$\Longrightarrow C \text{ ist } s - t - \text{Schnitt}$$

Außerdem $c(C) < \lambda$, d.h. es existiert ein Schnitt mit Kapazität $< \lambda$, das ist ein Widerspruch
Zeige '$\Longrightarrow$': $G^*_\lambda$ enthält keinen negativen Kreis.
$\Longrightarrow$ kürzeste Wege wohldefiniert; sei $x$ in $G^*_\lambda$ beliebiger Ursprung, $dist(p, \lambda) :=$ Distanz von $x$ zu $p$

**Definition**

$$\phi(\lambda, e) := dist(\lambda, head(e^*)) - dist(\lambda, tail(e^*)) + \lambda\pi(e)$$

Zeige $\phi$ ist gütliger $st$-Fluss

1. Für $v \in V$ gilt: $\sum_w \phi(v \to w) = \sum_w \lambda\pi(v \to w)$ es folgt: $\phi(\lambda, \cdot)$ ist Fluss mit Wert $\lambda$
2. $slack(\lambda, e^*) := dist(\lambda, tail(e^*)) + c(\lambda, e) - dist(\lambda, head(e^*))$ es gilt: $slack(\lambda, e) = c(e) - \phi(\lambda, e)$ $\phi(\lambda, e) \leq c(e) \iff slack(\lambda, e) \geq 0$ Wäre $slack(\lambda, e) < 0$, dann folgt: $dist(\lambda, head(e^*)) > dist(\lambda, tail(e^*)) + c(\lambda, e^*)$, das wäre ein Widerspruch

## 2.7 Satz

Ein maximaler $st$-Fluss in einem $st$-planaren Graph kann in $\mathcal{O}(n \log n)$ Zeit berechnet werden.
Max $\lambda$, s.d. kein neg. Kreis in $G^*_\lambda$ ist Länge des kürzesten $ts$-Weges in $G^*_\lambda$

# 3 Das Menger-Problem

## 3.1 Zur Erinnerung

$S \subset V$ heißt Separator in $G$, falls $G - S$ unzusammenhängend.
$S \subset E$ heißt Schnitt in $G$, falls $G - S$ unzusammenhängend.

## 3.2 Definitionen

Zu $u, v \in V$ definiere den **Knotenzusammenhang**

$$\kappa_G(u,v) := \begin{cases} |V| - 1, \text{ falls } \{u,v\} \in E \\ \min\limits_{S \subset V} |S|, \text{ für } S \text{ Separator, der u und v trennt} \end{cases}$$

und $\kappa_G := \min_{u,v \in V} \kappa_G(u,v)$

$$\lambda_G(u,v) := \min_{S \subset E,\ \text{S Schnitt und trennt u und v}} |S|$$

und

$$\lambda(G) := \min_{u,v \in V} \lambda_G(u,v)$$

Zwei Wege heißen **kantendisjunkt**, wenn sie keine gemeinsame Kante enthalten, und (intern) **knotendisjunkt**, wenn sie außer Anfangs- und Endknoten keinen gemeinsamen Knoten enthalten.

## 3.3 Satz von Menger

Seien $s$ und $t$ Knoten in $G = (V, E)$ ($\{s,t\} \notin E$ bei knotendisjunkter Version)

- $\kappa_G(s,t) \geq k \iff \exists_{\geq k}$ paarweise knotendisjunkte $st$-Wege
- $\lambda_G(s,t) \geq k \iff \exists_{\geq k}$ paarweise kantendisjunkte $st$-Wege

## 3.4 Menger-Problem

Finde zu $G, s, t$ maximale Anzahl knotendisjunkter bzw. kantendisjunkter $st$-Wege.

## 3.5 Menger-Problem in planaren Graphen: kantendisjunkte Variante

Linearzeitalgorithmus basierend auf Right-First-DFS.

**Spezialfall** $s$ und $t$ liegen auf derselben Facette:
Right-First = im Gegenuhrzeigersinn nächste freie Kante in Adjazenzliste (relativ zur aktuellen eingehenden Kante).

