5.4 Der Blossom-Algorithmus

12.07.11

Dazu sei

~~Verallgemeinerung auf allgemeine Graphen!~~

Ausgangsgraph: G

Abgeleiteter Graph: G' (durch ein Serie von Schrumpfungen ungerader Kreise entstanden)

Knotentypen in G': „Originalknoten"
„Pseudoknoten" ⇐ repräsentieren ungerade ~~Knotenmengen~~, die geschrumpft wurden

Also: Jeder Knoten $v \in V(G')$ repräsentiert eine ungerade Menge von Knoten $S(v) \subseteq V(G)$.

Klar: $S(v)$ ist immer ungerade! (ungerade Vereinigung ungerader Mengen).

Satz 5.14

Sei G' ein ~~abgeleiteter~~ Graph von G, sei M' ein Matching in G', und sei T ein M'-alternierender Baum von G', so dass kein Element von $S(T)$ ein Pseudoknoten ist.

Baum ist frustriert

Wenn jede Kante mit einem Ende in $W(T)$ das andere Ende in $S(T)$ hat, dann hat G kein perfektes Matching.

Beweis:

Entfernt man $S(T)$ aus G, dann erhält man im ursprünglichen Graphen aus jedem $v \in W(T)$ eine ungerade Komponente $S(v)$. Also gilt $oc(G \setminus S(T)) > |S(T)|$, d.h. G hat kein perfektes Matching. $\square$

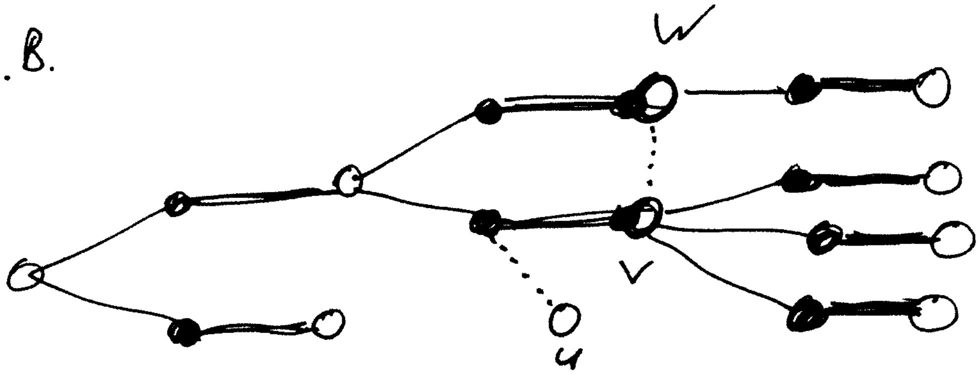
Also:

- abgeleiteter Graph hat perfektes Matching
  → liefert perfektes Matching im ursprünglichen Graphen!
- abgeleiteter Graph liefert Widerspruch
  → kein perfektes Matching in ursprünglichem Graphen!

Wie findet man ungerade Kreise zum Schrumpfen

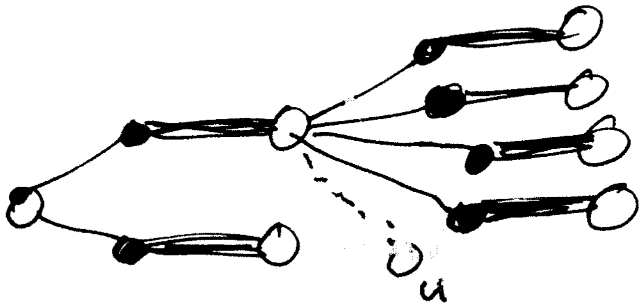
Schlüssel: Kanten von $W(T)$ nach $W(T)$!

Betrachte z. B.



Wegen Kante $\{v,w\}$ ist Baum nicht frustriert!
→ Kante liefert ungeraden Kreis!

