Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Lehrstuhl für
INFORMATIK I
Effiziente Algorithmen und
wissensbasierte Systeme

Institut für Informatik

Algorithmische Graphentheorie

Sommersemester 2017

5. Vorlesung

Matchings / Paarungen II

– Kombinatorischer Algorithmus, Anwendung für Handlungsreisende,
LP-Runden –

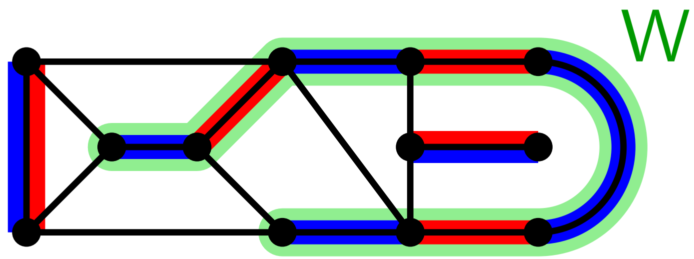
Prof. Dr. Alexander Wolff

Lehrstuhl für Informatik I

Alternierende und augmentierende Wege

Ziel: Besseres Problemverständnis $\to$ kombinatorische (d.h. nicht flussbasierte) Algorithmen für größte Matchings.

Bsp.



$G = (V, E)$ unger. Graph
$M \subseteq E$ Matching in G.

Wie können wir ein gegebenes (nicht-größtes) Matching vergrößern?

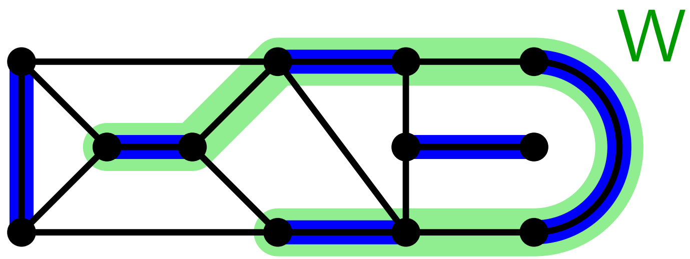
enthält abwechselnd M- und nicht-M-Kanten

- Finde einen (M-) *alternierenden Weg* W, dessen Endknoten *M-frei* sind. So ein Weg heißt (M-) *augmentierend*.
- Setze $M' = W \Delta M$, wobei $A \Delta B = (A \cup B) \setminus (A \cap B)$ die *symm. Differenz* von A und B ist.

Alternierende und augmentierende Wege

Ziel: Besseres Problemverständnis $\rightarrow$ kombinatorische (d.h. nicht flussbasierte) Algorithmen für größte Matchings.

Bsp.



$G = (V, E)$ unger. Graph
$M \subseteq E$ Matching in G.

Wie können wir ein gegebenes (nicht-größtes) Matching vergrößern?

enthält abwechselnd M- und nicht-M-Kanten

- Finde einen (M-) *alternierenden Weg* W, dessen Endknoten *M-frei* sind. So ein Weg heißt (M-) *augmentierend*.
- Setze $M' = W \Delta M$, wobei $A \Delta B = (A \cup B) \setminus (A \cap B)$ die *symm. Differenz* von A und B ist.

Beob.
– Augmentierende Wege haben ungerade Länge.
– M' ist um 1 Kante größer als M
(da M' zusätzlich die Endknoten von W paart).

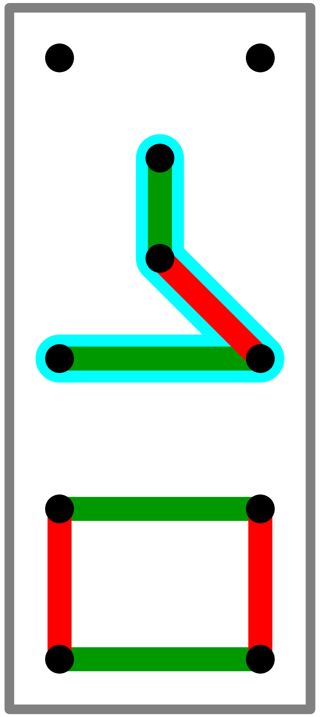
Satz von Berge

Paris, 1990 © Adrian Bondy

Claude Berge (1926–2002)

Satz. 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.

