Karlsruher Institut für Technologie

Algorithmen für Planare Graphen

Vorlesung am 02.05.2022

Torsten Ueckerdt | 02. Mai 2022

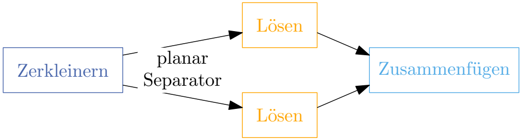
Gewichtsmaximales Matching – Algorithmus

Gegeben: planarer Graph $G = (V, E)$, Gewichtsfunktion $w\colon E \to \mathbb{R}$

Gesucht: Matching $M \subseteq E$ mit $w(M)$ maximal

Verwende wieder Planar Separator:

1. - Ist $n := |V(G)| \leq 5$, finde optimales Matching durch Brute-Force. .... $\mathcal{O}(1)$
2. - Andernfalls finde $\frac{3}{4}$-balancierten Separator S mit $|S| = \mathcal{O}(\sqrt{n})$. .... $\mathcal{O}(n)$
   - Berechne rekursiv optimale Matchings auf allen Komponenten von $G' := G - S$. .... $\mathcal{O}(?)$
   - Sei M' die Vereinigung dieser optimalen Matchings. (Beobachte: M' ist optimal für G'.)
3. - Solange $S \neq \emptyset$:
     - Wähle $v \in S$.
     - Finde alternierenden Weg P in $G' + v$ mit Endpunkt v mit $w(P - M') - w(P \cap M')$ maximal. . $\mathcal{O}(n \cdot |S|)$
     - Falls P erhöhend, ersetze M' durch $M' \triangle P$.
     - Lösche v aus S.
     - Ersetze G' durch $G' + v$.



Korrektheit? Laufzeit?

Korrektheit des Algorithmus

Karlsruher Institut für Technologie

Lemma 1.

Sei $G = (V, E)$ mit Gewichtsfunktion w, $v \in V$, M ein gewichtsmaximales Matching in $G - v$.
Dann sind äquivalent:

- M ist gewichtsmaximal in G.
- Es gibt keinen erhöhenden Weg mit Endpunkt v.

Lemma 2.

Sei weiterhin $\mathcal{P}_v$ die Menge aller erhöhenden Pfade mit Endpunkt v, $P \in \mathcal{P}_v$, sowie $M' := M \triangle P$. Dann sind äquivalent:

- M' ist gewichtsmaximal in G.
- $w(P - M) - w(P \cap M)$ ist maximal in $\mathcal{P}_v$.

Beobachtung:

- Daraus folgt direkt die Korrektheit des Algorithmus.

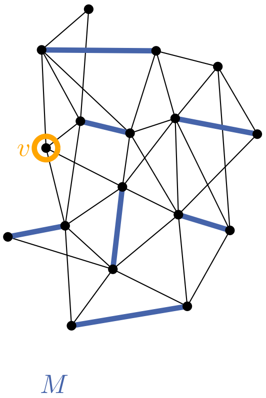
Beweis Lemma 1

Lemma 1.

Sei M ein gewichtsmaximales Matching in $G - v$. Dann sind äquivalent:

- M ist gewichtsmaximal in G.
- Es gibt keinen erhöhenden Weg mit Endpunkt v.

Beweis: "$\Rightarrow$" ✓



Beweis Lemma 1

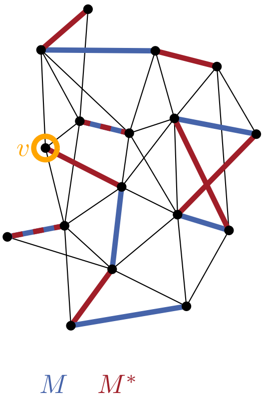
Lemma 1.

Sei M ein gewichtsmaximales Matching in $G - v$. Dann sind äquivalent:

- M ist gewichtsmaximal in G.
- Es gibt keinen erhöhenden Weg mit Endpunkt v.

Beweis: "$\Leftarrow$" Wir benutzen Kontraposition:

- Sei M nicht gewichtsmaximal.
- Also gibt es Matching M^* mit $w(M^*) > w(M)$.



Beweis Lemma 1

Karlsruher Institut für Technologie

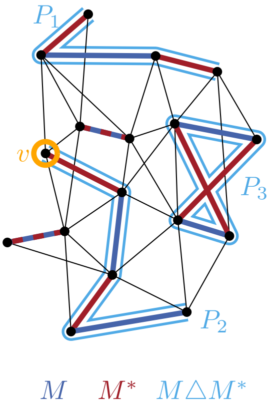
Lemma 1.

Sei M ein gewichtsmaximales Matching in $G - v$. Dann sind äquivalent:

- M ist gewichtsmaximal in G.
- Es gibt keinen erhöhenden Weg mit Endpunkt v.

