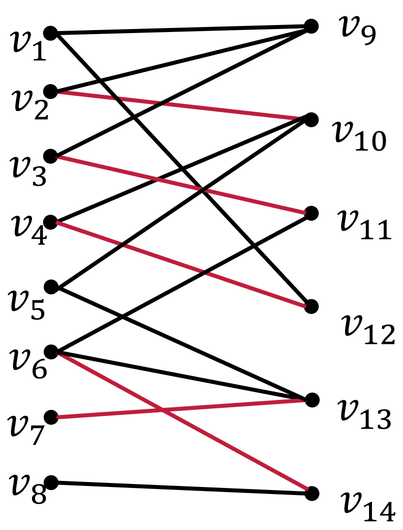
Technische
Universität
Braunschweig

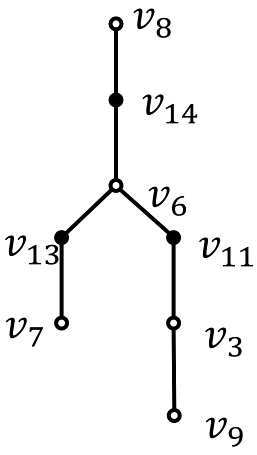
Netzwerkalgorithmen – Vorlesung #11

Arne Schmidt

Wiederholung

Bipartite Graphen





Satz 5.7 (Kőnig-Egerváry)
Sei G ein bipartiter Graph. Die Größe eines minimalen Vertex Covers in G entspricht der Größes eines maximalen Matchings in G.

Satz 5.13 (Satz von Hall)
Sei $G = (V, E)$ ein bipartiter Graph mit $V = V_1 \,\dot\cup\, V_2$. Dann hat G genau dann ein V_1 überdeckendes Matching, wenn $|N(X)| \geq |X|$ für alle $X \subseteq V_1$ gilt.

5.2 Allgemeine Graphen

Idee

Nutze die gleiche Struktur wie bei bipartiten Graphen:

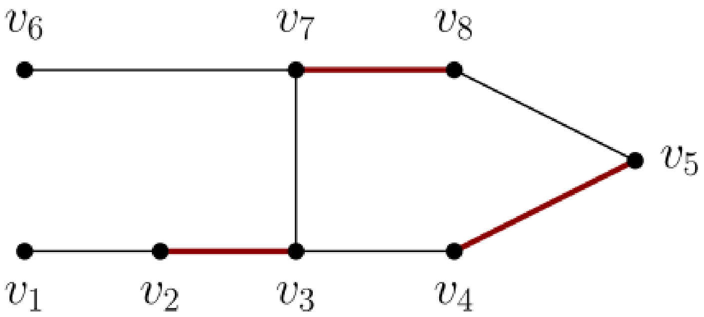
Konstruiere Baum mit abwechselnd Matchingkanten und Nicht-Matchingkanten

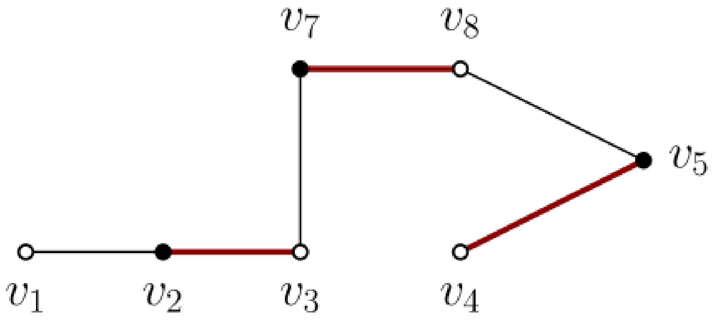
Augmentierende Pfade

Definition 5.15
Sei $G = (V, E)$ ein Graph und M ein Matching in G.

- Ein Pfad P heißt ***M*-alternierend**, wenn $E(P) \setminus M$ ein Matching ist.

- Ein M-alternierender Pfad heißt ***M*-augmentierend**, wenn für beide Endknoten v, w des Pfades keine Kante $e \in M$ existiert, die zu v oder w inzident ist.

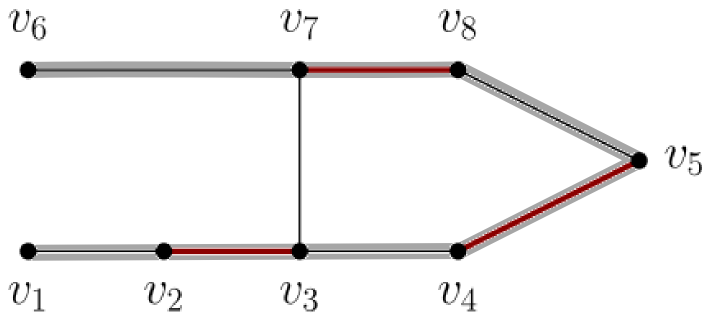


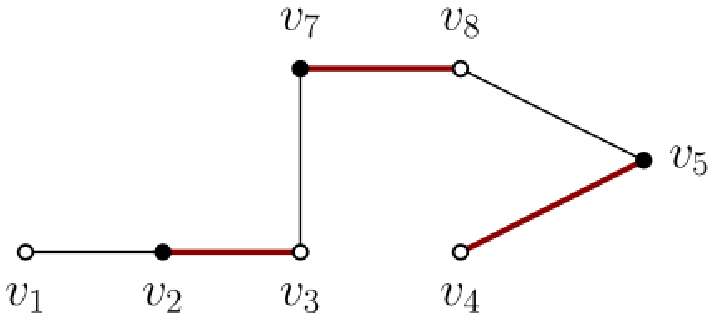


Nicht augmentierend!