**Algorithmus** $G$ planar eingebetteter Graph, OE $t$ auf äußerer Facette

1. Ersetze $G$ durch den gerichteten Graphen $\overrightarrow{G}$, indem $\{u,v\} \in E$ durch $(u,v)$ und $(v,u)$ ersetzt wird. (in $\mathcal{O}(n)$)
2. Berechne in $\mathcal{O}(n)$ Menge gerichteter einfacher kantendisjunkter Kreise $\overrightarrow{C_1}, \ldots, \overrightarrow{C_l}$ und konstruiere aus $\overrightarrow{G}$ den Graphen $\overrightarrow{G}_C$, indem die Richtung aller Kanten auf den $\overrightarrow{C_i}$ umgedreht wird.
3. Berechne in $\overrightarrow{G}_C$ in $\mathcal{O}(n)$ mittels RIGHT-FIRST-DFS eine maximale Anzahl kantendisjunkter gerichteter $st$-Wege.
4. Berechne aus den in Schritt 3 gefundenen $st$-Wegen in $\overrightarrow{G}_C$ gleiche Anzahl kantendisjunkter $st$-Wege in $G$ in $\mathcal{O}(n)$.

**Schritt 1**

## 3.6 Lemma

Seien $P_1, \ldots, P_r$ kantendisjunkte, gerichtete $st$-Wege in $\overrightarrow{G}$. Dann enthält

$$P = \{\{u,v\} \in E \mid \text{Genau eine der Kanten (u,v) und (v,u) liegt auf einem der } P_i\}$$

gerade $r$ kantendisjunkte $st$-Wege in $G$.

**Beweis** Zwei Fälle: Wir konstruieren in beiden Fällen aus gegebenen $st$-Wegen unproblematische $st$-Wege

1. $(u,v) \in P_i \wedge (v,u) \in P_i$: Entferne $(u,v,\ldots,v,u)$ bzw. $(v,u,\ldots,u,v)$ aus $P_i$
2. $(u,v) \in P_i \wedge (v,u) \in P_j$: $P_i = (A,u,v,B)$, $P_j = (C,v,u,D)$; konstruiere $\widetilde{P}_i = (A,D)$ und $\widetilde{P}_j = (C,B)$

**Schritt 2** $C_1, \ldots, C_l$ in $\overrightarrow{G}$, sodass

1. $\overrightarrow{G}_C$ enthält keine Rechtskreise, d.h. keine Kreise, deren Inneres rechts liegt (aus Sicht einer Kante).
2. Sei $\overrightarrow{P}_C \subset \overrightarrow{E}_C$ Menge der Kanten auf kantendisj. $s-t$ Wegen in $\overrightarrow{G}_C$ und $\overrightarrow{P} \subset \overrightarrow{E}$, wobei

   $$\overrightarrow{P} := (\overrightarrow{P}_C \cap \overrightarrow{E}) \cup \{(u,v) \in \overrightarrow{E} : (u,v) \text{ auf einem der } \overrightarrow{C}_i \text{ und } (v,u)' \notin \overrightarrow{P}_C\}$$

   Dann induziert $\overrightarrow{P}$ $k$ kantendisjunkte gerichtete $st$-Wege in $\overrightarrow{G}$ g.d.w $\overrightarrow{P}_C$ induziert $k$ kantendisjunkte gerichtete $st$-Wege in $\overrightarrow{G}_C$.

**Konstruktion der Kreise** $C_1, \ldots, C_l$ Sei $f$ Facette in $G$ bzw. $\overrightarrow{G}$; definiere Abstand von $f$ zur äußerer Facette $f_0$ als

$dist(f) :=$ Länge eines kürzesten Weges des Dualknotens $f^*$ zum Dualknoten der äußeren Facette $f_0^*$ in $G^*$

Definiere $C_i$ als Vereinigung der einfachen Kreise in $G$ für die alle Facetten $f$ im Inneren die Bedingung $dist(f) \geq i$ erfüllen. $\overrightarrow{C}_i$ sei entsprechender Rechtskreis in $\overrightarrow{G}$. Drehe alle diese $C_i$ um, erhalte $\overrightarrow{G}_C$.
$\overrightarrow{G}_C$ enthält keine Rechtskreise, da für jeden Rechtskreis in $\overrightarrow{G}$ beim Übergang zu $\overrightarrow{G}_C$ mindestens eine Kante des Kreises umgedreht wird.