Beweis: "$\Leftarrow$" Wir benutzen Kontraposition:

- Sei M nicht gewichtsmaximal.
- Also gibt es Matching M^* mit $w(M^*) > w(M)$.
- Betrachte Pfade und Kreise $P_1, \ldots, P_t$ in $M \triangle M^*$.
- Es gibt P_i mit $w(M^* \cap P_i) > w(M \cap P_i)$.
- Wäre $v \notin P_i$, dann wäre P_i bereits erhöhend in $G - v$. ✗
- Also liegt v auf P_i.
- Da $v \notin M$, ist v ein Endpunkt von P_i. ✓



Beweis Lemma 2

Lemma 2.

Sei M gewichtsmax. Matching auf $G - v$, $\mathcal{P}_v$ die Menge aller erhöhenden Pfade mit Endpunkt v, $P \in \mathcal{P}_v$, sowie $M' := M \triangle P$. Dann sind äquivalent:

- M' ist gewichtsmaximal in G.
- $w(P - M) - w(P \cap M)$ ist maximal unter allen $P \in \mathcal{P}_v$.

Beweis: "$\Rightarrow$" ✓

Beweis Lemma 2

Lemma 2.

Sei M gewichtsmax. Matching auf $G - v$, $\mathcal{P}_v$ die Menge aller erhöhenden Pfade mit Endpunkt v, $P \in \mathcal{P}_v$, sowie $M' := M \triangle P$. Dann sind äquivalent:

- M' ist gewichtsmaximal in G.
- $w(P - M) - w(P \cap M)$ ist maximal unter allen $P \in \mathcal{P}_v$.

Beweis: "$\Leftarrow$" Wir benutzen wieder Kontraposition:

- Sei nicht M' sondern M^* gewichtsmaximal in G.
- Es gibt einen erhöhenden Pfad P in $M \triangle M^*$ mit
  - v als Endpunkt (also $P \in \mathcal{P}_v$) und
  - $w(M^*) - w(M) = w(P_i - M) - w(P_i \cap M)$.
- Da $w(M^*) - w(M) > w(M') - w(M)$, war P nicht maximal. ✓

Gewichtsmaximales Matching – Algorithmus

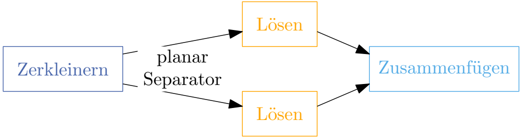
Karlsruher Institut für Technologie

Gegeben: planarer Graph $G = (V, E)$, Gewichtsfunktion $w\colon E \to \mathbb{R}$

Gesucht: Matching $M \subseteq E$ mit $w(M)$ maximal

Verwende wieder Planar Separator:

1. - Ist $n := |V(G)| \leq 5$, finde optimales Matching durch Brute-Force. .... $\mathcal{O}(1)$
2. - Andernfalls finde $\frac{3}{4}$-balancierten Separator S mit $|S| = \mathcal{O}(\sqrt{n})$. .... $\mathcal{O}(n)$
   - Berechne rekursiv optimale Matchings auf allen Komponenten von $G' := G - S$. .... $\mathcal{O}(?)$
   - Sei M' die Vereinigung dieser optimalen Matchings. (Beobachte: M' ist optimal für G'.)
3. - Solange $S \neq \emptyset$:
     - Wähle $v \in S$.
     - Finde alternierenden Weg P in $G' + v$ mit Endpunkt v mit $w(P - M') - w(P \cap M')$ maximal. .. $\mathcal{O}(n \cdot |S|)$
     - Falls P erhöhend, ersetze M' durch $M' \triangle P$.
     - Lösche v aus S.
     - Ersetze G' durch $G' + v$.



Korrektheit ✓ Laufzeit?

Laufzeit des Algorithmus

Karlsruher Institut für Technologie

Satz.

Ein gewichtsmax. Matching eines planaren Graphen mit n Knoten kann in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

Laufzeit des Algorithmus

Satz.

Ein gewichtsmax. Matching eines planaren Graphen mit n Knoten kann in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

- Wir finden $\frac{3}{4}$-balancierten Separator S in $\beta \cdot n$ Schritten.
- Dabei ist $|S| \in \mathcal{O}(\sqrt{n})$.
- Wir finden $|S|$ erhöhende Wege in $|S| \cdot \mathcal{O}(n) \leq \gamma \cdot n^{\frac{3}{2}}$ Schritten.
- Sei $T(n)$ die worst-case Gesamtlaufzeit für Instanzen mit n Knoten.

$$\Rightarrow \quad T(n) \leq T(n_1) + T(n_2) + \beta \cdot n + \gamma \cdot n^{\frac{3}{2}}$$

$$\text{mit } n \geq n_1 + n_2 \text{ und } \tfrac{1}{4} n \leq n_1, n_2 \leq \tfrac{3}{4} n.$$

Laufzeit des Algorithmus

Satz.

