

Matchings in bipartiten Graphen

Proseminar im SS 2001

Lehrstuhl Prof. Dr. Brandenburg, Universität Passau

Michael Coduro, coduro@fmi.uni-passau.de

Zusammenfassung:

Inhalt dieses Vortrages sind die theoretischen Hintergründe und das Generieren von maximalen Matchings in bipartiten Graphen. Dabei werde ich schrittweise an den ungarischen Algorithmus heranzuführen, der nach maximalen Matchings in bipartiten Graphen sucht. Dieser Algorithmus benutzt die Erkenntnisse aus dem Satz von Berge und dem Heiratssatz von Hall, die ich vorher erläutern und beweisen werde.

Es wird auch noch ein Ausblick zur Bestimmung von optimalen Matchings in bipartiten Graphen gegeben. Dabei ist der Algorithmus von Kuhn-Munkres von zentraler Bedeutung.

Grundlegende Begriffe

Ein **Matching** M (auch Paarung) ist eine Teilmenge der Kanten eines Graphen $G=(V, E)$, wenn M keine Schlingen enthält und keine zwei Kanten aus M inzident sind, d.h. $k \cap l = \emptyset$ für alle $k, l \in M$ mit $k \neq l$. Diese Kanten haben also keine gemeinsamen Endknoten.

Die Knoten, die in einem Matching M nicht mit einer Kante aus M inzident sind, werden **frei** genannt. Ansonsten nennt man sie gemacht oder gesättigt.

Normalerweise will man in einen Graphen das perfekte Matching, welches keine freien Knoten mehr enthält ($|M| = |V|/2$), bestimmen. Lässt sich ein solches nicht ermitteln, z.B. falls der Graph eine ungerade Anzahl an Knoten enthält, will man das maximale Matching bestimmen.

Man nennt ein Matching M **maximal**, falls G kein Matching M' enthält mit $|M| < |M'|$.

Man nennt ein Matching M **gesättigt**, falls man dieses Matching um keine Kante mehr vergrößern kann.

Ein gesättigtes Matching ist aber nicht immer das maximale Matching, wie man sich durch ein Gegenbeispiel veranschaulichen kann.

Bipartite Graphen

Oft ist es in der Anwendung nicht sinnvoll, Kanten zwischen den Knoten der gleichen Gruppe zuzulassen. Solche Graphen werden **bipartit** genannt. Man kann also die Knoten in einem bipartiten Graphen mit zwei Farben so einfärben, dass keine Kanten zwischen den Knoten mit der gleicher Farbe verlaufen.

Mit dieser Klasse von Graphen werden z.B. Lehrer und Schulklassen modelliert. Bestimmt man das maximale Matching in diesen Graphen, hat man einen optimalen Stundenplan gefunden.

Bipartite Graphen besitzen keine Zyklen ungerader Länge. Daher sind Algorithmen zur Konstruktion von maximalen Matchings in bipartiten Graphen auch weniger komplex.

Der Heiratssatz von Hall

Satz: Sei $G=(X \cup Y, E)$ ein bipartiter Graph mit $|X| \leq |Y|$.

G hat Matching M mit $|M| = |X| \Leftrightarrow |N(S)| \geq |S|$ für **alle** $S \subseteq X$,
wobei

$N(S) := \{y \in Y \mid \{x, y\} \in E, x \in S\}$

($N(S)$ ist die Nachbarschaft der Knotenmenge S .)

Dieser Satz gibt also ein Kriterium dafür an, wann ein bipartiter Graph ein Matching besitzt, durch das jeder Knoten in X gesättigt wird.

Im folgenden Beweis bezeichnen wir die Menge X als die Menge der Jungen, und Y als die Menge der Mädchen. Dann ist $N(S)$ die Menge der Mädchen, von denen jede mit mindestens einen Jungen aus S befreundet ist.

Beweis:

Sei $n = |X|$.

„ \Rightarrow “ Für jedes k , $1 \leq k \leq n$ existiert nach Voraussetzung eine Untermenge S von k Jungen, die nur dann alle heiraten können, wenn sie mindestens k Freundinnen haben. Also $|N(S)| \geq k$ und daraus $|N(S)| \geq |S|$.

„ \Leftarrow “ Durch vollständige Induktion nach der Anzahl der Jungen beweise ich, dass jeder Junge mit einer seiner Freundinnen verheiratet werden kann.

Induktionsanfang: $n=1$

Es gibt also nur einen Jungen, den wir x nennen wollen. Dann ist $S=\{x\}$, und aus der Voraussetzung folgt, dass $|N(S)| \geq |S| = 1$. Also besitzt der Junge eine Freundin, mit der wir ihn verheiraten können.