Augmentierende Pfade

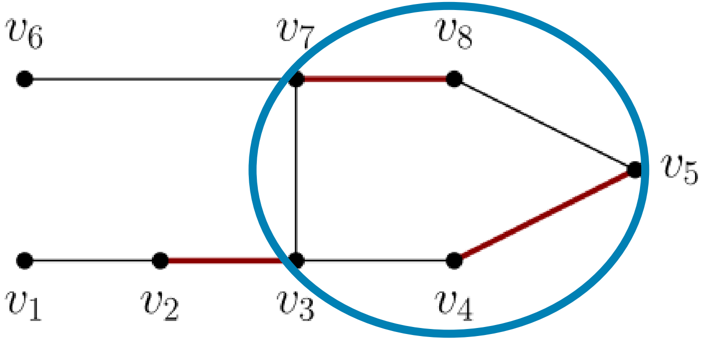
Satz 5.16
Sei $G = (V, E)$ ein Graph und M ein Matching in G.
M ist genau dann maximal, wenn kein M-augmentierender Pfad existiert.





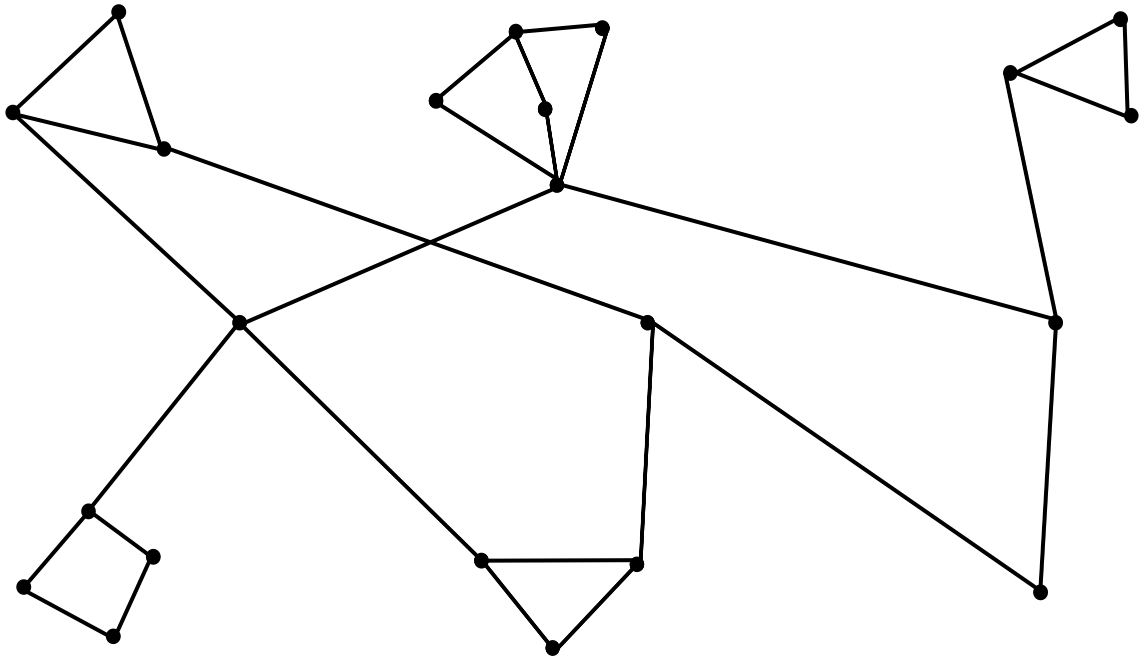
Augmentierende Pfade

Satz 5.16
Sei $G = (V, E)$ ein Graph und M ein Matching in G.
M ist genau dann maximal, wenn kein M-augmentierender Pfad existiert.

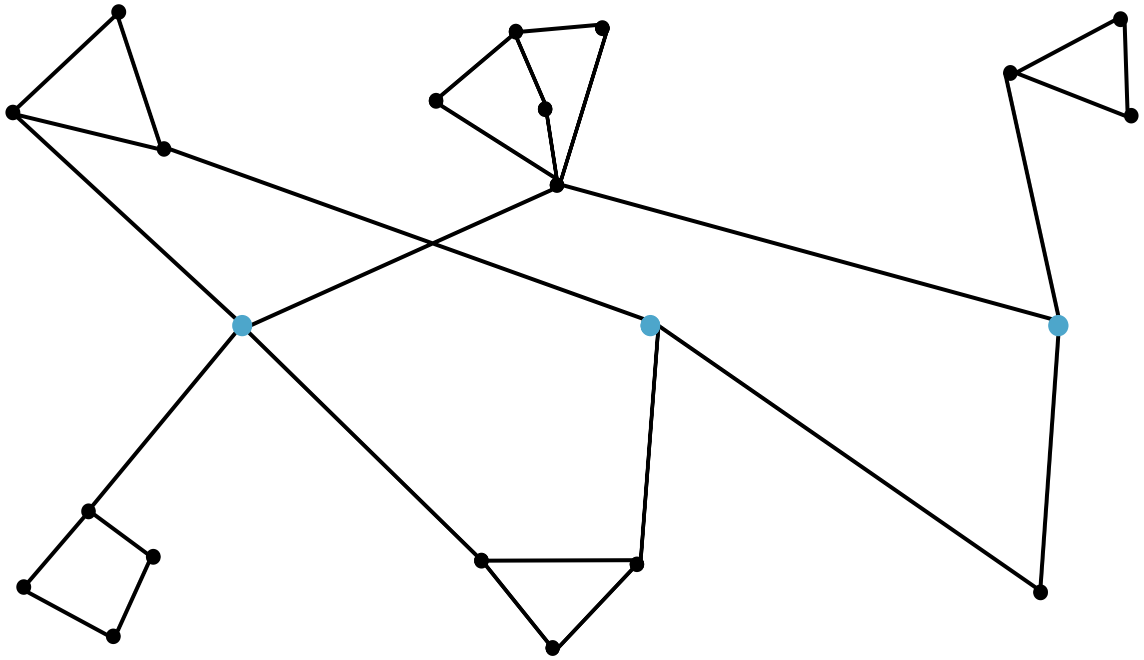


Braucht einen Knoten außerhalb für ein perfektes Matching.