Ein gewichtsmax. Matching eines planaren Graphen mit n Knoten kann in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

- Wir finden $\frac{3}{4}$-balancierten Separator S in $\beta \cdot n$ Schritten.
- Dabei ist $|S| \in \mathcal{O}(\sqrt{n})$.
- Wir finden $|S|$ erhöhende Wege in $|S| \cdot \mathcal{O}(n) \leq \gamma \cdot n^{\frac{3}{2}}$ Schritten.
- Sei $T(n)$ die worst-case Gesamtlaufzeit für Instanzen mit n Knoten.

$$\Rightarrow \; T(n) \leq T(n_1) + T(n_2) + \beta \cdot n + \gamma \cdot n^{\frac{3}{2}}$$

mit $n \geq n_1 + n_2$ und $\frac{1}{4} n \leq n_1, n_2 \leq \frac{3}{4} n$.

- Ansatz: $T(n) \leq c \cdot n^{\frac{3}{2}}$ für eine Konstante c
- Sei $x = \frac{n_1}{n}$. Dann genügt es zu zeigen, dass

$$c \cdot n^{\frac{3}{2}} \geq c \cdot n_1^{\frac{3}{2}} + c \cdot (n - n_1)^{\frac{3}{2}} + \beta \cdot n^{\frac{3}{2}} + \gamma \cdot n^{\frac{3}{2}}$$

$$= \left(c \cdot (x^{\frac{3}{2}} + (1 - x)^{\frac{3}{2}}) + \beta + \gamma \right) \cdot n^{\frac{3}{2}}$$

Laufzeit des Algorithmus

Satz.

Ein gewichtsmax. Matching eines planaren Graphen mit n Knoten kann in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

- Wir finden $\frac{3}{4}$-balancierten Separator S in $\beta \cdot n$ Schritten.
- Dabei ist $|S| \in \mathcal{O}(\sqrt{n})$.
- Wir finden $|S|$ erhöhende Wege in $|S| \cdot \mathcal{O}(n) \leq \gamma \cdot n^{\frac{3}{2}}$ Schritten.
- Sei $T(n)$ die worst-case Gesamtlaufzeit für Instanzen mit n Knoten.

$$\Rightarrow \quad T(n) \leq T(n_1) + T(n_2) + \beta \cdot n + \gamma \cdot n^{\frac{3}{2}}$$

$$\text{mit } n \geq n_1 + n_2 \text{ und } \tfrac{1}{4}n \leq n_1, n_2 \leq \tfrac{3}{4}n.$$

- Ansatz: $T(n) \leq c \cdot n^{\frac{3}{2}}$ für eine Konstante c
- Sei $x = \frac{n_1}{n}$. Dann genügt es zu zeigen, dass

$$\begin{aligned} c \cdot n^{\frac{3}{2}} &\geq c \cdot n_1^{\frac{3}{2}} + c \cdot (n - n_1)^{\frac{3}{2}} + \beta \cdot n^{\frac{3}{2}} + \gamma \cdot n^{\frac{3}{2}} \\ &= \left(c \cdot (x^{\frac{3}{2}} + (1-x)^{\frac{3}{2}}) + \beta + \gamma\right) \cdot n^{\frac{3}{2}} \end{aligned}$$

- Da $\frac{1}{4} \leq x \leq \frac{3}{4}$ folgt $x^{\frac{3}{2}} + (1-x)^{\frac{3}{2}} \leq \frac{4}{5}$.
- Also ist bei uns $c \cdot n_1^{\frac{3}{2}} + c \cdot (n - n_1)^{\frac{3}{2}} \leq \frac{4}{5} \cdot c \cdot n^{\frac{3}{2}}$.

Laufzeit des Algorithmus

Satz.

Ein gewichtsmax. Matching eines planaren Graphen mit n Knoten kann in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

- Wir finden $\frac{3}{4}$-balancierten Separator S in $\beta \cdot n$ Schritten.
- Dabei ist $|S| \in \mathcal{O}(\sqrt{n})$.
- Wir finden $|S|$ erhöhende Wege in $|S| \cdot \mathcal{O}(n) \leq \gamma \cdot n^{\frac{3}{2}}$ Schritten.
- Sei $T(n)$ die worst-case Gesamtlaufzeit für Instanzen mit n Knoten.

$\Rightarrow$ $T(n) \leq T(n_1) + T(n_2) + \beta \cdot n + \gamma \cdot n^{\frac{3}{2}}$

mit $n \geq n_1 + n_2$ und $\frac{1}{4} n \leq n_1, n_2 \leq \frac{3}{4} n$.