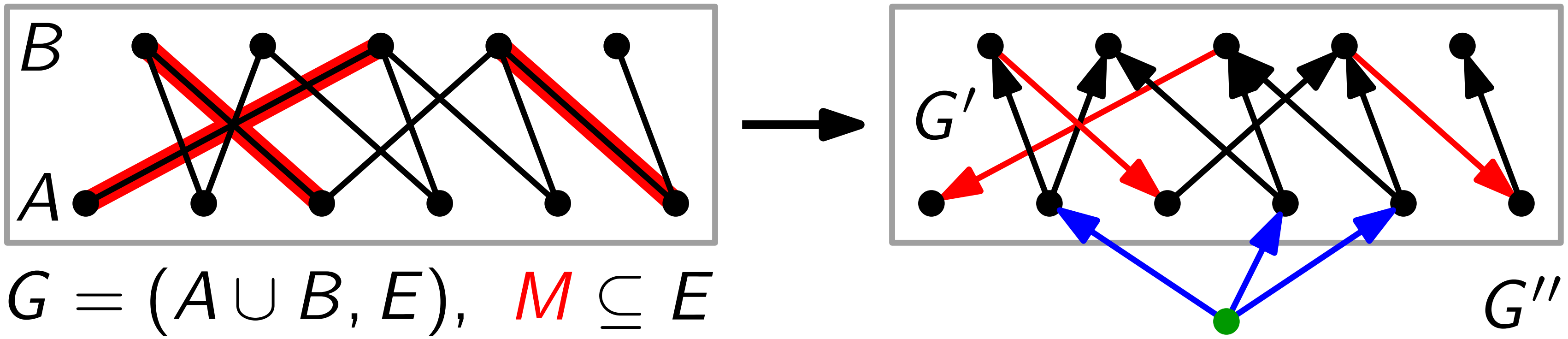
Beweis. „$\Rightarrow$" Klar: jeder augm. Weg würde M vergrößern ↯
„$\Leftarrow$" Annahme: M ist kein größtes Matching.



$\Rightarrow$ Es gibt ein Matching M' mit $|M'| > |M|$.
Betrachte $G_\Delta = (V, M \Delta M')$.
Beobachtung: Max-Grad$(G_\Delta) = 2$.
$\Rightarrow$ G_Δ besteht aus einfachen alt. Wegen & Kreisen.
Alle Kreise in G_Δ haben gerade Länge.
$\Rightarrow$ Ex. alt. Weg mit mehr Kanten aus M' als aus M.
$\Rightarrow$ Dieser Weg ist M-augm. ↯ zur Annahme. $\square$

Zurück zu: größte Matchings in bip. Graphen

Frage: Wie finden wir augmentierende Wege?



$G = (A \cup B, E)$, $M \subseteq E$

Idee: Richte M-Kanten nach unten, nicht-M-Kanten nach oben. $\longrightarrow$ $G' = (A \cup B, E')$

Augmentierende Wege in G $\longleftrightarrow$ gerichtete Wege mit M-freien Endknoten in G'

Definiere $G'' = \big(A \cup B \cup \{s\}, E' \cup \{sa \mid a \in A,\ M\text{-frei}\}\big)$

Breitensuche BFS(G'', s) erreicht einen M-freien Endknoten in B $\Leftrightarrow$ G hat M-augm. Weg.

Ergebnis

Algo: $M := \emptyset$.
Führe BFS $\leq |V|/2$ mal aus –
bis kein freier Knoten in B mehr gefunden wird.
Gib größtes Matching zurück.

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

Bem. Dinics Fluss-Algorithmus berechnet [KN, Kapitel 9.6]
– maximale Flüsse in allg. Graphen in $O(V^2E)$ Zeit
– Matchings in bipartiten Graphen in $O(\sqrt{V}E)$ Zeit.

Satz. Selbst in einem beliebigen Graphen $G = (V, E)$ lässt sich eine größte Paarung in $O(\sqrt{V}E)$ Zeit berechnen.
[Micali & Vazirani, FOCS'80]