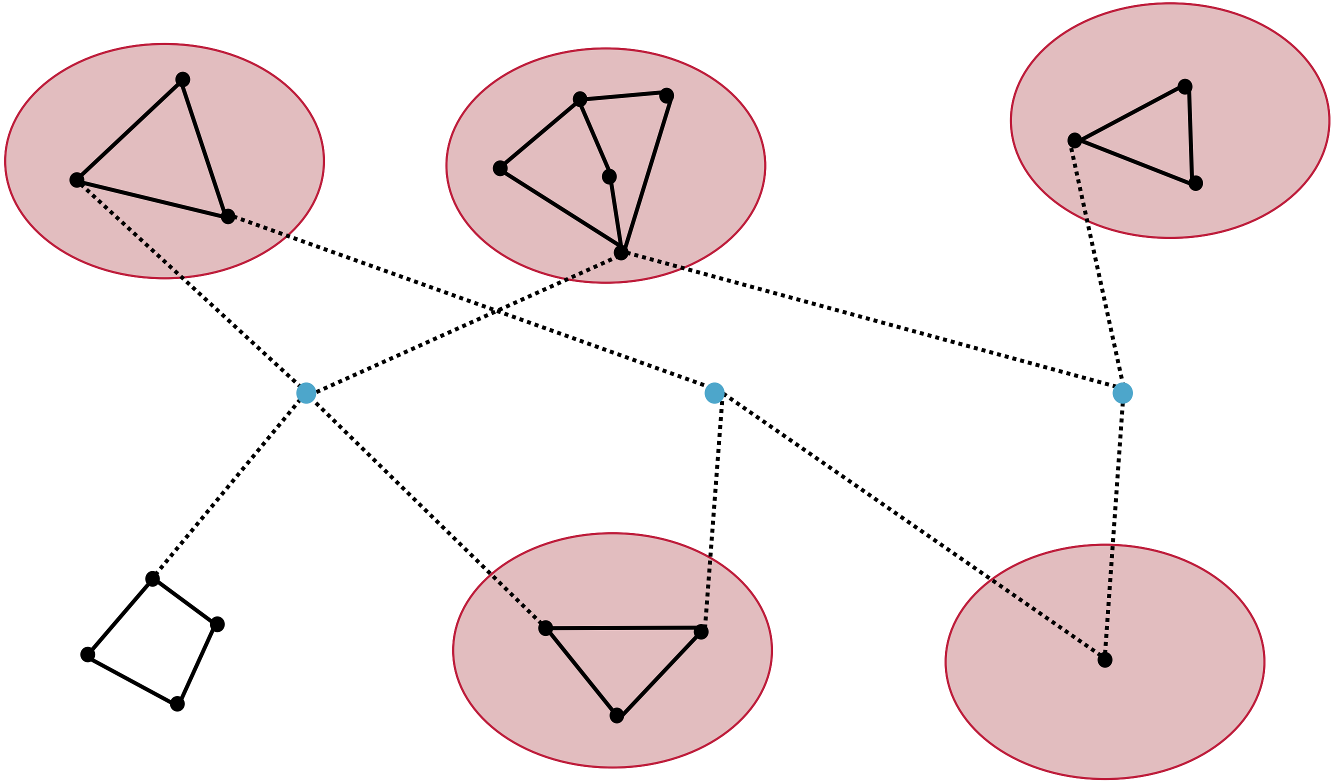
Bedingung für perfekte Matchings



Bedingung für perfekte Matchings



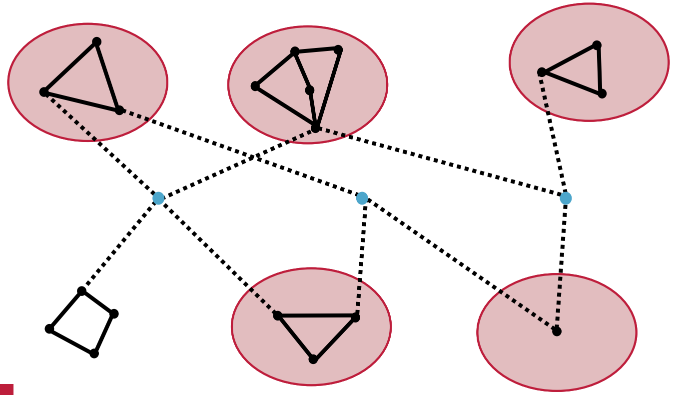
Bedingung für perfekte Matchings



Satz von Tutte

Definition 5.17
$\mathrm{oc}(G)$ bezeichnet die Anzahl an ZHK mit ungerade vielen Knoten in einem Graphen G.

Satz 5.18
Sei $G = (V, E)$ ein Graph.
G enthält genau dann ein perfektes Matching, wenn $\mathrm{oc}(G \setminus A) \leq |A|$ für alle $A \subseteq V$ gilt.