- Ansatz: $T(n) \leq c \cdot n^{\frac{3}{2}}$ für eine Konstante c
- Sei $x = \frac{n_1}{n}$. Dann genügt es zu zeigen, dass

$$c \cdot n^{\frac{3}{2}} \geq c \cdot n_1^{\frac{3}{2}} + c \cdot (n - n_1)^{\frac{3}{2}} + \beta \cdot n^{\frac{3}{2}} + \gamma \cdot n^{\frac{3}{2}}$$
$$= \left(c \cdot (x^{\frac{3}{2}} + (1 - x)^{\frac{3}{2}}) + \beta + \gamma\right) \cdot n^{\frac{3}{2}}$$

- Da $\frac{1}{4} \leq x \leq \frac{3}{4}$ folgt $x^{\frac{3}{2}} + (1 - x)^{\frac{3}{2}} \leq \frac{4}{5}$.
- Also ist bei uns $c \cdot n_1^{\frac{3}{2}} + c \cdot (n - n_1)^{\frac{3}{2}} \leq \frac{4}{5} \cdot c \cdot n^{\frac{3}{2}}$.
- Wähle also c groß genug so dass $c \geq \frac{4}{5} \cdot c + \beta + \gamma$, also $c \geq 5(\beta + \gamma)$ ✓

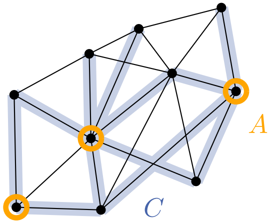
Mixed-Max-Cut

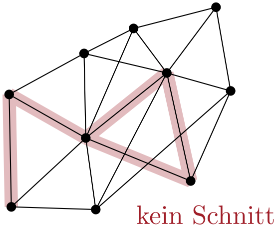
Definition.

Ein **Schnitt** in $G = (V, E)$ ist eine Kantenmenge $C \subseteq E$, die von einer Knotenmenge $A \subseteq V$ folgendermaßen induziert wird:

$$C = \{uv \in E \colon |A \cap \{u, v\}| = 1\}$$

Menge C enthält also genau die Kanten, die genau einen Endpunkt in A haben.





Mixed-Max-Cut

Definition.

Ein **Schnitt** in $G = (V, E)$ ist eine Kantenmenge $C \subseteq E$, die von einer Knotenmenge $A \subseteq V$ folgendermaßen induziert wird:

$$C = \{uv \in E \colon |A \cap \{u, v\}| = 1\}$$

Menge C enthält also genau die Kanten, die genau einen Endpunkt in A haben.

Problem MIXED-MAX-CUT.

- **Gegeben**:
  - Graph $G = (V, E)$,
  - Gewichtsfunktion $w \colon E \to \mathbb{R}$.
- **Gesucht**:
  - Schnitt $C \subseteq E$ mit $w(C) = \sum_{e \in C} w(e)$ maximal und $C \neq \emptyset$.

Spezialfälle:

- MAX-CUT
  Wenn $w(e) > 0 \quad \forall e \in E$.
  Allgemein $\mathcal{NP}$-schwer.
- MIN-CUT
  Wenn $w(e) < 0 \quad \forall e \in E$.
  Allgemein in $\mathcal{P}$.

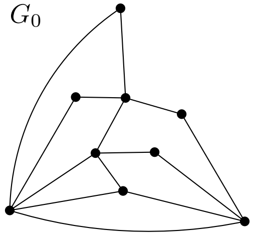
Mixed-Max-Cut – Schritt 1 und 2

Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir geben einen Algorithmus für $G_0 = (V, E_0)$ und $w\colon E_0 \to \mathbb{R}$:



Mixed-Max-Cut – Schritt 1 und 2

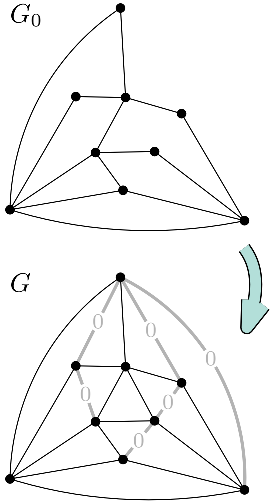
Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir geben einen Algorithmus für $G_0 = (V, E_0)$ und $w\colon E_0 \to \mathbb{R}$:

1. Trianguliere G_0 zu $G = (V, E)$.
   - Definiere $w(e) = 0$ für jede Kante $e \in E \setminus E_0$.



Mixed-Max-Cut – Schritt 1 und 2

Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir geben einen Algorithmus für $G_0 = (V, E_0)$ und $w\colon E_0 \to \mathbb{R}$:

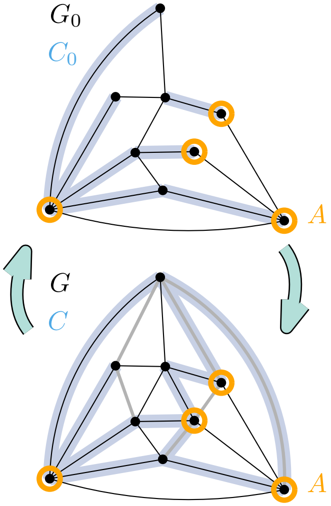
1. Trianguliere G_0 zu $G = (V, E)$.
   - Definiere $w(e) = 0$ für jede Kante $e \in E \setminus E_0$.

