A&W G-07 🎲

WhatsApp-Gruppe

Exercise T1.1 – *Disjoint Paths*

Let $G = (V, E)$ be an undirected graph. For two vertices $s, t \in V$ we denote by $\lambda_{s,t}$ the maximal number of edge-disjoint paths between s and t in G.

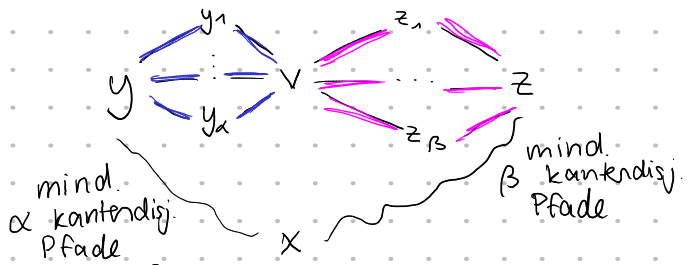
(a) Show that for three pairwise different vertices $a, b, c \in V$ we have $\lambda_{a,c} \geq \min\{\lambda_{a,b}, \lambda_{b,c}\}$.

Th. von Menger: Wenn wir $\lambda_{s,t} - 1$ Kanten löschen, dann sind s und t immer noch zusammenhängend

Annahme: $\min\{\lambda_{a,b}; \lambda_{b,c}\} \geq 1$, ansonsten trivial

→ wir löschen $\min\{\lambda_{a,b}; \lambda_{b,c}\} - 1$ Kanten
→ a und b connected + b und c connected
→ a und c connected
↔ $\lambda_{a,c} \geq \min\{\lambda_{a,b}; \lambda_{b,c}\}$

(b) Let $x, y, z \in V$ be three pairwise different vertices and $\alpha, \beta \in \mathbb{N}$ such that $\alpha \leq \lambda_{x,y}$, $\beta \leq \lambda_{x,z}$, and $\alpha + \beta \leq \max\{\lambda_{x,y}, \lambda_{x,z}\}$. Show that there are α many x-y paths and β many x-z-path such that all $\alpha + \beta$ paths are pairwise edge-disjoint.
(Hint: try to construct a graph that contains a vertex v such that the task reduces to finding $\alpha + \beta$ edge-disjoint x-v paths.)



w.t.s.: $\lambda_{x,v} \geq \alpha + \beta$
Löschen $\alpha + \beta - 1$ Kanten
o.B.d.A. $\lambda_{x,y} \geq \lambda_{x,z}$ $\qquad \alpha + \beta \leq \lambda_{x,y}$

→ x und y sind immer noch verbunden

→ Case ①: Wir löschen max. $\alpha - 1$ Kanten von den blauen Kanten
→ y und v noch verbunden
→ x und v noch verbunden

→ Case ②: Wir löschen mind. α Kanten
→ max. $\beta - 1$ Kanten übrig (↑ gelöschte)
→ mind. β kantendisjunkte Pfade von x nach v
→ x und v immer noch verbunden

∎

SHORT RECAP: Matchings

Matching := Kantenmenge $M \subseteq E$, kein Knoten des Graphen zu mehr als einer Kante aus M inzident ist

Knoten v wird von M überdeckt, falls es eine Kante $e \in M$ gibt, die v enthält.

perfektes Matching: Wenn jeder Knoten durch genau eine Kante des Matchings überdeckt wird ($|M| = \frac{|V|}{2}$)

Ein Matching M ist...
- ... inklusionsmaximal, falls $M \cup \{e\}$ kein Matching $\forall e \in E \setminus M$
- ... kardinalitätsmaximal, falls $|M| \geq |M'|$ für alle Matchings M' in G

Matchings in Graphen finden

GREEDY-MATCHING (G)

1: $M \leftarrow \emptyset$
2: **while** $E \neq \emptyset$ **do**
3: wähle eine beliebige Kante $e \in E$
4: $M \leftarrow M \cup \{e\}$
5: lösche e und alle inzidenten Kanten in G