Wiederholung – Tree-Doubling für Δ-TSP

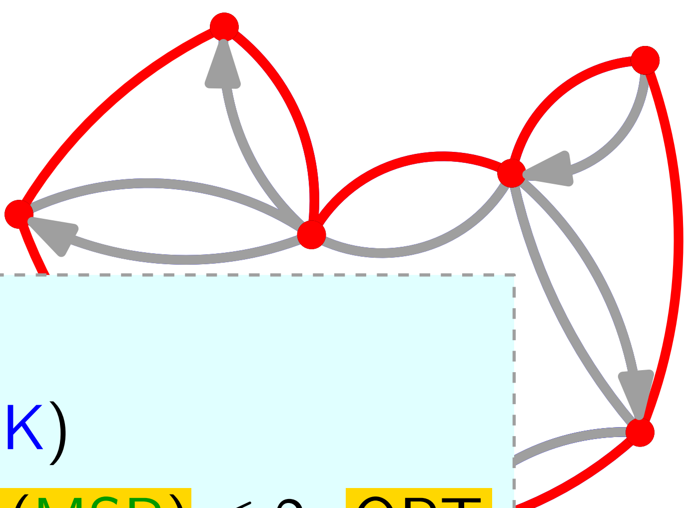
Problem: *Metrisches Traveling Salesperson Problem (Δ-TSP)*

Gegeben: unger. vollständiger Graph $G = (V, E)$
mit Kantenkosten $c \colon E \to \mathbb{R}_{\geq 0}$,
die die Dreiecksungleichung erfüllen.

Gesucht: Hamiltonkreis in G mit minimalen Kosten.

Satz. Es gibt eine 2-Approximation für Δ-TSP.

Beweis.



1. *Algorithmus*

Berechne MSB von G.

Verdopple MSB $\Rightarrow$ eulersch!

Durchlaufe Eulerkreis.

Überspringe besuchte Knoten.

Füge „Abkürzungen“ ein.

2. *Analyse*

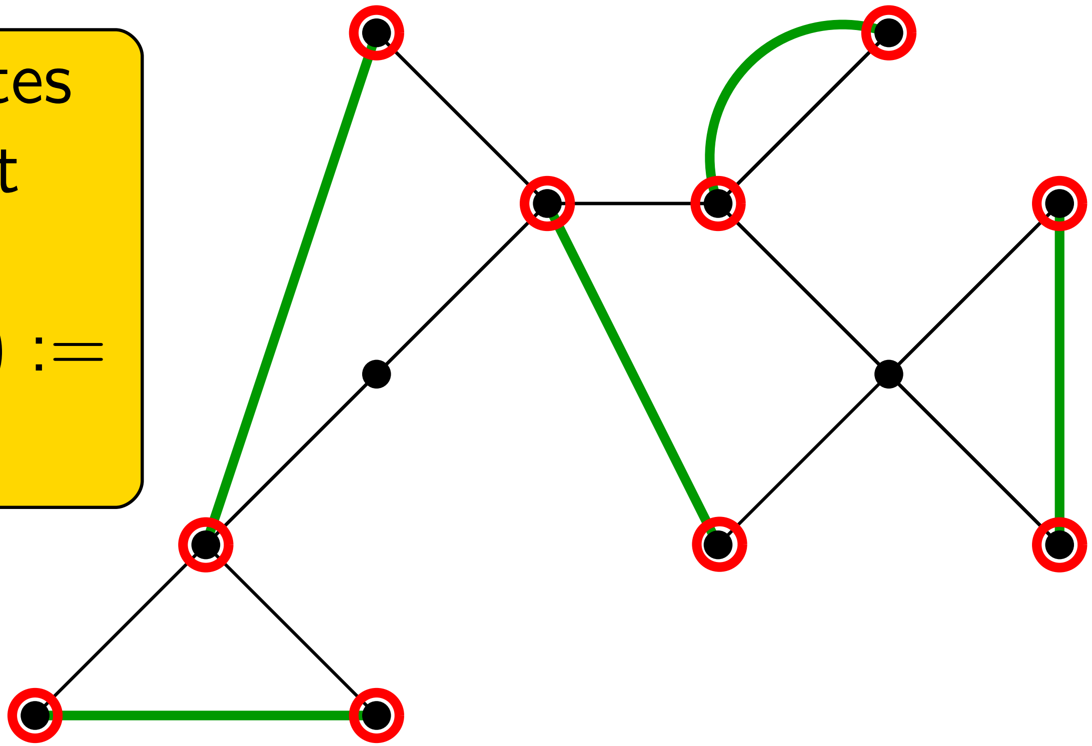
$$c(\text{ALG}) \leq c(\text{EK}) = 2 \cdot c(\text{MSB}) \leq 2 \cdot \text{OPT}$$

Opt. TSP-Tour minus eine Kante ist (i.A. nicht minimaler) Spannbaum!!
„Die Kunst der unteren Schranke“