Sei $\overrightarrow{P}_C \subset \overrightarrow{E}_C$ Kantenmenge zu $k$ $st$-Wegen in $\overrightarrow{G}_C$. Konstruiere dazu Kantenmenge $\overrightarrow{P}$ in $\overrightarrow{G}$.

$$\overrightarrow{P} := (\overrightarrow{P}_C \cap \overrightarrow{E}) \cup \{(u,v) \in \overrightarrow{E} : (u,v) \text{ auf einem } \overrightarrow{C}_j \text{ und } (v,u)' \notin \overrightarrow{P}_C\}$$

**Schritt 3** Berechnung einer maximalen Anzahl kantendisjunkter $st$-Wege in $\overrightarrow{G}_C$(in $\mathcal{O}(n)$)
Schleife über ausgehende Kanten aus $s$
RightFirstDepthSearch:
Suchschritt: rechteste nicht markierte auslaufende Kante in Bezug auf Referenzkante

Zwei Variationen, wie die **Referenzkante** zu wählen ist

1. Weihe: aktuell einlaufende Kante
2. Coupry: erste einlaufende Kante

**Korrektheitsbeweis zu Schritt 3** Beh.: $\overrightarrow{P}_C$ enthält maximale Anzahl kantendisjunkter $st$-Wegen.
Benutze dazu gewichtete Variante des Satz v. Menger, d.h. konstruiere $st$-Schnitt der entsprechenden Kapazität.
Schnitt A wird induziert durch geeigneten Kreis $\overrightarrow{K} \subset \overbrace{G}^{C}$ mit:

1. $s \in Innen(\overrightarrow{K})$ oder auf $\overrightarrow{K}$
2. $t \in Aussen(\overrightarrow{K})$
3. $A := \left\{(u,v) \in \overrightarrow{E}_C \mid u \text{ liegt auf } \overrightarrow{K}, v \in Aussen(\overrightarrow{K})\right\}$, $|A| =\#$ $st$-Wegen in $\overrightarrow{P}_C$

$\overrightarrow{K}$ wird mittels LeftFirst-Rückwärtssuche von $s$ aus in $\overrightarrow{P}_C$ konstruiert.
Wie sieht $\overrightarrow{K}$ aus:

**Variante 1** $\overrightarrow{K}$ geht von $s$ nach $s$

**Variante 2** $\overrightarrow{K}$ geht von $s \neq v_0$ nach $v_0$ und $s$
In diesem Fall den Kreis, der von $v_0$ nach $v_0$ beschrieben wird.

## 3.7 Lemma

Betrachte $\overrightarrow{G}_C = (V, \overrightarrow{E}_C)$ und $\overrightarrow{K}$, dann ist jede Kante $(u, v) \in \overrightarrow{E}_C$ mit $u$ auf $\overrightarrow{K}$ und $v \in Aussen(\overrightarrow{K})$ durch einen $st$-Weg aus $\overrightarrow{P}_C$ besetzt.

**Beweis**

1. Wenn $P_1, \ldots, P_l$ $st$-Wege sind und $\overrightarrow{K}$ ein Linkskreis von $s$ nach $s$, dann gehört keine der Kanten $(x, y)$, $x \in Aussen(\overrightarrow{K}), y \in \overbrace{K}$ zu einem der $p_i$.
   Wegen LeftFirst in Graph indziert durch $p_1, \ldots, p_l$ $(x, y) \in p_i$, für alle $1 \leq i \leq l$.
   Deswegen: Kante $(u, v)$ mit $u$ auf $\overrightarrow{K}$, $v \in Aussen(\overrightarrow{K})$ kann nicht auf einem Linkskreis aus $p_1, \ldots, P_l$ liegen.
2. betrachte $(u, v)$ mit $u$ auf $\overrightarrow{K}$, $v \in Aussen(\overrightarrow{K})$ und $(u, w)$ mit $w$ auf $\overrightarrow{K}$.