Induktionsschritt: $n-1 \rightarrow n$

Nach Induktionsannahme können alle $n-1$ Jungen eine ihrer Freundinnen heiraten. Wir wollen nun zeigen, dass auch alle n Jungen verheiratet werden können. Dazu betrachten wir zwei Fälle:

1. Fall:

Jede Menge von k ($1 \leq k < n$) Jungen habe zusammen mindestens $k+1$ Freundinnen. Dies bedeutet, dass für jede Teilmenge S von X mit $|S| < n$ gilt, dass $|N(S)| \geq |S| + 1$. Die Voraussetzung ist also erfüllt, wobei ein Mädchen übrigbleibt.

Wir verheiraten nun einen beliebigen Jungen mit einer seiner Freundinnen. Dann bleiben noch $n-1$ Jungen übrig, die verheiratet werden müssen. Es wäre jetzt möglich, dass durch diese Heirat ein weiterer Junge nicht mehr heiraten könnte. Wir können aber auf unsere Induktionsvoraussetzung zurückgreifen, die besagt, dass alle Jungen einer kleineren Teilmenge verheiratet werden können.

2. Fall:

Es gibt eine Teilmenge S mit k Jungen ($k < n$), die gemeinsam genau k Mädchen kennt. Also ist die Voraussetzung erfüllt. Nach Induktionsannahme können alle k Jungen verheiratet werden. Es bleiben also noch $n-k$ Jungen übrig, die noch nicht verheiratet sind. Aber jede Auswahl h von diesen $n-k$ Jungen ($1 \leq h \leq n-k$) muss mindestens h Freundinnen haben, da andernfalls

diese h Jungen zusammen mit der vorigen Auswahl von k Jungen weniger als $h+k$ Freundinnen haben würden.

Also können alle n Jungen verheiratet werden.

Erweiternde Wege

Wie man einfach zeigen kann, führt ein greedy – Algorithmus nicht immer zu einem maximalen Matching. Wir müssen daher anders vorgehen.

Wir zeigen nun ein einfaches Verfahren aus einem nicht maximalen Matching ein größeres zu konstruieren. Dabei wird von einem freien Knoten ein alternierender Weg zu einem anderen freien Knoten gesucht.

Alternierende Wege bestehen aus Kanten, die abwechselnd in den Kanten des Matchings und in den restlichen Kanten des Graphen liegen.

Gibt es einen alternierenden Weg zwischen freien Knoten, so heißt dieser **erweiternder Weg** und das Matching kann um eine Kante vergrößert werden, indem die Kanten die auf dem Weg und im Matching liegen, aus dem Matching entfernt werden, und die anderen Kanten auf diesen Weg in das Matching aufgenommen werden. Dieses Verfahren heißt Umfärben entlang eines erweiternden Weges.

Besitzt der Graph nun weniger als zwei freie Knoten oder kann kein erweiternder Weg mehr ermittelt werden, dann ist dieses Matching maximal. Diesen Zusammenhang drückt der Satz von Berge aus, dessen Beweis folgt.

Um den Satz von Berge zu beweisen, muss ich noch auf die symmetrische Differenz zwischen zwei Matchings eingehen.

Symmetrische Differenz

Seien M und M' zwei Matchings in einem Graphen $G=(V, E)$. Die symmetrische Differenz der beiden Matchings ist

$$M \Delta M' = (M \setminus M') \cup (M' \setminus M)$$

Die symmetrische Differenz gibt also die Kanten an, die in M oder M' , aber nicht in den beiden Matchings liegen.

Bildet man mit der Kantenmenge $M \Delta M'$ einen Untergraphen T , dann lässt sich jede zusammenhängende Komponente von T einem der folgenden zwei Typen zuordnen:

- (1) einen Zyklus gerader Länge, dessen Kanten abwechselnd in M und M' liegen.
- (2) einem Weg, dessen Kanten abwechselnd in M und M' liegen und dessen Endknoten in einem der beiden Matchings frei sind.

Satz von Berge

Satz: Ein Matching M in einem Graphen $G=(V, E)$ ist genau dann maximal, wenn G keinen M -erweiternden Weg enthält.

Beweis:

„ \Rightarrow “ Annahme: Es gibt einen M -erweiternden Weg in G

Dann könnte aber M durch Umfärben entlang dieses erweiternden Weges vergrößert werden. Widerspruch!

„ \Leftarrow “ Sei M' ein weiteres beliebiges maximales Matching von G .

Beh. $|M| = |M'|$, denn:

Betrachte die zusammenhängenden Komponenten des Untergraphen T von G , der durch die Kantenmenge $M \Delta M'$ gegeben ist:

Falls es einen Weg ungerader Länge gibt, wäre dieser entweder M oder M' - erweiternd. Widerspruch!

Also enthält jede zusammenhängende Komponente von T entweder Wege oder Zyklen gerader Länge.

Aber jeder Zyklus oder Weg gerader Länge besitzt dieselbe Anzahl von Kanten aus M wie aus M' .