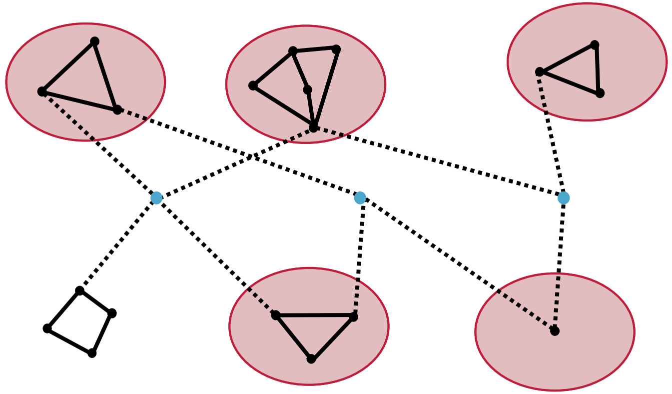


Im Beispiel:
$|A| = 3 < \mathrm{oc}(G \setminus A) = 5$

Tutte-Berge-Formel

Satz 5.19
Sei $G = (V, E)$ ein Graph. Dann ist

$$\max_{\text{Matching } M} |M| = \min_{A \subseteq V} \frac{1}{2}\big(|\mathrm{V}| - (\mathrm{oc}(G \setminus A) - |A|)\big)$$

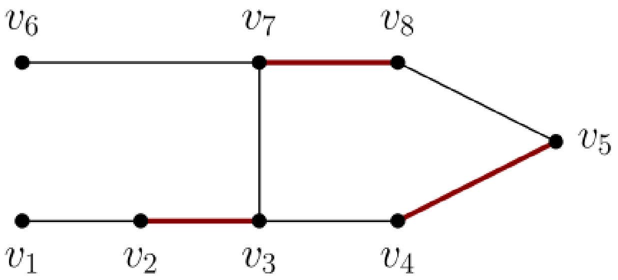


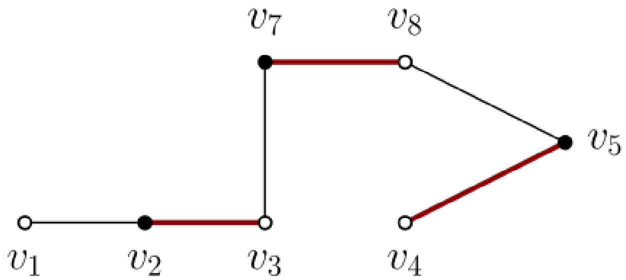
Im Beispiel:

$$\begin{aligned}\mathrm{MaxMat} &\leq \tfrac{1}{2}\big(|V| - (\mathrm{oc}(G \setminus A) - |A|)\big)\\ &= \frac{1}{2}\big(22 - (5 - 3)\big)\\ &= 10\end{aligned}$$

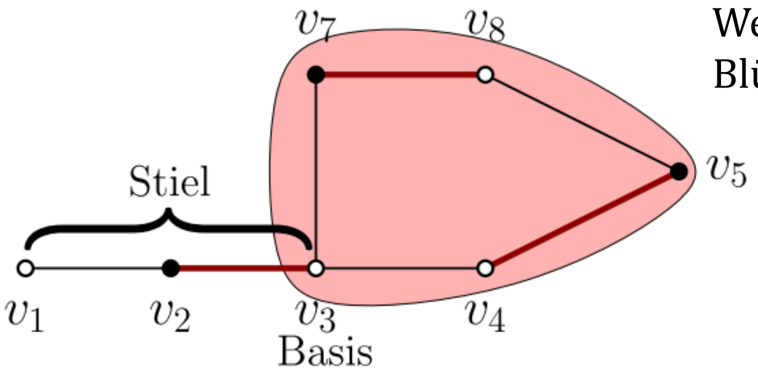
Angabe eines Matchings mit 10 Kanten zeigt Optimalität.

Problem: Ungerade Kreise





Shrink it!



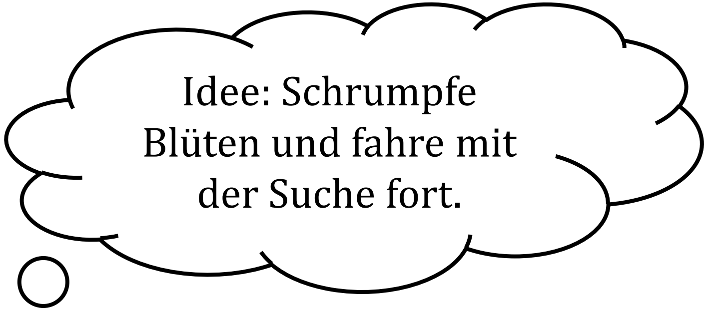
Weg: $(v_1, v_2, v_3, v_7, v_8, v_5, v_4, v_3)$
Blüte: $(v_3, v_7, v_8, v_5, v_4, v_3)$

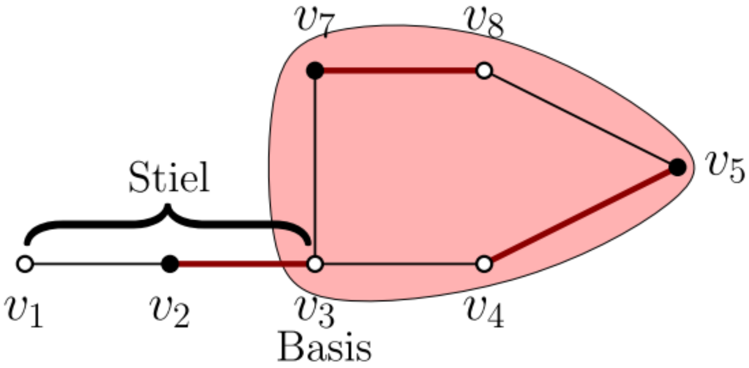
Definition 5.20
Ein M-alternierender Weg $(v_1, \dots, v_\ell)$ heißt M-Blume, wenn

- v_1 nicht von M überdeckt ist.
- $v_1, \dots, v_{\ell-1}$ paarweise verschieden sind.
- $v_i = v_\ell$ für ein ungerades $i \in \{1, \dots, \ell - 1\}$ gilt.

