Kap. I.2: Matching in Graphen

Teil 1: Bipartite Graphen

Professor Dr. Petra Mutzel

Abteilung für Computational Analytics
Institut für Informatik (Abt. 1)
Universität Bonn

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

INSTITUT FÜR INFORMATIK DER UNIVERSITÄT BONN

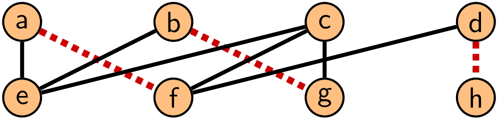
SoSe 2024 VO 3 23. April 2024

Outline

Table of Contents

Table of Contents

Motivation



Matching-Probleme fallen in die Klasse der Zuordnungsprobleme

- Zuordnung von Medizinstudierenden zu Krankenhäusern in den USA (Heiratsproblem, Prioritäten)
- Zuordnung von Studierenden zu Übungsgruppen (gewichtetes perfektes Matching)
- Zuordnung von Lehrveranstaltungen zu Räumen (maximales Matching)
- Preisfindung bei Auktionen (maximalen Gesamtgewinn)
- Satellitenkommunikation (Zeitschlitz-Zuordnungsproblem),
- ...

Motivation

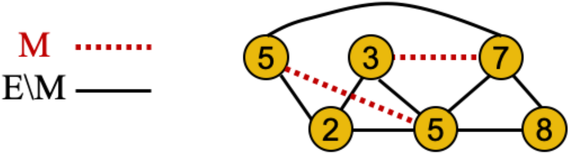
Matching Varianten tauchen in vielen Algorithmen und Problemen auf

- Travelling Salesman Problem: Christofides-Heuristik
- Chinese Postman Problem (kürzester Zyklus in Graph, der jede Kante mindestens einmal durchläuft, z.B. Briefträger, Müllabfuhr)
- Maximaler Schnitt in planaren Graphen (meine Diplomarbeit)
- Controllability von Netzwerken
- Steganographie (Verstecken geheimer Informationen in Bildern)
- ...

Voraussetzung hier: Graph ohne Schleifen und ohne Mehrfachkanten

Table of Contents

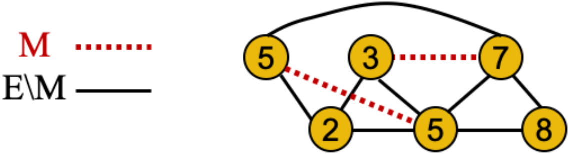
Zentrale Begriffe zu Matching



Definition

- Eine Kantenmenge $M \subseteq E$ in einem ungerichteten Graphen $G = (V, E)$ heißt Matching, falls jeder Knoten in V mit höchstens einer Kante in M inzidiert.
- Eine Kante in $E \setminus M$ heißt frei.
- Ein Matching M überdeckt einen Knoten $v \in V$, falls ein $e \in M$ mit v inzidiert, andernfalls ist v ein exponierter (bzw. freier) Knoten.

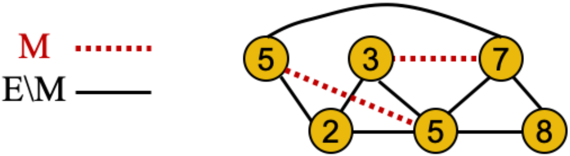
Zentrale Begriffe zu Matching



Definition

- D.h., für ein Matching M haben wir $2|M|$ überdeckte und $|V| - 2|M|$ exponierte Knoten.
- Mit $\nu(G)$ bezeichnen wir die Kardinalität eines maximum Matchings in G (d.h. das größte Matching bezüglich der Kantenanzahl).
- $\mathrm{def}(G) = |V| - 2\nu(G)$ ist dann die kleinste Anzahl exponierter Knoten in G („Defizit“).
- Ein perfektes Matching überdeckt alle Knoten in G.

Graph Matching Probleme



Wir betrachten die folgenden Probleme

- Hat G ein perfektes Matching?
- Finde ein maximum Matching in G.

Weitere Matching Probleme in Graphen

Falls zusätzliche Gewichte w für die Kanten $e \in E$ gegeben sind, dann ist das Gewicht eines Matchings gegeben als $\sum_{e \in M} w(e)$.