Nach Schrumpfen:



Algorithmus 5.15 (Blossom-Algorithmus für perfektes Matching)

Eingabe: $G=(V,E)$

Ausgabe: Perfektes Matching M

oder ein Beweis, dass keines existiert

(1) Setze $M' = M = \emptyset$, $G' = G$

(2) Wähle einen ungematchten Knoten r in G',
setze $T = (\{r\}, \emptyset)$, $W(T) = \{r\}$, $S(T) = \emptyset$

(3) WHILE (es gibt $\{v,w\} \in E'$ mit $v \in W(T)$, $w \notin A(T)$) Do

CASE: w ist ungematcht

Benutze $\{v,w\}$, um M' zu augmentieren

Erweitere M' zu einem Matching M von G

Ersetze M' durch M, G' durch G

IF (es gibt keinen ungematchten Knoten in G)

RETURN perfektes Matching M in G

ELSE

Ersetze T durch $(\{r\}, \emptyset)$ mit r
ungematcht in M'

Case: $w \notin V(T)$, w ist in M' gematcht

Benutze $\{v,w\}$ um T zu erweitern.

Case: $w \in W(T)$

Bentze $\{v,w\}$ zum Schrumpfen

und Aktualisieren von M' und T.

RETURN G', M', T : G hat kein perfektes Matching

<u>Bentze $\{v,w\}$ zum Schrumpfen und Aktalisieren von M' und T</u>

Eingabe: Matching M' eines Graphen G',

M'-alternierender Baum T,

Kante $\{v,w\}$ von G' mit $v, w \in W(T)$

Sei C der Kreis, den $\{v,w\}$ zusammen mit dem Pfad von v nach w in T bildet.

Ersetze G' durch G'/C ← C geschrumpft,

M' durch $M' \setminus E(C)$

T durch den Baum (in G')

mit Kantenmenge $E(T) \setminus E(C)$

Satz 5.16

Algorithmus 5.15 terminiert nach (a) $O(n)$ Augmentierungen,
(b) $O(n^2)$ Schrumpfungen,
(c) $O(n^2)$ Baumerweiterungen.

Er entscheidet in korrekter Weise, ob G ein perfektes Matching hat.

Beweis:

M' ist immer ein Matching; da jede Augmentierung die Zahl der ungematchten Knoten verringert, ist (a) klar.

Zwischen Augmentierungen verringert jede Schrumpfung die Zahl der Knoten in G', während jede Baumerweiterung die Zahl der Knoten nicht in $V(T)$ verringert. Also ist diese Zahl von Schritten zwischen Augmentierungen durch $O(n)$ begrenzt, was (b) und (c) impliziert.

Wenn der Algorithmus terminiert, dann ist das Ergebnis nach Satz 6.12 korrekt. $\square$

Maximales Matching

Am Ende von Algorithmus 5.14 haben wir:

- einen frustrierten Baum T:
  - die Wurzel ist ungematcht
  - Entfernen der schwarzen Knoten zeigt, dass ~~innerhalb des~~ ~~Knoten des~~ ~~Baumes~~ ~~kein~~ ~~gefundenes~~ nicht für alle weißen Knoten Matching-Partner gefunden werden können.

Sind wir damit fertig?

Nein! Es kann anderswo noch ungematchte Knoten geben, für die sich Nachbarn finden lassen. Dort müssen wir weitersuchen.

Algorithmus 5.17

Wie Algorithmus 5.15;

Am Ende: Entferne $V(T)$ aus G.

~~falls~~ if (Es gibt einen ungematchten Knoten)

CONTINUE mit Algorithmus 5.15

Satz 5.18

Algorithmus 5.17 bestimmt ein maximales Matching.

Beweis:

Angenommen, wir entfernen nacheinander die Bäume $T_1, \dots, T_k$. Dann sind in G am Ende nur die k Wurzeln ungematcht.

Entfernen aller schwarzen Knoten aus allen Bäumen hinterlässt für jeden Baum eine weiße ungerade Komponente mehr, also haben wir

$$oc(G \setminus S) = |S| + k$$

Da wir ein Matching der Größe $|S|$ haben, ist dieses optimal. $\square$

Bemerkung ~~KOROLLAR~~ **5.19**

Aus dem obigen kann man einen algorithmischen Beweis der Tutte-Berge-Formel ableiten.

Laufzeit:

Satz 5.20

Der Blossom-Algorithmus kann mit Laufzeit $O(n\,m \log n)$ implementiert werden.

Erläuterung:

Cook / Cunningham / Pulleyblank / Schrijver S. 141/42.