Christofides' Algorithmus

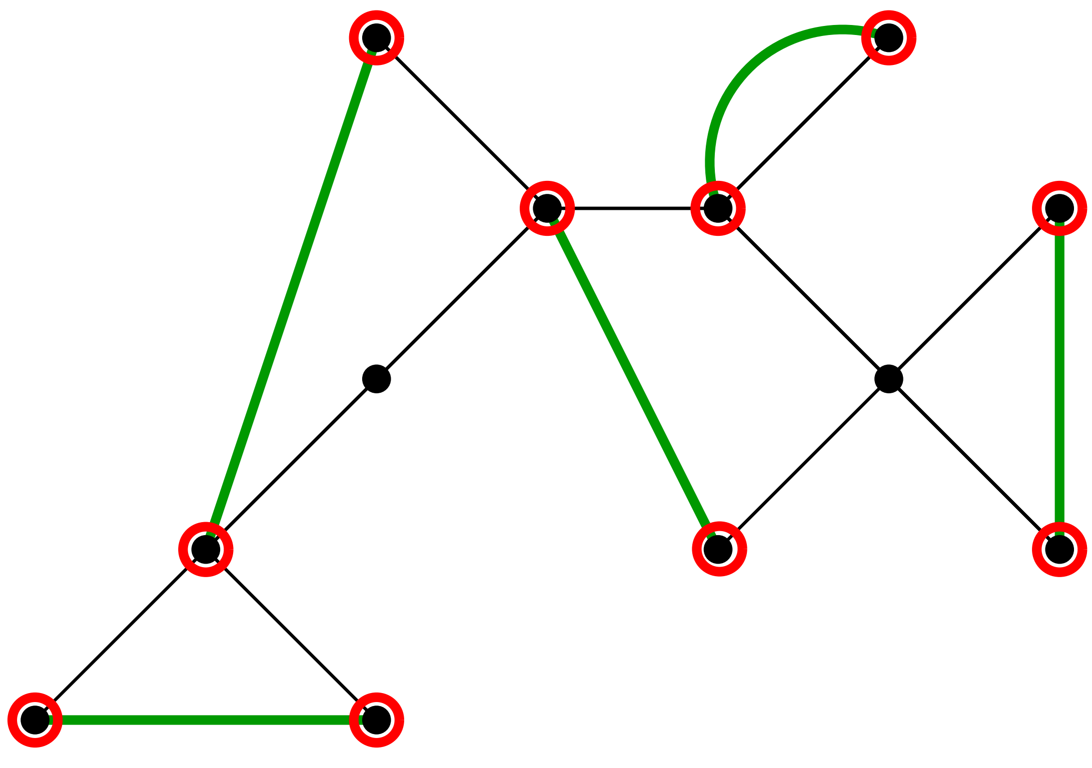
- Ermittle einen minimalen Spannbaum B für G.
- Betrachte $G[U]$ mit U Menge der Knoten ungeraden Grades in B.
- Ermittle *kostenminimales perfektes* Matching M für $G[U]$
  (existiert, da $|U|$ gerade und $G[U]$ vollständig).

M ist perfektes Matching mit minimalen Kosten $c(M) := \sum_{e \in M} c(e)$



Christofides' Algorithmus

- Ermittle einen minimalen Spannbaum B für G.
- Betrachte $G[U]$ mit U Menge der Knoten ungeraden Grades in B.
- Ermittle *kostenminimales perfektes* Matching M für $G[U]$
  (existiert, da $|U|$ gerade und $G[U]$ vollständig).
- Berechne aus eulerschem Graphen $B \cup M$ erst Eulertour und dann eine Rundtour T wie bei Tree-Doubling.

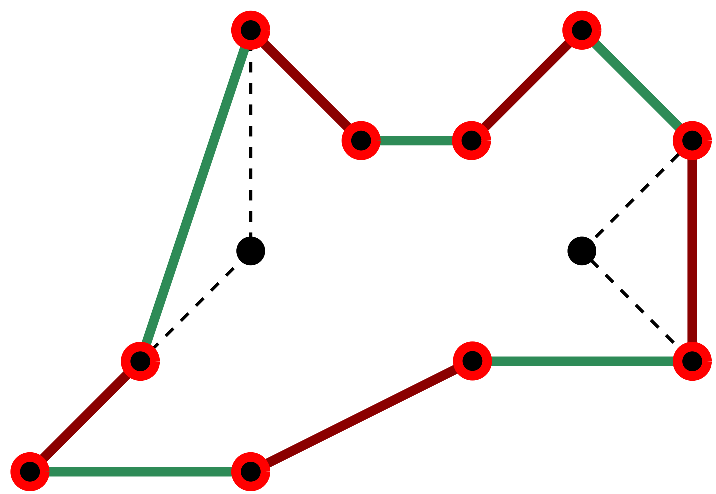


Analyse von Christofides

Satz. Christofides liefert eine 3/2-Approximation für Δ-TSP.