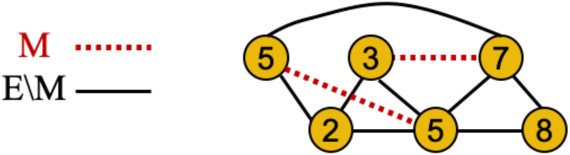
Gewichtete Matchings

- Das minimum gewichtete perfekte Matching Problem: ein perfektes Matching mit kleinstem Gewicht in G
- Das (perfekte) gewichts-maximale Matching Problem: Ein (perfektes) Matching mit möglichst großem Gewicht

Bemerkung: Gewichtete (allg.) Matchings werden hier nicht behandelt, weil Algorithmen deutlich komplexer

Maximale vs. Maximum Matchings

In der Literatur gibt es manchmal zwei unterschiedliche Bedeutungen von Maximales Matching.



Mögliche Bedeutungen von Maximales Matching

- Maximum Matching
- Matching, das nicht mehr durch Hinzunahme weiterer Kanten aus E erweitert werden kann.

Das rot gezeichnete Matching im Beispiel wäre demnach ein maximales Matching aber kein maximum Matching.

Table of Contents

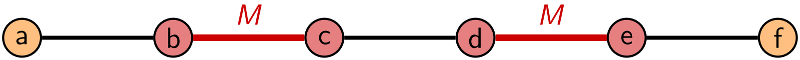
Table of Contents

Matchings und alternierende Pfade

Definition

Sei $G = (V, E)$ ungerichteter Graph, $M \subseteq E$ Matching auf G.

- Ein nicht-leerer Pfad $P = (v_0, \ldots, e_k, v_k)$ mit $k \geq 1$ bei dem sich M-Kanten und freie Kanten *abwechseln* heißt M-alternierender Pfad.
- Ein *kreisfreier* M-alternierender Pfad $P = (v_0, \ldots, e_k, v_k)$, $k \geq 1$, mit v_0 und v_k frei heißt M-verbessernd (oder M-augmentierend).



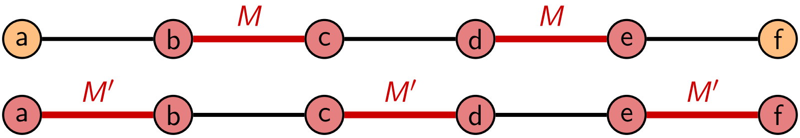
Ein M-augmentierender Pfad ist also auch immer ein Weg.

Augmentierende Pfade

Definition (Augmentierung eines M-augmentierenden Pfades P)

Sei $G = (V, E)$ ungerichteter Graph, $M \subseteq E$ Matching in G und P ein M-augmentierender Pfad.

- P ist also ein kreisfreier nicht-leerer Pfad $P = (v_0, \ldots, e_k, v_k)$ mit $k \geq 1$ bei dem sich M-Kanten und freie Kanten *abwechseln* und v_0 sowie v_k frei sind.
- Das Ergebnis der Augmentierung von M entlang des Pfades P ist: $M' = M \triangle P$ (wobei $\triangle$ die symmetrische Differenz ist).
- M' ist wiederum ein Matching mit $|M'| = |M| + 1$.



Theorem von Berge

Theorem (Berge 1957)

Ein Matching M in G ist maximum genau dann, wenn kein M-augmentierender Pfad existiert.

Beweis Wir zeigen: Ein Matching M ist nicht maximum g.d.w. ein M-augmentierender Pfad existiert.

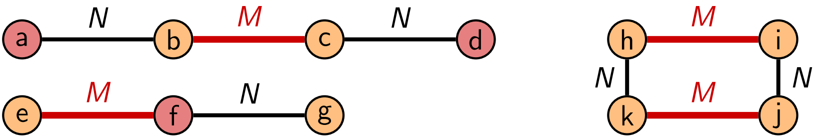
$\Leftarrow$: Gibt es einen M-augmentierenden Pfad P, dann enthält die symmetrische Differenz $M' = M \triangle E(P)$ eine Kante mehr und überdeckt zwei Knoten mehr.

Deshalb kann M nicht maximum sein.