Beobachtung:

Für Schnitt C in G und $C_0 = C \cap E_0$ in G_0 sind äquivalent:

- $A \subseteq V$ induziert $C_0 \subseteq E_0$ in G_0.
- $A \subseteq V$ induziert $C \subseteq E$ in G.

Es gilt $w(C_0) = w(C)$.



Mixed-Max-Cut – Schritt 1 und 2

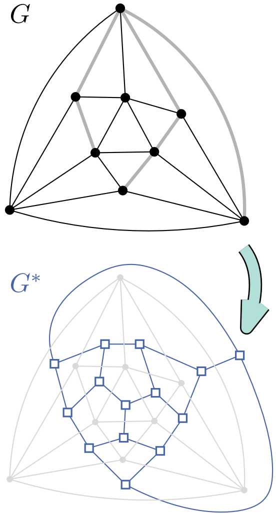
Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir geben einen Algorithmus für $G_0 = (V, E_0)$ und $w\colon E_0 \to \mathbb{R}$:

2. Betrachte Dualgraph $G^* = (F, E^*)$ von $G = (V, E)$.
   - Setze $w(e^*) = w(e)$ für alle $e \in E$.
   - G^* ist 3-regulär.
   - Für jede Kantenmenge $C^* \subseteq E^*$ hat jeder Dualknoten 0, 1, 2 oder 3 inzidente Kanten in C^*.



Mixed-Max-Cut – Schritt 1 und 2

Satz.

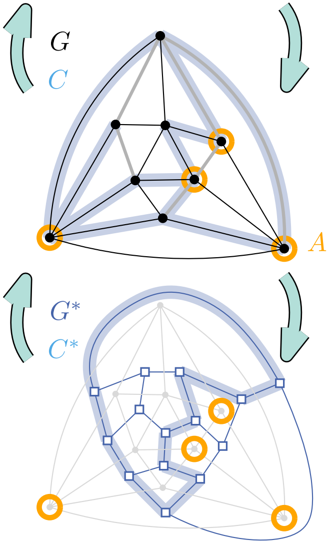
MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

2. Betrachte Dualgraph $G^*=(F,E^*)$ von $G=(V,E)$.
   - Setze $w(e^*)=w(e)$ für alle $e\in E$.
   - G^* ist 3-regulär.
   - Für jede Kantenmenge $C^*\subseteq E^*$ hat jeder Dualknoten 0, 1, 2 oder 3 inzidente Kanten in C^*.
   - Wenn $C\subseteq E$ ein Schnitt ist, ist $C^*\subseteq E^*$ speziell:

Definition.

Eine Kantenmenge $X\subseteq E^*$ heißt **gerade**, wenn jeder Knoten zu gerade vielen Kanten in X inzident ist.



C Schnitt $\leftrightarrow$ C^* gerade Menge

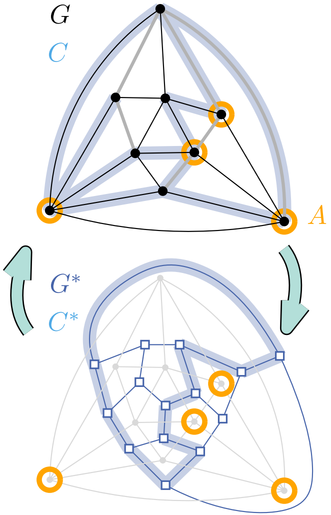
Lemma.

Sei $G = (V, E)$ trianguliert, G^* zu G dual. Dann sind äquivalent:

- $C \subseteq E$ ist Schnitt.
- $C^* \subseteq E^*$ ist eine gerade Kantenmenge.

Außerdem ist $w(C) = w(C^*)$.

Beweis:



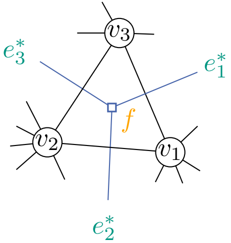
C Schnitt $\leftrightarrow$ C^* gerade Menge

Lemma.

Sei $G = (V, E)$ trianguliert, G^* zu G dual. Dann sind äquivalent:

- $C \subseteq E$ ist Schnitt.
- $C^* \subseteq E^*$ ist eine gerade Kantenmenge.

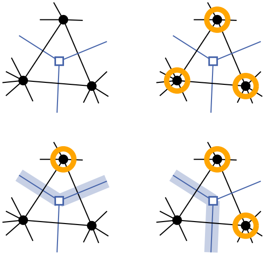
Außerdem ist $w(C) = w(C^*)$.