Dabei heißt $(v_1, \dots, v_i)$ M-Stiel, $(v_i, \dots, v_\ell)$ M-Blüte und v_i Basis.

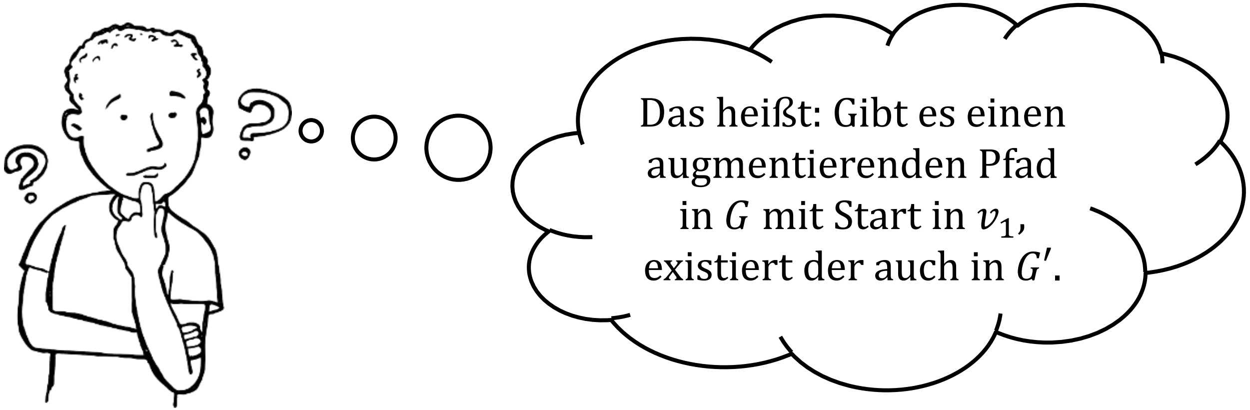
Shrink it!


Idee: Schrumpfe Blüten und fahre mit der Suche fort.


v_7
v_8
v_5
Stiel
v_1
v_2
v_3
v_4
Basis

Shrink it!

Lemma 5.21
Sei $G = (V, E)$ ein Graph mit Matching M. Sei $(v_1, \ldots, v_\ell)$ eine M. Blume mit Blüte B. Sei außerdem G' der Graph und M' das Matching, das durch Schrumpfen von B zu einem Knoten entsteht.
Dann ist M genau dann in G maximal, wenn M' maximal in G' ist.



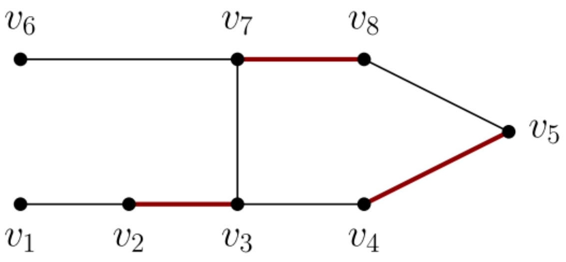
Algorithmus von Edmonds

Algorithmus 5.22
Eingabe:
Bipartiter Graph $G = (V, E)$
Ausgabe:
Maximales Matching M

1. Function $\textsc{Blossom}(G)$
2. Setze $M' = M = \emptyset$ und $G' = G$
3. **for** $r \in V$ mit r ungematcht **do**
4. Setze $T \coloneqq (\{r\}, \emptyset)$, $W(T) \coloneqq \{r\}$ und $S(T) \coloneqq \emptyset$
5. **While** (es ex. Kante $\{v, w\} \in E'$ mit $v \in W(T)$ und $w \notin S(T)$)
6. **If** w ist ungematcht **then**
7. Benutze $\{v, w\}$, um M' zu augmentieren.
8. Erweitere M' zu einem Matching M von G
9. Ersetze M' durch M und G' durch G.
10. Gehe zu Zeile 3.
11. **else if** $w \notin V(T)$, aber w ist in M' gematcht **then**
12. Benutze $\{v, w\}$, um T zu erweitern.
13. **else if** $w \in W(T)$ **then**
14. Benutze $\{v, w\}$ zum Schrumpfen einer Blüte.
15. Aktualisiere M', G$'$ und T entsprechend.
16. **Return** M

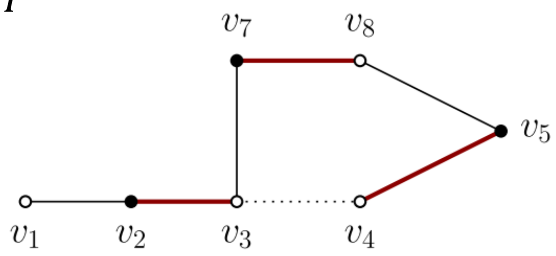
Beispiel

G'