Laufzeit: $O(|E|)$

Satz 1.47. Der Algorithmus GREEDY-MATCHING bestimmt in Zeit $O(|E|)$ ein inklusionsmaximales Matching M_{Greedy} für das gilt:

$$|M_{Greedy}| \geq \frac{1}{2}|M_{max}|,$$

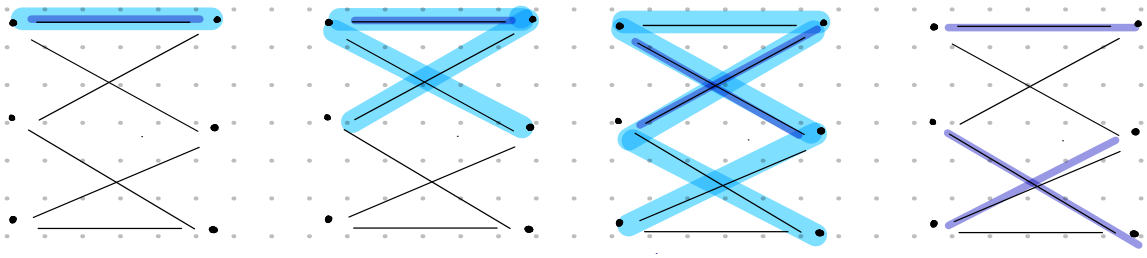
wobei M_{max} ein kardinalitätsmaximales Matching sei.

Konzept der augmentierenden Pfade

Sei M ein Matching in G.
Der M-augmentierende Pfad P...

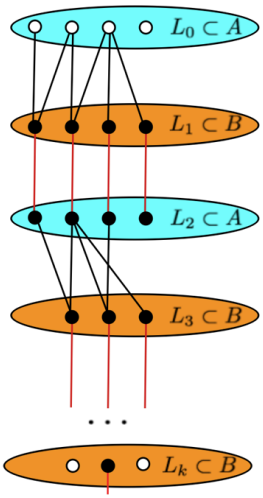
- ... besteht abwechselnd aus Kanten $E \setminus M$ und M
- ... beginnt und endet in einem von M unüberdeckten Knoten

Wir können M nun zu M' vergrößern: $M' = M \oplus P$



AUGMENTING_PATH $(G = (A \uplus B, E), M)$

1: $L_0 := \{$unüberdeckte Knoten in $A\}$
2: Markiere alle Knoten aus L_0 als besucht.
3: **if** $L_0 = \emptyset$ **then**
4: **return** M ist maximal
5: **for all** $i = 1$ **to** n **do**
6: **if** i ungerade **then**
7: $L_i := \{$unbesuchte Nachbarn von L_{i-1} via Kanten in $E \setminus M\}$
8: **else**
9: $L_i := \{$unbesuchte Nachbarn von L_{i-1} via Kanten in $M\}$
10: Markiere alle Knoten aus L_i als besucht.
11: **if** L_i enthält unüberdeckten Knoten v **then**
12: Finde Pfad P von L_0 nach v durch backtracking
13: **return** P // *terminiert Algorithmus*
14: **return** M ist schon maximal



Satz 1.48 (Satz von Berge). Ist M ein Matching in einem Graphen $G = (V, E)$, das nicht kardinalitätsmaximal ist, so existiert ein augmentierender Pfad zu M.

Satz von Hall

Satz 1.52 (Satz von Hall, Heiratssatz). Für einen bipartiten Graphen $G = (A \uplus B, E)$ gibt es genau dann ein Matching M der Kardinalität $|M| = |A|$, wenn gilt

$$|N(X)| \geq |X| \quad \text{für alle } X \subseteq A. \tag{1.1}$$

Aufgabe:

Die exklusive Käse Kochschule in einem abgelegenen Schweizer Bergdorf hat $2k$ Schüler. Ironischerweise haben heute alle Schüler aus versehen ihre Brotdosen mit dem Snack fürs Mittagessen zu Hause vergessen. Die entscheiden sich stattdessen ein Festmahl zu kochen: ein legendäres k-Gang Menü. Sie haben schon 15 Minuten ihrer einstündigen Mittagspause darauf verwendet, zu planen, welche Gerichte in dem extravaganten Menü enthalten sein sollen und beschliessen nun schnell mit dem Kochen zu beginnen.