Beweis:

"$\Rightarrow$" Sei $C \subseteq E$ Schnitt in G induziert von $A \subseteq V$.

- Sei $f \in V(G^*)$ und e_1^*, e_2^*, e_3^* seine drei inzidenten Kanten.
- Betrachte das zu f zugehöriges Dreieck v_1, v_2, v_3 in G.
  - Ist $|A \cap \{v_1, v_2, v_3\}| = 0, 3,$ dann $|C^* \cap \{e_1^*, e_2^*, e_3^*\}| = 0$
  - Ist $|A \cap \{v_1, v_2, v_3\}| = 1, 2,$ dann $|C^* \cap \{e_1^*, e_2^*, e_3^*\}| = 2$
- Also ist C^* gerade. ✓



C Schnitt $\leftrightarrow$ C^* gerade Menge

Lemma.

Sei $G = (V, E)$ trianguliert, G^* zu G dual. Dann sind äquivalent:

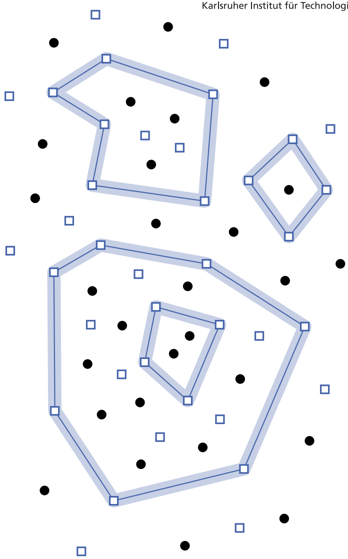
- $C \subseteq E$ ist Schnitt.
- $C^* \subseteq E^*$ ist eine gerade Kantenmenge.

Außerdem ist $w(C) = w(C^*)$.

Beweis:

"$\Leftarrow$" Sei $C^* \subseteq E^*$ eine gerade Kantenmenge in G^*.

- Dann hat jeder Dualknoten 0 oder 2 inzidente Kanten in C^*.
- Als Teilgraph ist C^* disjunkte Vereinigung von Kreisen und isolierten Punkten $C_1, \ldots, C_k$.
- Sei $A = \{v \in V \mid v \in \mathrm{int}(C_i)$ für ungerade viele $i\}$.



C Schnitt $\leftrightarrow$ C^* gerade Menge

Lemma.

Sei $G = (V, E)$ trianguliert, G^* zu G dual. Dann sind äquivalent:

- $C \subseteq E$ ist Schnitt.
- $C^* \subseteq E^*$ ist eine gerade Kantenmenge.

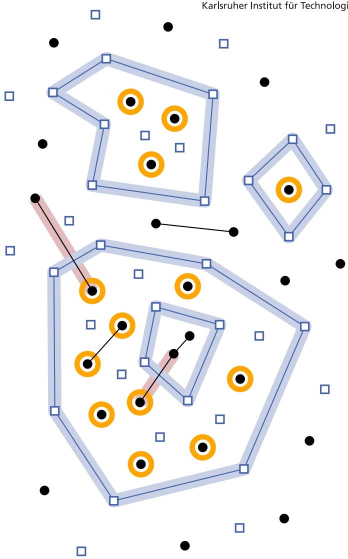
Außerdem ist $w(C) = w(C^*)$.

Beweis:

- Sei $A = \{v \in V \mid v \in \mathrm{int}(C_i) \text{ für ungerade viele } i\}$. Dann gilt

$$
\begin{aligned}
& e \in E \text{ ist in } C \text{ (dual zu } C^*) \\
\Leftrightarrow\ & e^* \in C^* \\
\Leftrightarrow\ & e^* \in C_i \text{ für ein } i \in [k]. \\
\Leftrightarrow\ & \text{Endpunkte von } e \text{ liegen auf verschiedenen Seiten von } C_i. \\
\Leftrightarrow\ & \text{Genau einer der Endpunkte von } e \text{ ist in } A.
\end{aligned}
$$

- Also ist C ein Schnitt. ✓



Mixed-Max-Cut – Schritt 3

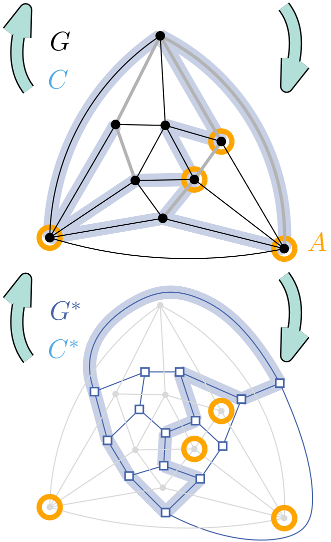
Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir suchen jetzt also eine gewichtsmaximale gerade Kantenmenge C^* in G^*.