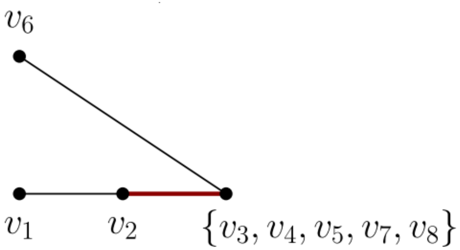


v_6 v_7 v_8
v_5
v_1 v_2 v_3 v_4

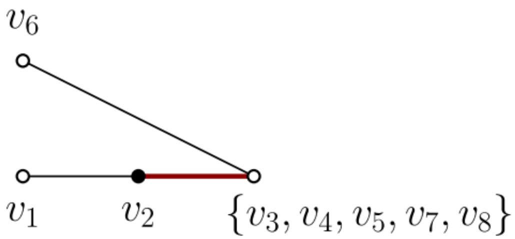
T



v_7 v_8
v_5
v_1 v_2 v_3 v_4



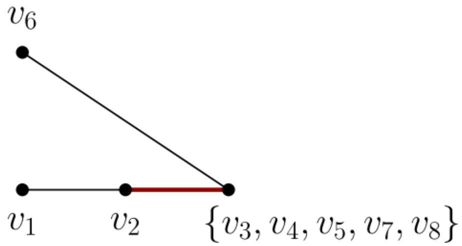
v_6
v_1 v_2 $\{v_3, v_4, v_5, v_7, v_8\}$



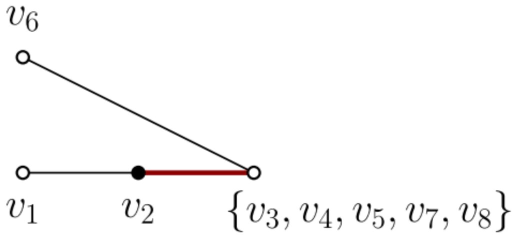
v_6
v_1 v_2 $\{v_3, v_4, v_5, v_7, v_8\}$

Beispiel

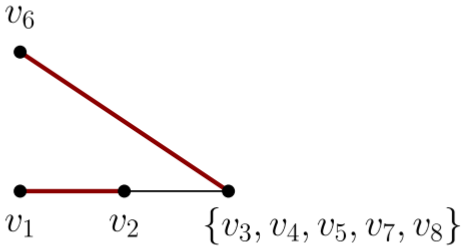
G'



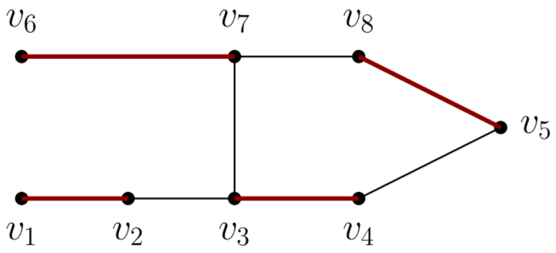
T



Augmentierender Pfad gefunden!



Blüte auflösen



Analyse

Lemma 5.23
Der Algorithmus von Edmonds (Blossom-Algorithmus) löst Problem 5.2 korrekt in Zeit $O(mn^2)$.

Korrektheit: Klar aus Satz 5.16 und Lemma 5.21

Laufzeit:
Maximal n Iterationen der for-Schleife.
Pro Iterationen, maximal $\boxed{n}$ Iterationen der while-Schleife.
Darin:
- Kante finden $O(m)$
- Augmentieren $O(n)$
- T erweitern $O(n)$
- Blüte Schrumpfen $O(m)$

$\Rightarrow$ Laufzeit ist in $O(mn^2)$.

Maximal n Blüten schrumpfen
Maximal n-Mal T erweitern

Geht es noch schneller?

An $O(\sqrt{|V|} \cdot |E|)$ Algorithm for Finding Maximum Matching in General Graphs

*Silvio Micali * and Vijay V. Vazirani ***

University of California - Berkeley

ABSTRACT

In this paper we present an $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding a maximum matching in general graphs. This algorithm works in 'phases'. In each phase a maximal set of disjoint minimum length augmenting paths is found, and the existing matching is increased along these paths.

In this paper, matched edges will be drawn wiggly and unmatched edges will be drawn straight.

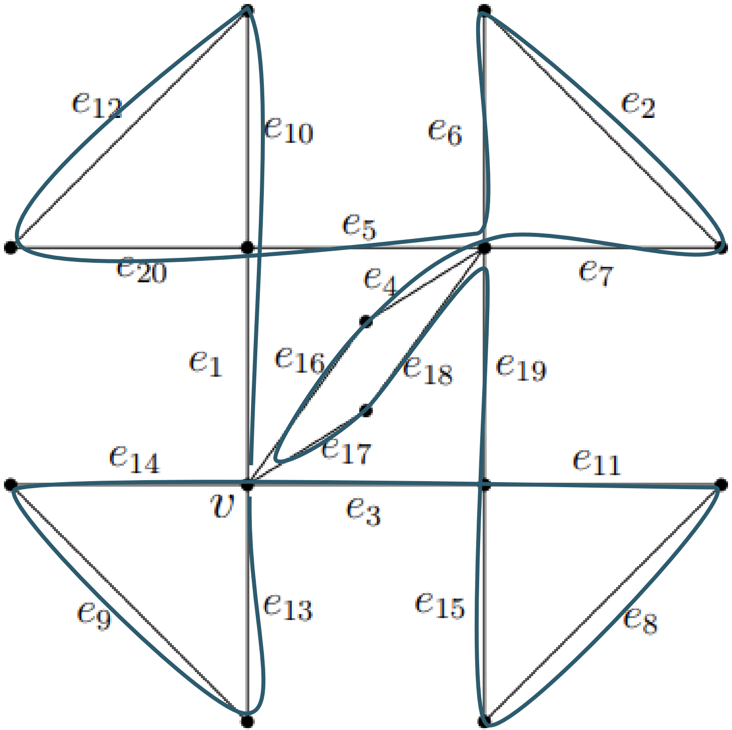
A vertex is 'free' if all edges incident at it are unmatched.