Jedes der gewählten Gerichte ist so kompliziert, dass es die volle Aufmerksamkeit von 2 Köchen braucht. Andererseits, da die Köche sich ja noch in der Ausbildung befinden, kann jeder Schüler nur eine Teilmenge der Gerichte zubereiten. Damit das Menü auch gelingt müssen beide Köche wissen, wie das jeweilige Gericht zubereitet wird.

Jeder der $2k$ Köche hat eine Liste der Gerichte, die er/sie zubereiten kann zur verfügung gestellt. Jetzt ist die Frage, wie die Köche sich in Paare aufteilen sollen, sodass sie alle k Gerichte kochen können. Oder ist dies etwa unmöglich?

(a) Modellieren Sie das Problem mithilfe eines Graphen G, sodass ein Aufteilung der Köche wie oben beschrieben existiert genau dann wenn G ein perfektes Matching hat.

(b) Nehmen Sie an, dass es möglich ist alle Gerichte zu kochen. Zeigen Sie, dass es für jede Menge an Gerichten X wenigstens $2|X|$ Köche gibt die wissen, wie man mindestens eines dieses Gericht zubereitet.

(c) Umgekehrt, nehmen Sie an, dass es für jede Menge an Gerichten X wenigstens $2|X|$ Köche gibt die wissen, wie man mindestens eines dieses Gericht zubereitet. Zeigen Sie, dass es möglich ist, alle Gerichte zuzubereiten.

Key Idea für Graphenkonstruktion: Kopie der Knoten, die Gerichte repräsentieren

$G = (A \uplus B, E)$

$A :=$ Menge der Köche
$B :=$ Menge der verdoppelten Gerichte
$(u,v) \in E$ falls Koch u Gericht v kochen kann

Algorithmus von Hopcroft & Karp

MAXIMAL_MATCHING $(G = (A \oplus B, E))$ (Hopcroft und Karp)

1: $M := \{e\}$ für irgendeine Kante $e \in E$.
2: **while** es gibt noch augmentierende Pfade **do**
3: $k :=$ Länge eines kürzesten augmentierenden Pfades
4: Finde eine inklusionsmaximale Menge S von paarweise disjunkten augmentierenden Pfaden der Länge k.
5: **for all** P aus S **do**
6: $M := M \oplus P$. // *augmentiere entlang der Pfade aus* S
7: **return** M

Satz 1.49. Der Algorithmus von Hopcroft und Karp durchläuft die while-Schleife nur $O(\sqrt{|V|})$ Mal. Er berechnet daher ein maximales Matching in einem bipartiten Graphen in Zeit $O(\sqrt{|V|} \cdot (|V| + |E|))$.

Gabow's Algorithmus

Satz 1.54. Ist $G = (V, E)$ ein 2^k-regulärer bipartiter Graph, so kann man in Zeit $O(|E|)$ ein perfektes Matching bestimmen.

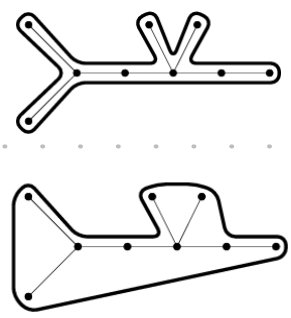
$k \geq 1$
→ Eulertouren finden (pro ZHKs)
→ Löschen jeder zweiten der Eulertour
→ 2^{k-1}-regulären bipartiten Graphen
→ solange wiederholen bis 2^0-regulär
→ perfektes Matching

Satz von Frobenius

Satz 1.53. Sei $G = (A \uplus B, E)$ ein k-regulärer bipartiter Graph. Dann gibt es $M_1, \ldots, M_k$ so dass $E = M_1 \uplus \ldots \uplus M_k$ und alle M_i, $1 \leq i \leq k$, perfekte Matchings in G sind.

TRAVELLING SALESMAN PROBLEM (TSP)

Satz: Für das metrische TSP gibt es einen 2-Approximationsalgorithmus mit Laufzeit $O(n^2)$.