Das heißt, jeder Knoten hat Grad 0 oder 2 in C^*.



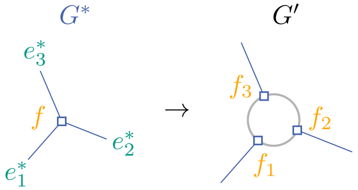
Mixed-Max-Cut – Schritt 3

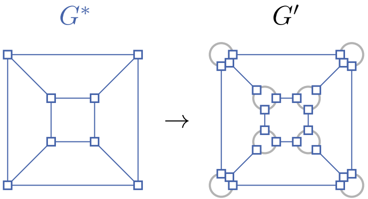
Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

3. Modifiziere $G^* = (F, E^*)$ zu $G' = (V', E')$.
   - G' wird folgendermaßen aus G^* konstruiert.
   - Für jeden Knoten $f \in F$ und seine drei inzidenten Kanten e_1^*, e_2^*, e_3^*:
     - Lösche f und füge drei neue Knoten f_1, f_2, f_3 hinzu.
     - Lösche ursprüngliche Kante $e_i^* = uf$ und füge uf_i ein. Diese übernehmen das Gewicht der e_i^*.
     - Füge außerdem für jedes Paar f_if_j eine Kante ein ($i \neq j$). Diese Kanten bekommen Gewicht 0.
   - G' ist wieder planar und 3-regulär.





C^* gerade Menge $\leftrightarrow$ C' 2-Faktor

Definition.

Sei $k \in \mathbb{N}$ eine Zahl. Eine Kantenmenge $X \subseteq E$ heißt ***k*-Faktor**, wenn jeder Knoten zu genau k Kanten in X inzident ist.

- 1-Faktoren heißen auch **perfekte Matchings**.

C^* gerade Menge $\leftrightarrow$ C' 2-Faktor

Karlsruher Institut für Technologie

Definition.

Sei $k \in \mathbb{N}$ eine Zahl. Eine Kantenmenge $X \subseteq E$ heißt **k-Faktor**, wenn jeder Knoten zu genau k Kanten in X inzident ist.

- 1-Faktoren heißen auch **perfekte Matchings**.

Lemma.

- Für jede gerade Menge $C^* \subseteq E^*$ existiert ein 2-Faktor $C' \subseteq E'$ mit $C' \cap E^* = C^*$.
- Für jeden 2-Faktor $C' \subseteq E'$ ist $C^* = C' \cap E^*$ eine gerade Menge.
- Es gilt $w(C') = w(C^*)$.

C^* gerade Menge $\leftrightarrow$ C' 2-Faktor

Definition.

Sei $k \in \mathbb{N}$ eine Zahl. Eine Kantenmenge $X \subseteq E$ heißt ***k*-Faktor**, wenn jeder Knoten zu genau k Kanten in X inzident ist.

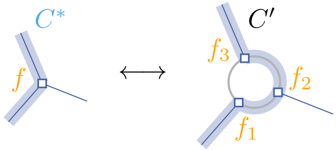
- 1-Faktoren heißen auch **perfekte Matchings**.

Lemma.

- Für jede gerade Menge $C^* \subseteq E^*$ existiert ein 2-Faktor $C' \subseteq E'$ mit $C' \cap E^* = C^*$.
- Für jeden 2-Faktor $C' \subseteq E'$ ist $C^* = C' \cap E^*$ eine gerade Menge.
- Es gilt $w(C') = w(C^*)$.

Beweis:

1. Fall $f \in F$ hat zwei inzidente Kanten in C^*:



C^* gerade Menge $\leftrightarrow$ C' 2-Faktor

Definition.

Sei $k \in \mathbb{N}$ eine Zahl. Eine Kantenmenge $X \subseteq E$ heißt **k-Faktor**, wenn jeder Knoten zu genau k Kanten in X inzident ist.

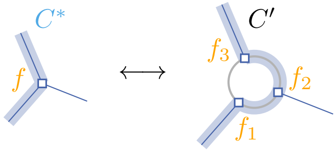
- 1-Faktoren heißen auch **perfekte Matchings**.

Lemma.

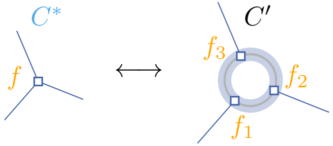
- Für jede gerade Menge $C^* \subseteq E^*$ existiert ein 2-Faktor $C' \subseteq E'$ mit $C' \cap E^* = C^*$.
- Für jeden 2-Faktor $C' \subseteq E'$ ist $C^* = C' \cap E^*$ eine gerade Menge.
- Es gilt $w(C') = w(C^*)$.