Theorem von Berge ff

Beweis ff $\Rightarrow$:

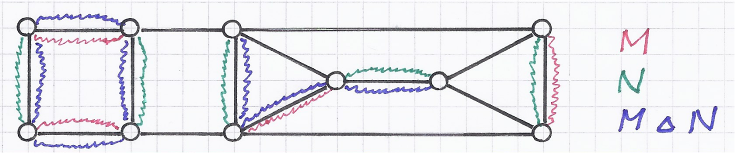
- Ist M nicht maximum, so existiert ein Matching N mit $|N| > |M|$.
- betrachte $J = M \triangle N = \{e \mid e \in M \wedge e \notin N\} \cup \{e \mid e \notin M \wedge e \in N\}$
- in J sind alle Knotengrade ≤ 2
- also zerfällt J in knotendisjunkte alternierende Pfade und alternierende Kreise
- Kreise haben gerade Länge



Theorem von Berge ff

Beweis ff $\Rightarrow$:

- Ist M nicht maximum, so existiert ein Matching N mit $|N| > |M|$.
- betrachte $J = M \triangle N = \{e \mid e \in M \wedge e \notin N\} \cup \{e \mid e \notin M \wedge e \in N\}$
- in J sind alle Knotengrade ≤ 2
- also zerfällt J in knotendisjunkte alternierende Pfade und alternierende Kreise
- Kreise haben gerade Länge
- Da $|N| > |M|$, existiert ein Pfad in J, der mehr N-Kanten als M-Kanten besitzt. Dieser ist ein M-augmentierender Pfad in G. $\square$



Algorithmus-Idee

Einfacher Algorithmus basierend auf Berge

- Bestimme ein Start-Matching M (z.B. $M = \emptyset$)
- Wiederhole:
-     Bestimme M-augmentierenden Pfad P
-     Augmentiere M mit P: $M \leftarrow M \triangle P$
- bis kein augmentierender Pfad mehr existiert.

Probleme dabei:

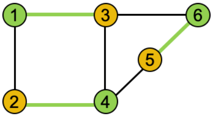
- Wie findet man M-augmentierende Pfade?
- Garantie, dass kein M-augmentierender Pfad mehr existiert
- Hierbei hilfreich: Schranken für maximum Matchings

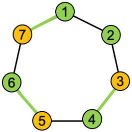
Table of Contents

Schranken für Maximum Matchings

Definition

Eine Knotenüberdeckung (vertex cover) in einem Graphen $G = (V, E)$ ist eine Teilmenge $A \subseteq V$, so dass jede Kante in E wenigstens einen Endknoten in A hat.





Lemma

Seien M ein Matching und A eine beliebige Knotenüberdeckung in G. Dann gilt: $|M| \leq |A|$.

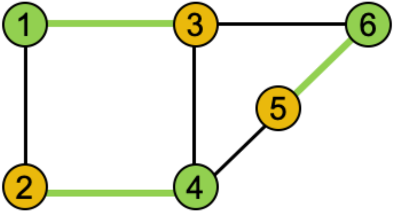
Beweis: A deckt jede Kante, also auch jede M-Kante ab. Da die M-Kanten knotendisjunkt sind, wird für jede M-Kante mindestens ein Knoten aus A benötigt.

Theorem von König

Für die Klasse der bipartiten Graphen gilt sogar etwas stärkeres:

Theorem (König (ohne Beweis))

In einem bipartiten Graphen $G = (V, E)$ gibt es eine Knotenüberdeckung A und ein Matching M mit $|A| = |M|$.



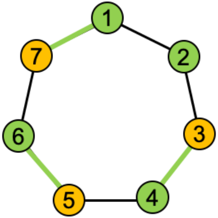
ACHTUNG: Das Theorem von König gilt nicht für allgemeine G.

Ungerade Kreise

Wir betrachten einen ungeraden Kreis der Länge $2h + 1$.

Hier gilt offensichtlich für jedes Matching $|M| \leq h$ und jede Knotenüberdeckung $|A| > h$.