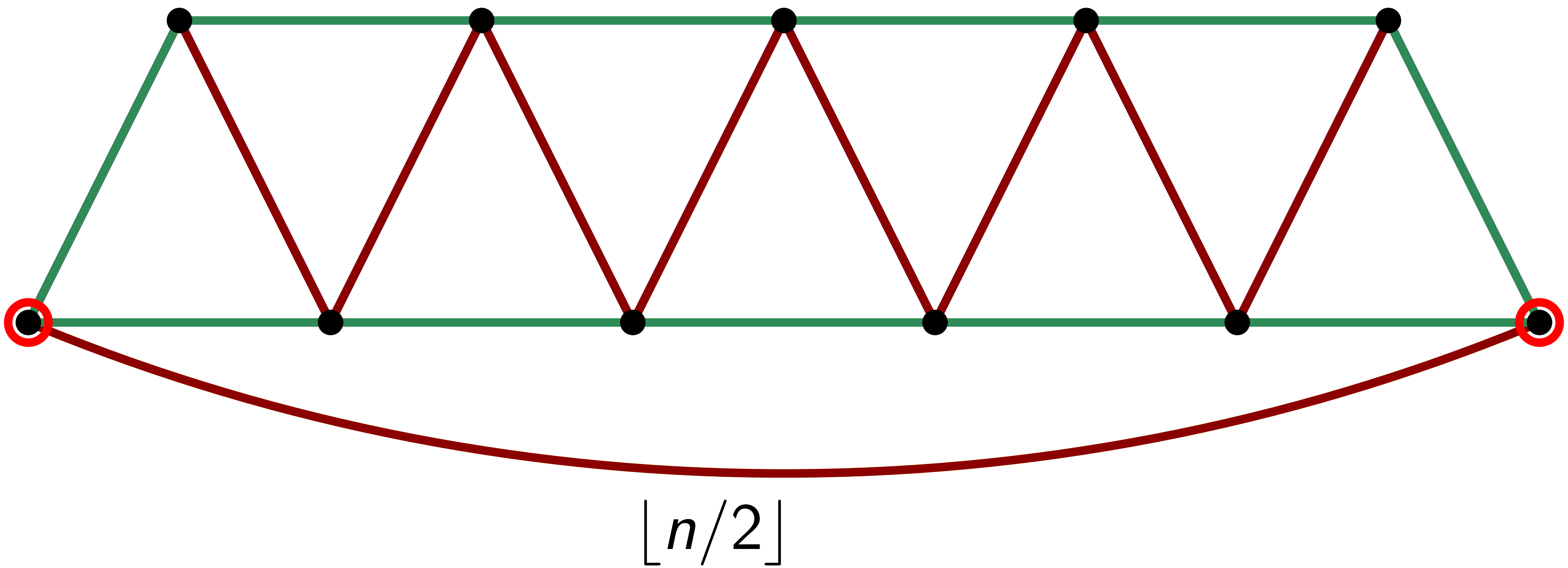
Beweis.

- Genügt zu zeigen, dass $c(B \cup M) \leq 3/2 \cdot \mathrm{OPT}$, da $c(T) \leq c(B \cup M)$.
- Da $c(B) \leq \mathrm{OPT}$, genügt es zu zeigen, dass $c(M) \leq \mathrm{OPT}/2$.
- Betrachte optimale Tour T^*.
- Abkürzen von T^* liefert Tour T^{**} in $G[U]$ mit $c(T^{**}) \leq c(T^*) = \mathrm{OPT}$.
- Zerlege in T^{**} in zwei disjunkte perfekte Matchings M', M'' für $G[U]$.
- O.B.d.A. $c(M') \leq c(M'')$.
- Also gilt $c(M') \leq c(T^{**})/2 \leq \mathrm{OPT}/2$.



