Technische Universität München
Institut für Informatik
Prof. Dr. Angelika Steger

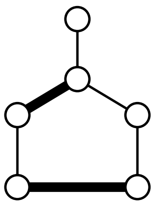
SS 2003
Aufgabenblatt 2
14.04.2003

Praktikum Diskrete Optimierung

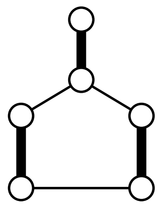
Abgabetermin: Montag, den 28.04.2003, 14.^{00} Uhr

Matchings in Graphen

Es sei ein ungerichteter Graph $G = (V, E)$ gegeben. Ein *Matching* in G ist eine Teilmenge $M \subseteq E$, so daß keine zwei Kanten aus M einen Endpunkt gemeinsam haben. Ein Matching M heißt *maximal*, falls es in G kein größeres Matching M' mit $M' \supset M$ gibt. Ein Matching M heißt *Matching maximaler Kardinalität* (engl. *maximum matching*), falls es in G kein Matching M' mit $|M'| > |M|$ gibt. Wir sind an einem Algorithmus interessiert, der für einen gegebenen Graphen effizient ein Matching maximaler Kardinalität berechnet.



(a) maximales Matching



(b) Matching max. Kardinalität

Matching-Probleme treten in der Praxis meistens als Zuordnungsprobleme auf. Ein Beispiel könnte etwa die Zuordnung von Professoren zu Vorlesungen sein, wenn jeder Professor höchstens eine Vorlesung halten will: die Knoten des Eingabegraphen repräsentieren die Professoren und die Vorlesungen, eine Kante zwischen einem Professor und einer Vorlesung gibt an, daß der Professor die Vorlesung halten kann. Ein Matching maximaler Kardinalität entspricht dann genau einer Zuordnung von Professoren zu Vorlesungen, bei der so viele Vorlesungen angeboten werden können wie möglich.
Für einen Graphen $G = (V, E)$ und ein Matching $M \subseteq E$ nennen wir die Kanten aus M *gematcht* und die Kanten aus $E \setminus M$ *ungematcht*. Ein Knoten heißt *gematcht*, wenn eine seiner inzidenten Kanten gematcht ist. Knoten, die keine inzidente gematchte Kante haben, heißen *frei*. Hilfreich ist weiterhin das Konzept alternierender bzw. augmentierender Pfade bzgl. eines Matchings M:

(i) Ein einfacher Pfad $v_0, v_1, \ldots, v_r$ heißt *alternierend*, falls die Kanten (v_{i-1}, v_i) abwechselnd in M und in $E \setminus M$ sind.

(ii) Ein alternierender Pfad heißt *augmentierend*, falls er an beiden Enden ungematchte Kanten hat und nicht verlängert werden kann, also wenn beide Endknoten des Pfades frei sind.

Beachte, daß ein augmentierender Pfad auch aus einer einzigen ungematchten Kante zwischen zwei freien Knoten bestehen kann.

Man sieht leicht, daß ein Matching M mit Hilfe eines augmentierenden Pfades p zu einem Matching M' vergrößert werden kann: wenn man die gematchten Kanten des Pfades aus M herausnimmt und die ungematchten Kanten des Pfades in M einfügt, erhält man nämlich ein gültiges Matching mit einer zusätzlichen Kante. Diesen Prozeß bezeichnen wir auch als *Invertieren* eines augmentierenden Pfades. Das obige Matching (b) läßt sich zum Beispiel aus Matching (a) durch ein derartiges Invertieren eines augmentierenden Pfades erhalten.

Satz 1 *Ein Matching M in $G = (V, E)$ hat genau dann maximale Kardinalität, wenn es keinen augmentierenden Pfad gibt.*

Beweis: Die Behauptung des Satzes ist äquivalent zu: Ein Matching M hat genau dann nicht maximale Kardinalität, wenn es einen augmentierenden Pfad gibt. Wir beweisen diese Behauptung. Die Richtung $\Leftarrow$ ist offensichtlich richtig. Die Richtung $\Rightarrow$ ist noch zu zeigen.
Sei M ein Matching in G, das nicht maximale Kardinalität hat. Sei M' ein Matching in G mit maximaler Kardinalität. Betrachte den Graphen $G' = (V, M \oplus M')$, wobei $M \oplus M' = (M \cup M') \setminus (M \cap M')$ die symmetrische Differenz von M und M' ist. Offensichtlich haben in G' alle Knoten Grad ≤ 2. Also besteht G' aus einer Reihe von einfachen Pfaden und Kreisen, die alle alternierend bzgl. M sind. M' läßt sich aus M durch Invertieren aller dieser Pfade und Kreise erhalten. Da das Invertieren von Kreisen und von nicht-augmentierenden Pfaden das Matching nicht vergrößert, müssen unter den Pfaden auch $|M'| - |M| \geq 1$ augmentierende Pfade sein.
Aus dem Beweis des Satzes folgt sogar, daß es in G bzgl. M genau $|M'| - |M|$ knotendisjunkte augmentierende Pfade gibt, wobei M' ein Matching maximaler Kardinalität ist. Da sich die $|M|$ gematchten Kanten auf diese $|M'| - |M|$ augmentierenden Pfade verteilen, läßt sich auch ableiten, daß es einen augmentierenden Pfad der Länge höchstens $2 \cdot \lfloor |M| / (|M'| - |M|) \rfloor + 1$ gibt.
Aufgrund des Satzes wissen wir auch, daß wir ein Matching maximaler Kardinalität finden können, indem wir mit einem beliebigen Matching M, zum Beispiel $M = \emptyset$, starten und so lange augmentierende Pfade suchen und diese invertieren, bis es keinen augmentierenden Pfad mehr gibt. Es ist nur noch zu klären, wie die Suche nach augmentierenden Pfaden realisiert werden soll.
Dabei stellt es sich zunächst als sehr günstig heraus, nicht beliebige augmentierende Pfade zu suchen, sondern eine maximale Menge *kürzester* augmentierender Pfade. Sei M das aktuelle Matching und sei ℓ die Länge (Anzahl Kanten) eines kürzesten augmentierenden Pfades bzgl. M. Dann suchen wir eine Menge $p_1, p_2, \ldots, p_r$ von knotendisjunkten augmentierenden Pfaden der Länge ℓ, so daß es keinen weiteren solchen knotendisjunkten Pfad mehr gibt, und invertieren alle diese Pfade $p_1, \ldots, p_r$. Dieses Vorgehen hat den Vorteil, daß wir höchstens $O(\sqrt{|V|})$-mal augmentierende Pfade suchen müssen, wie Hopcroft und Karp gezeigt haben. Läßt sich die Suche nach augmentierenden Pfaden in Zeit $O(|V| + |E|) = O(|E|)$ realisieren, erhält man also einen Matching-Algorithmus mit Gesamtlaufzeit $O(\sqrt{|V|}|E|)$.