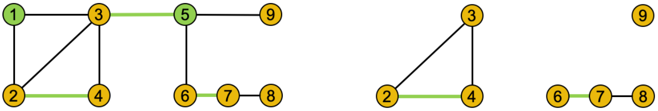
Z.B. für $h = 3$:



$\Rightarrow$ Wir suchen eine bessere Schranke.

Tutte-Berge-Schranke

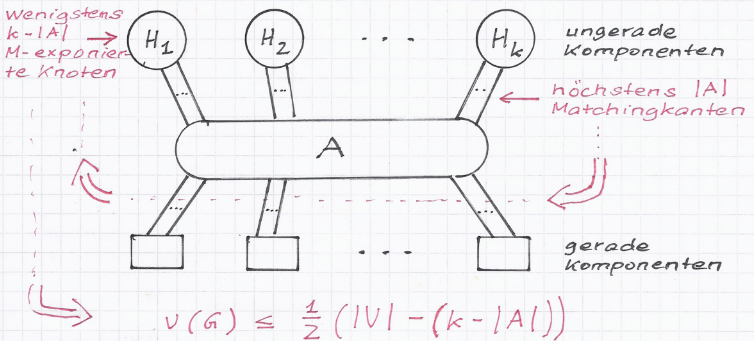
- Sei $A \subseteq V$ eine beliebige Knotenmenge und M beliebiges Matching
- $G \setminus A$ zerfällt in mehrere Zusammenhangskomponenten
- Ungerade Komponente ist eine Zusammenhangskomponente mit ungerader Knotenanzahl
- Seien $H_1, H_2, \ldots, H_k$ die ungeraden Komponenten von $G \setminus A$



- Für jedes H_i gilt: H_i enthält mindestens einen ungematchten Knoten oder M enthält eine Kante zwischen H_i und A
- Falls $k > |A|$ gibt es also mindestens $(k - |A|) > 0$ viele ungematchte Knoten

Tutte-Berge-Schranke

Sei $A \subseteq V$ eine beliebige Knotenmenge. $G \setminus A$ habe k Komponenten $H_1, H_2, \ldots, H_k$ mit $|V(H_i)|$ ungerade:

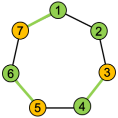


Wenn $\mathrm{oc}(G)$ die Anzahl ungerader Komponenten von G bezeichnet, so erhalten wir die Tutte-Berge-Schranke

$$\nu(G) \leq \frac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|).$$

Tutte-Berge-Schranke

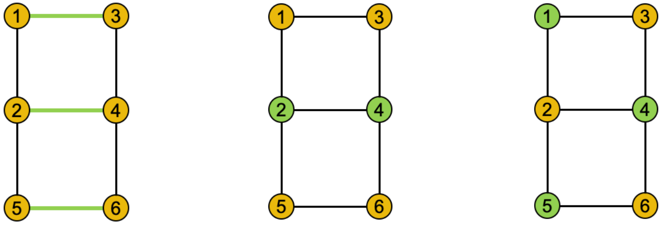
- Ist insbesondere A eine Knotenüberdeckung, so besteht $G \setminus A$ aus einzelnen Knoten.
- Dies sind $|V| - |A|$ Knoten, die ungerade Komponenten bilden, also:
- $\nu(G) \leq \frac{1}{2}(|V| - (|V| - |A|) + |A|) = |A|$.
- D.h. die neue Schranke ist wenigstens so gut wie die Überdeckungsschranke.

Sie ist sogar besser: 

- Für einen ungeraden Kreis der Länge $2h + 1$ wählen wir $A = \emptyset$ und erhalten
- $\nu(G) \leq \frac{1}{2}(|V| - 1) = h$. (bisher nur „$\leq h + 1$“).