Die Analyse ist scharf

Unbeschriftete Kanten haben Länge 1.



MSB

Matching

Christofides mit Kosten $n + \lfloor n/2 \rfloor - 1$

$\mathrm{OPT} = n$

$(n + \lfloor n/2 \rfloor - 1)/n \longrightarrow 3/2$

Online-Evaluierung

https://evasys.zv.uni-wuerzburg.de/evasys_10/online.php?pswd=...

JETZT!

Kostenminimale perfekte Matchings

Def. Gegeben: vollständiger Graph $G = (V, E)$ mit Kantenkosten $c \colon E \to \mathbb{R}_{\geq 0}$.
Gesucht: perfektes Matching M mit minimalen Kosten $c(M) = \sum_{e \in M} c(e)$.

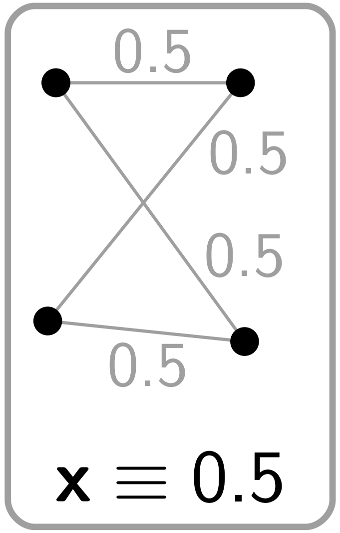
Satz. Ein kostenminimales perfektes Matching kann in $O(V^3)$ Zeit berechnet werden.

Beweis. Siehe [Edmonds '65]; ziemlich kompliziert :-($\square$

Im Folgenden betrachten wir das Problem nur in *bipartiten* Graphen $G = (A \cup B, E)$.

Kostenminimale perfekte Matchings in bipartiten Graphen

- Aufgabe: Formulieren Sie ein ILP!



Minimiere $\sum_{uv \in E} c_{uv} x_{uv}$

unter den Nebenbed. $\sum_{uv \in E} x_{uv} = 1$ für $u \in A \cup B$

$\cancel{x_{uv} \in \{0, 1\}}$ für $uv \in E$

$x_{uv} \geq 0$

- Effizient lösbar? I.A. nicht – VC is Spezialfall von ILP.
- Betrachte sogenannte *LP-Relaxierung* $\Rightarrow$ effizient lösbar!
  Bei Minimierungsproblemen gilt $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}_{\mathrm{ILP}}$.
- Problem: *fraktionale* Lösungen!

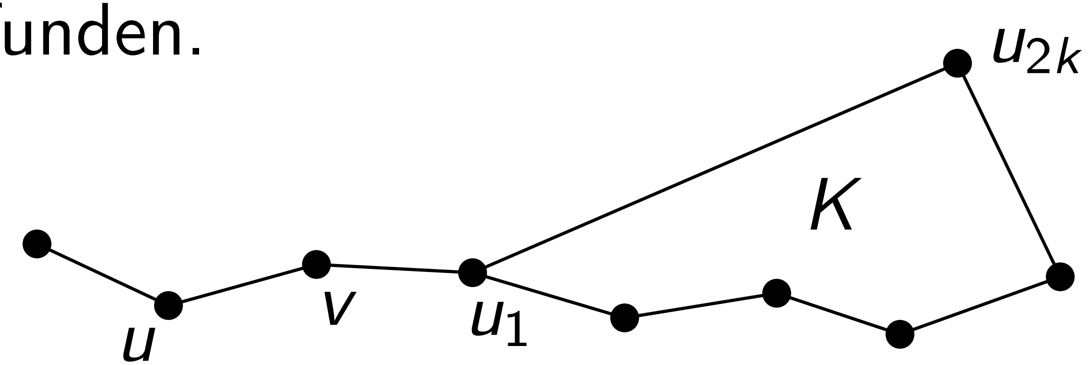
LP-Runden

Minimiere	$\sum_{uv \in E} c_{uv} x_{uv}$	
unter den Nebenbed.	$\sum_{uv \in E} x_{uv} = 1$	für $u \in A \cup B$
	$x_{uv} \geq 0$	für $uv \in E$

Betrachte *fraktionale* Kante uv (d.h. $0 < x_{uv} < 1$):

u und v sind jeweils zu weiterer fraktionaler Kante adjazent.