An 'alternating path' is a simple path whose edges are alternately in M and not in M.

An 'augmenting path' is an alternating path between two free vertices.


And now for something completely different.
whendoiturnbackintoapumpkin

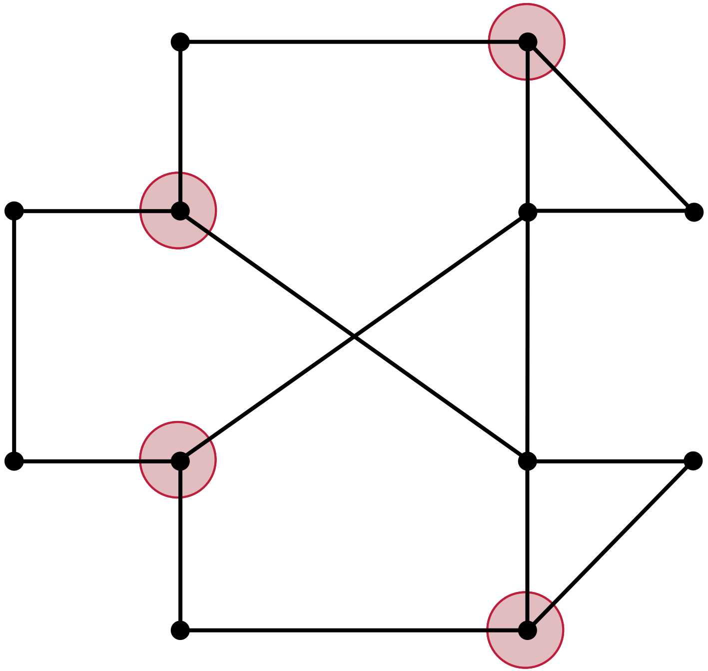
Eulertouren



Aufgabe:
Laufe jede Kante genau 1x ab.

Eigenschaften für Eulertouren:
Zusammenhängend + jeder Knotengrad ist gerade.

Briefträgerproblem



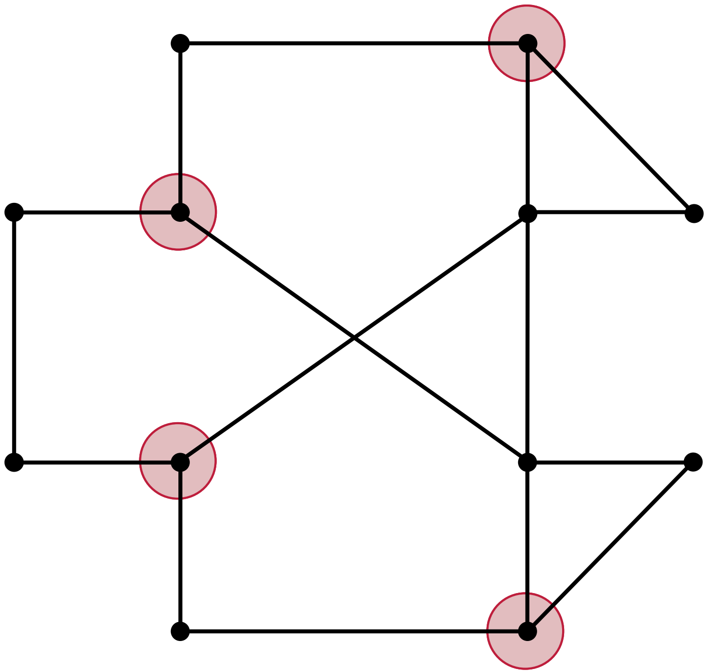
Aufgabe:
Laufe jede Kante mind. 1x ab.

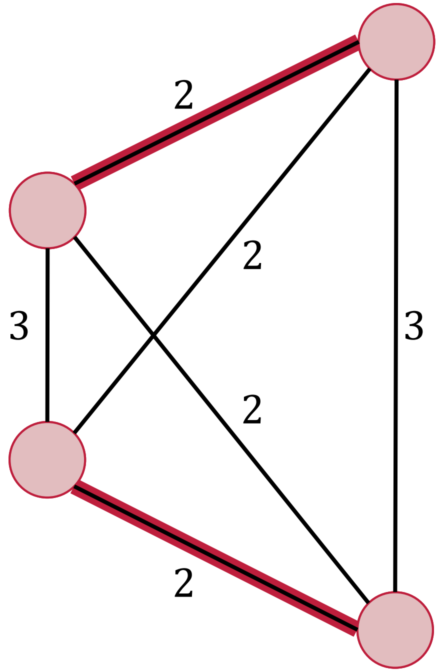
Idee Eulertouren:
Verdoppele Kanten so, dass eine Eulertour entsteht.

Problem:
Briefträger will möglichst wenig Kanten doppelt besuchen.