Tutte-Berge-Schranke

Kleines Beispiel



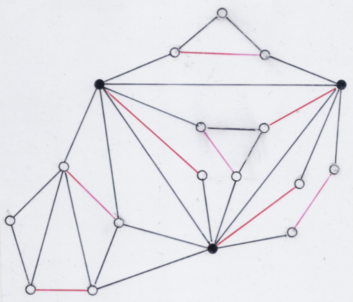
Hier ist offensichtlich $\nu(G) = 3$. Die „schlechte" Wahl von A (grüne Knoten) im mittleren Bild ergibt

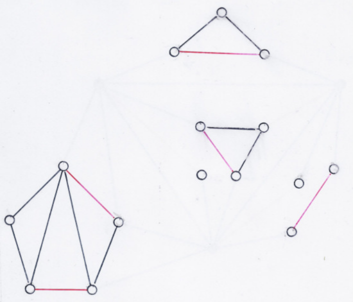
$$\nu(G) \leq \tfrac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|) = \tfrac{1}{2}(6 - 0 + 2) = 4,$$

die „gute" Wahl von A im rechten Bild ergibt

$$\nu(G) \leq \tfrac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|) = \tfrac{1}{2}(6 - 3 + 3) = 3.$$

Tutte-Berge-Schranke: Größeres Beispiel





Wir haben $|V| = 18$, $|M| = 8$ (rote Kanten: zwei exponierte Knoten), $|A| = 3$ (schwarze Knoten) und $\mathrm{oc}(G \setminus A) = 5$.
Wir wenden die Tutte-Berge-Schranke an und erhalten

$$\nu(G) \leq \tfrac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|) = \tfrac{1}{2}(18 - 5 + 3) = 8.$$

Damit ist bewiesen, dass M ein Matching mit maximum Kardinalität ist.

Tutte-Bedingung

Aus der Tutte-Berge-Schranke für $G = (V, E)$ und $A \subseteq V$

$$\nu(G) \leq \frac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|)$$

erhalten wir die Tutte-Bedingung als notwendige Bedingung für die Existenz eines perfekten Matchings.

Tutte-Bedingung

Hat $G = (V, E)$ ein perfektes Matching, so gilt für jede Teilmenge $A \subseteq V$: $\mathrm{oc}(G \setminus A) \leq |A|$.

Wir werden später sehen, dass die Tutte-Berge-Schranke angenommen wird und die Tutte-Bedingung auch hinreichend ist.

Table of Contents

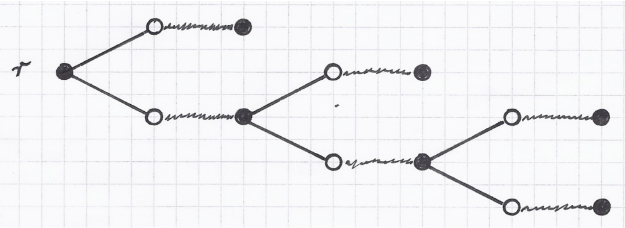
Table of Contents

Alternierende Bäume

Definition

Sei $G = (V, E)$, $M \subseteq E$ ein Matching und r ein exponierter Knoten. Ein M-alternierender Baum ist ein Baum T mit Wurzel r, in dem gilt:

- für jedes $v \in V(T)$ ist der (r, v)-Pfad in T M-alternierend
- alle Knoten $v \neq r$ in T sind überdeckt mit M-Kanten in T.



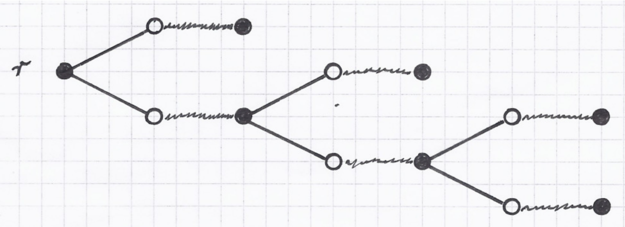
Alternierende Bäume

Wir unterscheiden die ungeraden Knoten

$$\text{„}\circ\text{“} \quad A(T) = \{v \in V(T) \mid v \text{ hat ungerade Distanz von } r\}$$

und die geraden Knoten

$$\text{„}\bullet\text{“} \quad B(T) = \{v \in V(T) \mid v \text{ hat gerade Distanz von } r\}.$$



Für alternierende Bäume T gilt: $|B(T)| = |A(T)| + 1$

Baumerweiterung

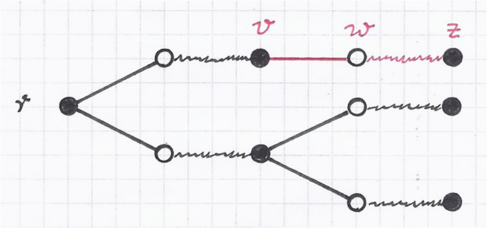
Wir definieren zwei Unterprogramme, die auf alternierenden Bäumen operieren. Das erste dient der Baumerweiterung:

Benutze vw zur Baumerweiterung

Eingabe: Matching M von G,
M-alternierender Baum T in G,
$vw \in E(G)$ mit $v \in B(T)$, $w \notin V(T)$, w ist M-überdeckt

Sei wz die w überdeckende Kante in M.
$T \longleftarrow (V(T) \cup \{w, z\}, E(T) \cup \{vw, wz\})$;

Skizze



Baum Augmentierung

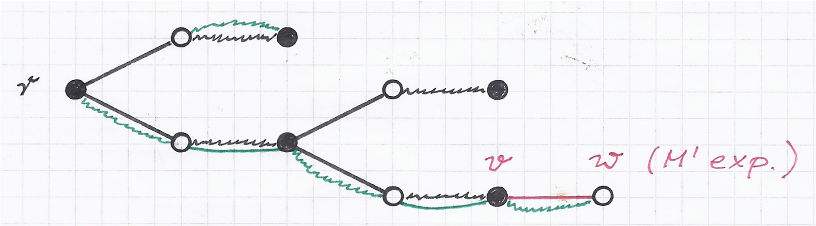
Das zweite dient der Augmentierung:

Benutze vw zur M-Augmentierung

Eingabe: Matching M von G,
M-alternierender Baum T in G mit Wurzel r,
$vw \in E(G)$ mit $v \in B(T)$, $w \notin V(T)$, w ist M-exponiert

Sei P der r-w-Pfad in $T \cup \{vw\}$.
$M \longleftarrow M \triangle E(P)$;

Skizze



Perfektes Matching

Definition

Ein M-alternierender Baum T in G ist frustriert, falls jede Kante in $E(G)$ mit einem Ende in $B(T)$ das andere Ende in $A(T)$ hat.

Lemma (Frustrierter M-alternierender Baum (Lemma 2))

Hat G ein Matching M und einen frustrierten M-alternierenden Baum T, so hat G kein perfektes Matching.

Beweis Für jedes $v \in B(T)$ ist $\{v\}$ eine ungerade Komponente von $G \setminus A(T)$

D.h. für $A = A(T) \subseteq V$ gilt:

$$\mathrm{oc}(G \setminus A) \geq |B(T)| > |A(T)| = |A|$$

Durch Anwendung der Tutte-Bedingung erhalten wir, dass G kein perfektes Matching besitzt. $\square$

Table of Contents

Perfektes Matching für bipartite Graphen

Diese Überlegungen führen zu einem Algorithmus für den bipartiten Fall:

Perfektes Matching Algorithmus für bipartite Graphen (PMB)

$M \longleftarrow \emptyset$;
Wähle beliebigen Knoten $r \in V(G)$;
$T \longleftarrow (\{r\}, \emptyset)$;
while $(\exists\ vw \in E$ mit $v \in B(T),\ w \notin V(T))$ {
    if (w ist M-exponiert) {
        Benutze vw zur M-Augmentierung;
        if ($\nexists$ M-exponierter Knoten in G)
            STOP „M ist ein perfektes Matching“;
        else $T \longleftarrow (\{r\}, \emptyset)$ für einen M-exponierten Knoten r;
    }
    else Benutze vw zur Baumerweiterung;
}
STOP „G hat kein perfektes Matching“;

Korrektheit im bipartiten Fall

Lemma (Korrektheit des Algorithmus PMB (Lemma 3))

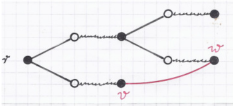
Ist im Algorithmus PMB die **while***-Bedingung nicht erfüllt, so hat G kein perfektes Matching.*

Beweis Wir zeigen, dass T frustriert ist, dann folgt mit Lemma 2, dass kein perfektes Matching existiert.

Die **while**-Bedingung sei nicht erfüllt: Es gibt kein $vw \in E$ mit $v \in B(T)$ und $w \notin V(T)$.

Also gilt für alle $vw \in E$ mit $v \in B(T)$: $w \in A(T)$ oder $w \in B(T)$.