Erweitere Pfad iterativ, bis fraktionaler Kreis $K = (u_1, \ldots, u_{2k})$ gefunden.



Kreis gerade, da Graph bipartit (siehe ÜA).

LP-Runden – Fortsetzung (I)

Minimiere	$\sum_{uv \in E} c_{uv} x_{uv}$	
unter den Nebenbed.	$\sum_{uv \in E} x_{uv} = 1$	für $u \in A \cup B$
	$x_{uv} \geq 0$	für $uv \in E$

Zerlege K in disjunkte Matchings M_1, M_2. Sei $c(M_1) \leq c(M_2)$.

Wähle $\epsilon := \min\{\, x_e \mid e \in M_2 \}$.

Setze $x'_e := x_e + \epsilon$ für $x_e \in M_1$,

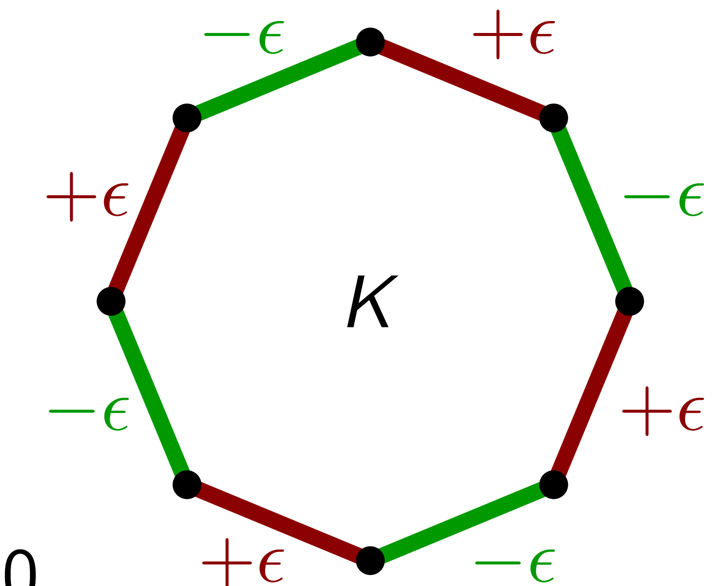
setze $x'_e := x_e - \epsilon$ für $x_e \in M_2$,

und $x'_e := x_e$ sonst $(x_e \in E \setminus K)$.

Resultierender Lösungsvektor $\mathbf{x}'$ zulässig.

Für mind. ein $e \in M_2$ gilt dann $x'_e = 0$.

Kostenänderung $\epsilon \cdot c(M_1) - \epsilon \cdot c(M_2) \leq 0$.



LP-Runden – Fortsetzung (II)

Minimiere	$\sum_{uv \in E} c_{uv} x_{uv}$	
unter den Nebenbed.	$\sum_{uv \in E} x_{uv} = 1$	für $u \in A \cup B$
	$x_{uv} \geq 0$	für $uv \in E$

Wiederhole vorige Schritte, solange fraktionale Kanten existieren.

Beob.: Integrale Kanten werden nicht weiter verändert.
Kostenfunktion erhöht sich nicht.

$\Rightarrow$ Algorithmus terminiert nach $\leq |E|$ Iterationen mit ganzzahliger und optimaler (!) Lösung.

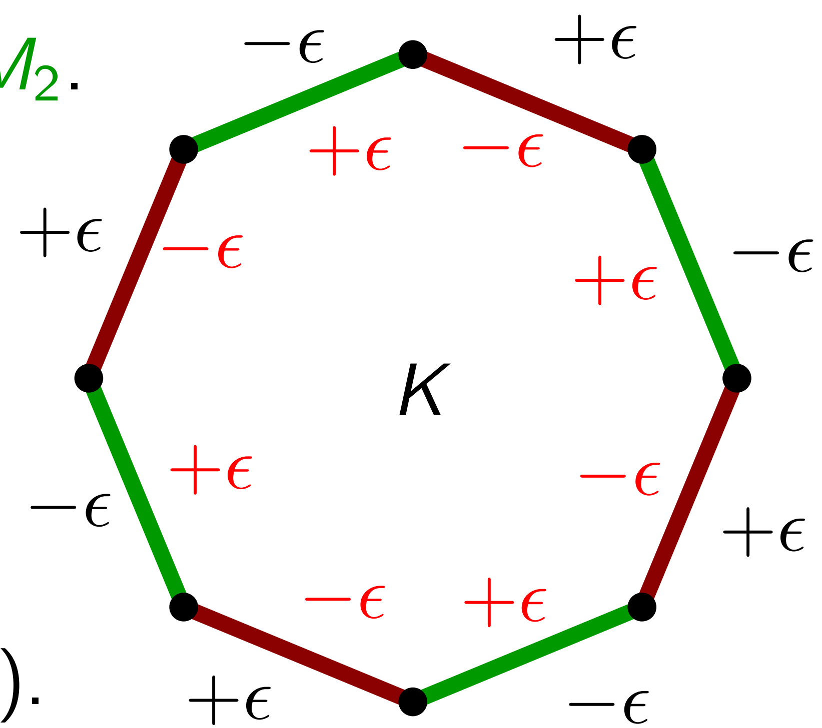
Satz. Ein kostenminimales perfektes Matching eines bipartiten Graphen lässt sich in Polynomialzeit ermitteln.