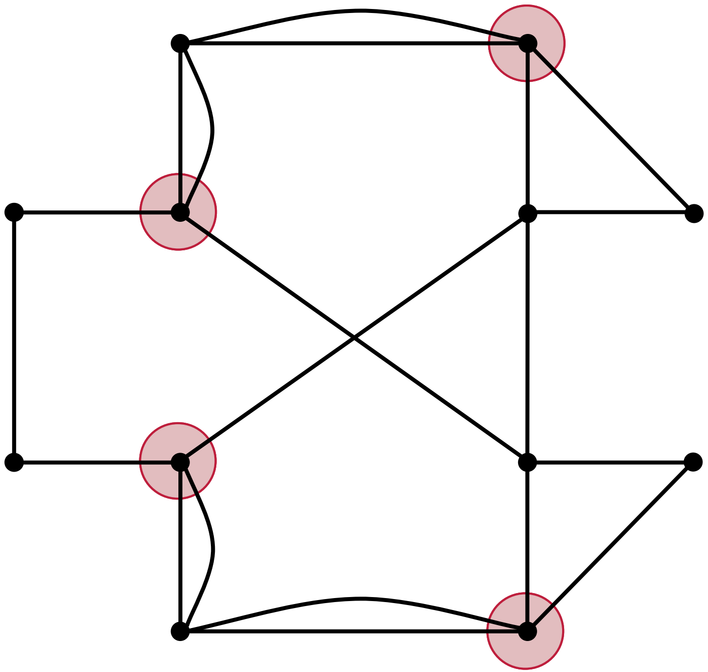
Also:
Ungerade Knoten gerade machen.
⇒ Füge Pfade zwischen diesen Knoten ein!

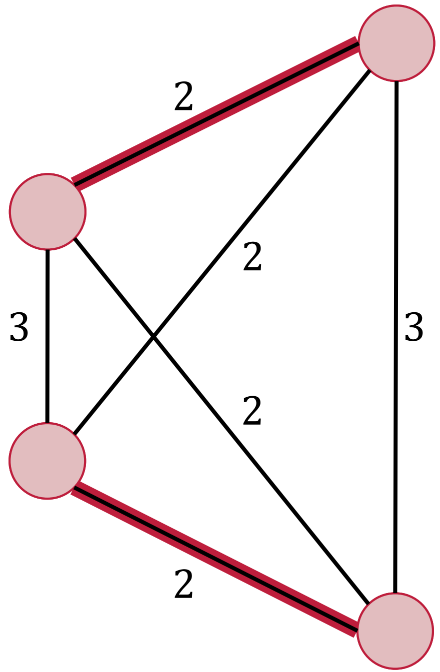
Briefträgerproblem





Briefträgerproblem

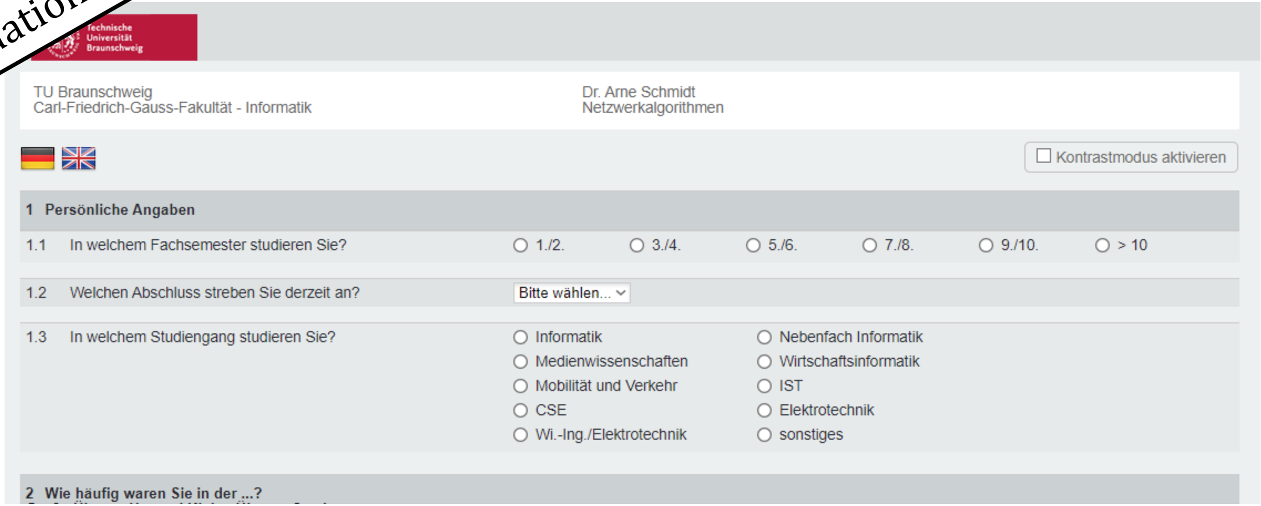




5.3 Gewichtete Matchings

Evaluation!

Nächste Woche!


TU Braunschweig
Carl-Friedrich-Gauss-Fakultät - Informatik

Dr. Arne Schmidt
Netzwerkalgorithmen

☐ Kontrastmodus aktivieren

1 Persönliche Angaben

1.1 In welchem Fachsemester studieren Sie? ○ 1./2. ○ 3./4. ○ 5./6. ○ 7./8. ○ 9./10. ○ > 10

1.2 Welchen Abschluss streben Sie derzeit an? Bitte wählen...

1.3 In welchem Studiengang studieren Sie?
○ Informatik ○ Nebenfach Informatik
○ Medienwissenschaften ○ Wirtschaftsinformatik
○ Mobilität und Verkehr ○ IST
○ CSE ○ Elektrotechnik
○ Wi.-Ing./Elektrotechnik ○ sonstiges

2 Wie häufig waren Sie in der ...?

Start nächste Woche, Einladung per Mail über Mailingliste!

Teaser

Skript

Netzwerkalgorithmen

Arne Schmidt

Institute of Operating Systems and Computer Networks

5. Juli 2021

1 69

Klausur

Klausur

Die **Klausur** findet voraussichtlich am Montag, den **16.08.2021 zwischen 08:00 Uhr und 10:30 Uhr** über EvaExam statt. Genauere Informationen werden noch bekanntgegeben.

Klausur findet online statt!

Zur Vorgehensweise und Inhalt: VL 13!