Beweis:

1. Fall $f \in F$ hat zwei inzidente Kanten in C^*:



2. Fall $f \in F$ hat keine inzidente Kante in C^*:



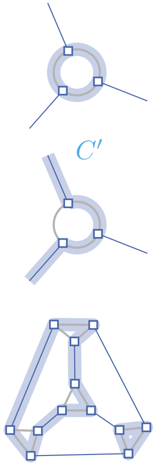
Mixed-Max-Cut – Schritt 4

Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

Wir suchen jetzt also einen gewichtsmaximalen 2-Faktor C' in G'.
Jeder Knoten soll also genau 2 inzidente Kanten in C' haben.



Mixed-Max-Cut – Schritt 4

Satz.

MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

Beweis:

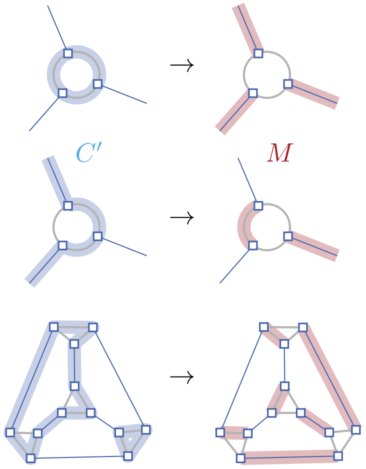
Wir suchen jetzt also einen gewichtsmaximalen 2-Faktor C' in G'. Jeder Knoten soll also genau 2 inzidente Kanten in C' haben.

4. Betrachte 1-Faktoren (perfekte Matchings) statt 2-Faktoren.
   - Da G' 3-regulär ist, ist das Komplement eines 2-Faktors C' in G' ein perfektes Matching M.

$$M = E' - C'$$

   - Der 2-Faktor C' ist gewichtsmaximal genau dann, wenn das komplementäre perfekte Matching M gewichtsminimal ist.

$$w(M) = w(E') - w(C')$$



Mixed-Max-Cut – Überblick

Eingabe: $G_0 = (V, E_0)$ planar.

1\. $\downarrow$

Trianguliere zu $G = (V, E)$.

2\. $\downarrow$

Dualisiere zu $G^* = (F, E^*)$.

3\. $\downarrow$

Modifiziere zu $G' = (V', E')$.

4\. $\searrow$

Berechne ein gewichtsminimales perfektes Matching M in G'.

Ausgabe: $C_0 = C \cap E_0$ Mixed-Max-Cut in G_0.

$\uparrow$

$C = (C^*)^*$ Mixed-Max-Cut in G.

$\uparrow$

$C^* = C' \cap E^*$ gew.max. gerade Menge in G^*

$\uparrow$

$C' = E' - M$ gew.max. 2-Faktor in G'

$\nearrow$

Mixed-Max-Cut – Überblick

Karlsruher Institut für Technologie

$$
\begin{array}{ccc}
\textbf{Eingabe: } G_0 = (V, E_0) \text{ planar.} & & \textbf{Ausgabe: } C_0 = C \cap E_0 \text{ Mixed-Max-Cut in } G_0. \\
1.\ \downarrow & & \uparrow \\
\text{Trianguliere zu } G = (V, E). & & C = (C^*)^* \text{ Mixed-Max-Cut in } G. \\
2.\ \downarrow & & \uparrow \\
\text{Dualisiere zu } G^* = (F, E^*). & & C^* = C' \cap E^* \text{ gew.max. gerade Menge in } G^* \\
3.\ \downarrow & & \uparrow \\
\text{Modifiziere zu } G' = (V', E'). & & C' = E' - M \text{ gew.max. 2-Faktor in } G' \\
4.\ \searrow & & \nearrow \\
& \text{Berechne ein gewichtsminimales perfektes Matching } M \text{ in } G'. &
\end{array}
$$

- Wie berechnen wir M? (Existenz?)
- Warum ist das korrekt?
- Warum ist $C_0 \neq \emptyset$?
- Was ist die Laufzeit?

Mixed-Max-Cut – Überblick

Eingabe: $G_0 = (V, E_0)$ planar.

1. $\downarrow$

Trianguliere zu $G = (V, E)$.

2. $\downarrow$

Dualisiere zu $G^* = (F, E^*)$.

3. $\downarrow$

Modifiziere zu $G' = (V', E')$.

4. $\searrow$

Berechne ein gewichtsminimales perfektes Matching M in G'.

$\nearrow$

$C' = E' - M$ gew.max. 2-Faktor in G'

$\uparrow$

$C^* = C' \cap E^*$ gew.max. gerade Menge in G^*

$\uparrow$

$C = (C^*)^*$ Mixed-Max-Cut in G.

$\uparrow$

Ausgabe: $C_0 = C \cap E_0$ Mixed-Max-Cut in G_0.

- Wie berechnen wir M? (Existenz?)
- Warum ist das korrekt?
- Warum ist $C_0 \neq \emptyset$?
- Was ist die Laufzeit?

$\longrightarrow$ In der nächsten Vorlesung!