Extrempunkt-Lösungen (I)

Minimiere	$\sum_{uv \in E} c_{uv} x_{uv}$	
unter den Nebenbed.	$\sum_{uv \in E} x_{uv} = 1$	für $u \in A \cup B$
	$x_{uv} \geq 0$	für $uv \in E$

Zerlege K in disjunkte Matchings M_1, M_2.
Wähle $\epsilon := \min\{ x_e \mid e \in \boxed{M_1} \cup M_2 \}$.

Setze $x'_e := x_e + \epsilon$ für $x_e \in M_1$,
setze $x'_e := x_e - \epsilon$ für $x_e \in M_2$.
Setze $x''_e := x_e \boxed{-} \epsilon$ für $x_e \in M_1$,
setze $x''_e := x_e \boxed{+} \epsilon$ für $x_e \in M_2$.
Setze $x''_e := x'_e := x_e$ sonst $(e \in E \setminus K)$).



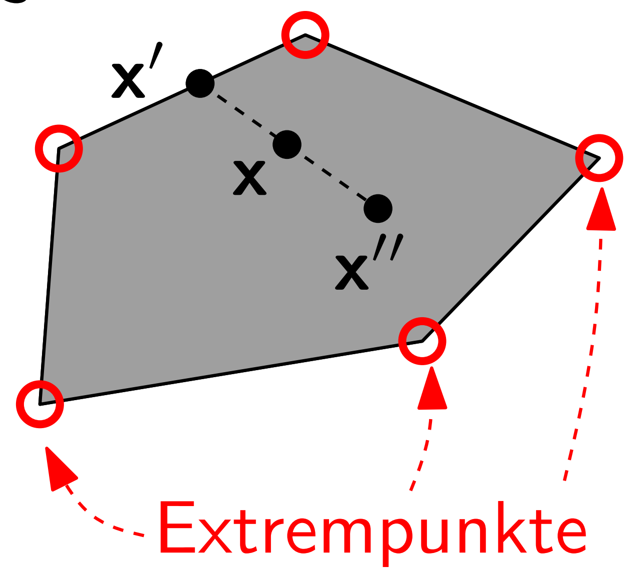
Beob.: Lösungen $\mathbf{x}', \mathbf{x}''$ sind beide zulässig und $\mathbf{x} = \frac{1}{2}(\mathbf{x}' + \mathbf{x}'')$.

Extrempunkt-Lösungen (II)

$$\left.\begin{array}{|lll|}\hline \text{Minimiere} & \displaystyle\sum_{uv\in E} c_{uv}x_{uv} & \\ \text{unter den Nebenbed.} & \displaystyle\sum_{uv\in E} x_{uv} = 1 & \text{für } u \in A \cup B \\ & x_{uv} \geq 0 & \text{für } uv \in E \\ \hline \end{array}\right\} (\star)$$

Es gilt $\mathbf{x} = \frac{1}{2}(\mathbf{x}' + \mathbf{x}'') \Rightarrow \mathbf{x}$ Konvexkombination von $\mathbf{x}', \mathbf{x}''$.

$\Rightarrow$ $\mathbf{x}$ ist *kein* Extrempunkt des Lösungsraums (konvexes Polytop) von $(\star)$.



Satz. Polytop $(\star)$ für bipartite Matchings hat ganzzahlige Extrempunkte.

Clou: Gängige LP-Algorithmen terminieren mit Extrempunkt-Lösungen (sofern existent) und erfordern für (1) *kein* anschließendes LP-Runden!!