1. Bestimme MST T
2. Verdopple alle Kanten von T
3. Bestimme Eulertour W
4. Durchlaufe W, mit Abkürzungen, so dass jeder Knoten nur einmal besucht wird $\Rightarrow$ Hamiltonkreis C

$\ell(T) \leq opt(K_n, \ell)$

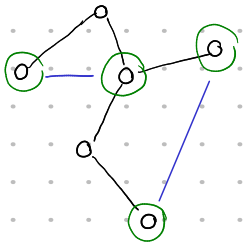
$2\ell(T) \leq 2 \cdot opt(K_n, \ell)$

$\ell(W) = 2\ell(T)$

$\ell(C) \leq \ell(W) = 2 \cdot \ell(T) \leq 2 opt(K_n, \ell)$

↑ Dreiecksungleichung

Satz 1.51. Für das METRISCHE TRAVELLING SALESMAN PROBLEM gibt es einen 3/2-Approximationsalgorithmus mit Laufzeit $O(n^3)$.



1. Bestimme MST T
2. X := Knoten mit ungeradem Grad in T
   Bestimme min. Matching M für X
3. Bestimme Eulertour W
4. Kürze Eulertour W ab und finde Hamiltonkreis C

$\ell(T) \leq opt(K_n, \ell)$

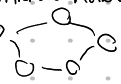
$\ell(T) + \ell(M) \leq \frac{3}{2} opt(K_n, \ell)$

$\ell(W) = \ell(T) + \ell(M) \leq \frac{3}{2} opt(K_n, \ell)$

$\ell(C) \leq \ell(W) \leq \frac{3}{2} opt(K_n, \ell)$

MINIQUIZVORBEREITUNG

1) Angenommen, G ist ein Graph, der eine Eulertour enthält, und die Anzahl Knoten von G ist gerade. Dann enthält G ein perfektes Matching.
→ Gegenbeispiel: 

2) Der Satz von Dirac impliziert, dass Graphen, welche einen Knoten mit weniger als $\frac{n}{2}$ Nachbarn besitzen, keinen Hamiltonkreis haben.
→ Gegenbeispiel: 

3) Sei M ein Matching in einem Graphen G. Dann gilt für jeden M-augmentierenden Pfad, dass er Länge mind. $\frac{|M|}{2}$ hat.
→ Falsch

4) Für das metrische TSP-Problem gibt es einen 2-Approximationsalgorithmus, aber keinen 4-Approximationsalgorithmus.
→ Falsch

5) Für einen bipartiten Graphen $G=(A \dot\cup B, E)$ gilt genau dann $|N(X)| \geq |X|$ für alle $X \subseteq A$, wenn es ein Matching M der Kardinalität $|M|=|A|$ gibt.
→ Wahr, Satz von Hall

6) Jeder Baum ist bipartit.
→ Wahr

7) Sei G ein vollständiger Graph mit Gewichtsfunktion auf den Kanten, sodass das Gewicht auf jeder Kante mind. 1 ist. Angenommen, dass eine optimale TSP-Route in G Kosten 10 hat. Geben Sie eine bestmögliche obere Schranke auf die Kosten eines minimalen Spannbaums in G.
→ 9

8) In der Vorlesung haben wir einen Algorithmus für eine 2-Approximation des TSP gesehen, der Laufzeit $O(n^2)$ hat.
→ Falsch, nur für metrische MSTs

9) Wir nehmen an, dass G ein inklusionsmax. Matching M der Grösse 8 enthält. Welche der folgenden Optionen sind mögliche Grössen eines kardinalitätsmax. Matchings in G?
4 **8** **13** 20 → $|M_{max}| \leq 2|M_{inkl}|$

10) Jeder k-reguläre Graph mit $k \geq 1$ enthält ein perfektes Matching.
→ Gegenbeispiel: 

11) Sei G ein Graph mit einem nicht-perfekten Matching, zu dem es keinen augmentierenden Pfad gibt. Dann gibt es kein perfektes Matching in G.
→ Wahr

12) Ist ein Matching inklusionsmax., gibt es keine augmentierenden Pfade mehr.
→ Falsch,