Definition 1 *Ein Graph $G(V, E)$ heißt genau dann bipartit, wenn sich die Menge der Knoten V in zwei Partitionen V_1 und V_2 mit $V_1 \cap V_2 = \emptyset$ und*

$V_1 \cup V_2 = V$ aufteilen lässt, so dass $E \subseteq V_1 \times V_2$ gilt, d.h. Kanten verlaufen nur zwischen Knoten in V_1 und V_2.

Aufgabe 3 Matchings in bipartiten Graphen

Implementieren Sie die Suche nach einem kardinalitätsmaximalen Matching in ungerichteten bipartiten ungewichteten Graphen. Verwenden Sie dabei eine Methode, die auf augmentierenden Pfaden basiert und eine Worstcase-Laufzeit von $O(\sqrt{|V|}|E|)$ besitzt (näheres hierzu finden Sie u.a. in der angegebenen Literatur). Ihr C++-Programm soll den Graphen von der *Standardeingabe* lesen und ein beliebiges kardinalitätsmaximales Matching auf der *Standardausgabe* ausgeben.

Wichtiger Hinweis: Halten Sie das im folgenden angegebene Eingabe- und Ausgabeformat bitte genau (keinerlei zusätzliche Leerzeichen, Zeilenumbrüche etc.) ein! Die abgegebenen Programme werden u.a. auch mit Hilfe eines automatisierten Verfahrens auf verschiedenen Eingaben getestet.

Eingabeformat

Als Eingabe erhält Ihr Programm zunächst eine Zeile mit der Knotenanzahl n_1 der ersten Partition getrennt durch ein Whitespace von der Knotenanzahl n_2 der zweiten Partition des Graphen. Alle Knoten in der ersten Partition werden durch die Zahlen zwischen 0 und $n_1 - 1$, die der zweiten Partition durch Zahlen zwischen n_1 und $n_1 + n_2 - 1$ repräsentiert. In den darauffolgenden Zeilen werden die Kanten des Graphen als Paare von Knoten angegeben, wobei jede (ungerichtete) Kante in der Auflistung nur einmal vorkommt und nur Kanten vorhanden sind, die Knoten aus den zwei verschiedenen Partitionen verbinden.

Beispieleingabe

```
6 8
0 6
0 7
0 8
1 9
2 10
3 10
3 11
4 11
4 12
5 12
```

Ausgabeformat

Ihr Programm soll das gefundene kardinalitätsmaximale Matching im Eingabegraphen in einer beliebigen Reihenfolge ausgeben. Für jede Kante im Matching sollen jeweils die inzidenten Knoten durch ein Whitespace getrennt ausgegeben werden. Hierbei soll jede Kante *genau einmal* ausgegeben werden.

Beispielausgabe

```
0 6
1 9
2 10
3 11
4 12
```

Weitere Beispieleingaben und -ausgaben stehen unter
`http://www14.in.tum.de/lehre/2003SS/optprak/data.html`
zur Verfügung.

Zeitlimit und Abgabe

Pro Testeingabe stehen maximal 30 Sek. Laufzeit zur Verfügung.
Die Abgabe der Lösung muß bis Montag, 28.04.2003, 14.^{00} Uhr, per E-Mail an die Adresse `optprak@in.tum.de` erfolgen.

Literaturhinweise

- Turau. Algorithmische Graphentheorie. Addison Wesley, 1996
- Papadimitriou, Steiglitz. Combinatorial Optimization - Algorithms and Complexity. Dover Publications, 1998
- U. Manber. Introduction to Algorithms. Addison-Wesley, Reading MA, 1989

Allgemeine Hinweise

Bitte halten Sie sich an folgende Punkte bei der Abgabe der weiteren Aufgaben. Dadurch wird eine schnellere Korrektur in Zukunft möglich.

- Name des Makefiles soll `Makefile` sein.
- Abgabe der Aufgabe als Attachment per mail in einem `.tar.gz` oder `.tgz` File, das *keine* Unterverzeichnisse enthält.
- Name des Executeables soll `a<Aufgabennummer>` sein.
- Verwenden sie Return-Werte $\neq 0$ in `main` nur für Fehlerfälle.