Wäre $w \in B(T)$, so hätten wir einen ungeraden Kreis in G und G wäre nicht bipartit:



Also gilt $w \in A(T)$ für alle $vw \in E$ mit $v \in B(T)$.

Deshalb ist T frustriert und wir können mit Lemma 2 schließen, dass G kein perfektes Matching besitzt. $\square$

Laufzeit des Algorithmus

Theorem (Terminierung des Algorithmus PMB)

Sei $n = |V|$. Der Algorithmus terminiert nach

- $O(n)$ Augmentierungen sowie
- $O(n^2)$ Baumerweiterungs-Schritten.

Beweis

Mit jeder Augmentierung erniedrigt sich die Anzahl der exponierten Knoten um zwei. Also gibt es $O(n)$ Augmentierungen mit jeweils $O(n)$ Baumerweiterungsschritten.

Korrektheit und Laufzeit des Algorithmus

Theorem (Korrektheit und Laufzeit des Algorithmus PMB)

Sei $G = (V, E)$ ein bipartiter Graph. Der Algorithmus PMB berechnet ein perfektes Matching, wenn G ein solches besitzt. Andernfalls stoppt er mit einem frustrierten Baum als Beweis, dass kein solches existiert. Der Algorithmus PMB kann in Laufzeit $O(|V||E|)$ realisiert werden.

Beweis

M ist stets ein Matching. Wenn der Algorithmus terminiert, dann entweder mit einem Perfekten Matching oder mit einem frustrierten Baum (besitzt dann also kein Perfektes Matching (Lemma 3)).

Realisierung der einzelnen Schritte mit Hilfe von Datenstrukturen: siehe Übungen □

Table of Contents

Algorithmus für Maximum Matching (MMB)

Idee: Ableitung aus Algorithmus PMB für perfektes Matching:

1. Starte Algorithmus PMB für aktuelles G'
2. Stoppt dieser mit perfektem Matching $\Rightarrow$ fertig.
3. Stoppt dieser mit Matching M' in G' und einem frustrierten Baum T, so ist das zu M' korrespondierende Matching M in G nicht notwendigerweise ein maximum Matching in G. Dann:
   - Wir entfernen $V(T)$ aus G'.
   - Falls ein exponierter Knoten übrig ist, wenden wir PMB auf das neue G' an. $\Rightarrow$ Gehe zu (1)
4. Wiederhole (1)-(3), bis kein exponierter Knoten übrig bleibt.
5. Wir restaurieren das Original G mit allen produzierten Matchingkanten.
6. Wir bestimmen das korrespondierende Matching M in G.

Korrektheit von MMB für Maximum Matching

Seien $T_1, T_2, \ldots, T_k$ die generierten frustrierten Bäume.

Die Wurzeln dieser Bäume sind k exponierte Knoten in G' und G.

Also gilt $|M| = \frac{1}{2}(|V| - k)$.

Sei $A := \bigcup_{i=1}^{k} A(T_i)$. Die Entfernung von A aus G ergibt eine ungerade Komponente für jeden Knoten aus $B(T_i)$ für alle $i \in \{1, 2, \ldots, k\}$.

Für alle $i \in \{1, 2, \ldots, k\}$ haben wir: $|B(T_i)| = |A(T_i)| + 1$

$$\begin{aligned} \mathrm{oc}(G \setminus A) &\geq \sum_{i=1}^{k} |B(T_i)| \\ &= \sum_{i=1}^{k} |A(T_i) + 1| \\ &= |A| + k \end{aligned}$$

Korrektheit für Maximum Matching ff

Mit $\mathrm{oc}(G \setminus A) \geq |A| + k$ folgt:

$$\begin{aligned}
\tfrac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|) &\leq \tfrac{1}{2}(|V| - |A| - k + |A|) \\
&= \tfrac{1}{2}(|V| - k) \\
&= |M|
\end{aligned}$$

Mit der Tutte-Berge-Schranke $|M| \leq \frac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|)$ erhalten wir

$$|M| = \tfrac{1}{2}(|V| - \mathrm{oc}(G \setminus A) + |A|).$$

Deshalb ist M ein maximum Matching in G. $\square$