Daraus: $|M \setminus M'| = |M' \setminus M|$. Es folgt: $|M| = |M'|$

M hat also genauso viele Kanten wie das maximale Matching auf G , daher ist auch M maximales Matching.

Um jetzt einen Algorithmus anzugeben, müssen wir noch wissen, wie man einen M -erweiternden Weg sucht. Wir konstruieren durch Tiefensuche einen Baum H mit der Wurzel x .

Wir erinnern uns an die Eigenschaft bipartiter Graphen, keine Zyklen ungerader Länge zu enthalten. Daher gibt es bezogen auf den Startknoten x nur zwei Arten Knoten: mit geradem oder ungeradem Abstand zu x .

Deshalb hat dieser Baum folgende Eigenschaften:

- x ist ein freier Knoten in G bzgl. des Matchings M .
- Für jeden Knoten v in H ist der einzige Weg von v zur Wurzel x ein M -alternierender Weg.

Deshalb wird dieser Baum auch **M -alternierender Baum** genannt.

Ungarische Methode

Jetzt müssen wir nur noch wissen, wann wir die Suche nach einem M -erweiternden Weg von einem freien Knoten x aus abbrechen können.

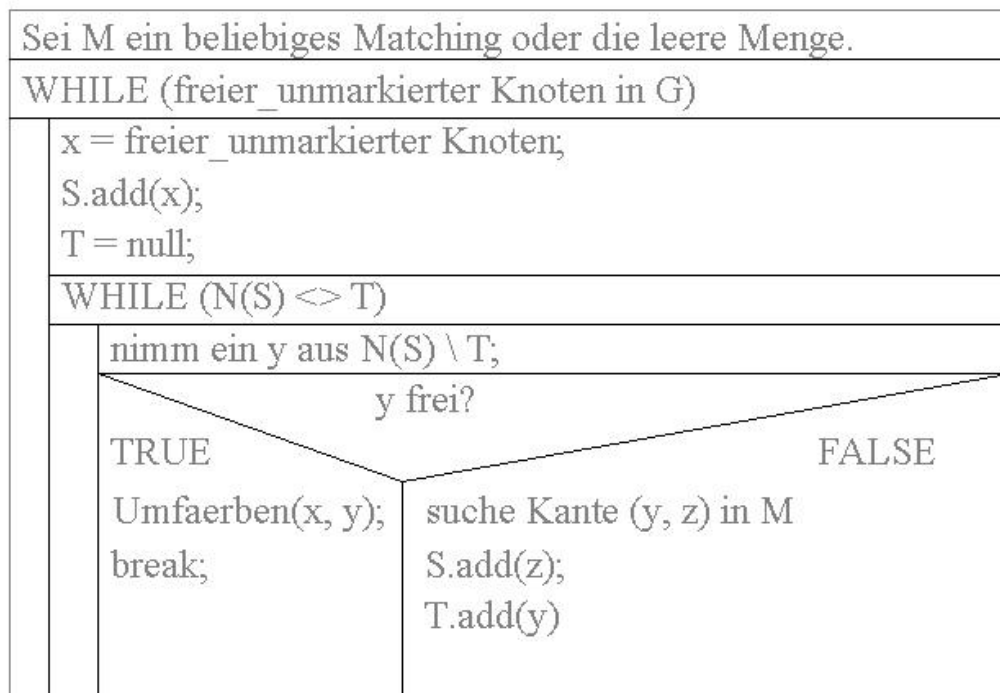
Zu diesem Zweck besitzt die ungarische Methode zwei Mengen S und T .

S enthält alle Knoten die geraden Abstand zum freien Knoten x haben, T enthält alle Knoten mit ungeraden Abstand zu x .

Falls nun $N(S) = T$ können wir die Suche nach einen M-erweiternden Weg abbrechen, da der Heiratssatz von Hall besagt, dass kein Matching existiert, dass alle Knoten von S sättigt.

($|N(S)| = |T| = |S \setminus \{x\}| = |S| - 1$, also $|N(S)| < |S|$)

Falls $N(S) \neq T$ kann die Suche nach einen M-erweiternden Weg fortgesetzt werden. Es ergibt sich nun folgender Algorithmus:



Die innere Schleife terminiert, da nach jedem Durchlauf entweder durch Umfärben zwei frei Knoten weniger im Graphen vorhanden sind, oder die Mengen S und T vergrößert werden(maximal alle Knoten des Graphen $O(|V|)$).

Da nur endlich viele freie Knoten jeweils einmal getestet werden müssen, terminiert die äußere Schleife. Ihre maximale Laufzeit beträgt $O(|V|)$.

Die Komplexität des ungarischen Algorithmus beträgt daher $O(|V|^2)$.

Ford-Fulkerson Methode

Das maximale Matching kann in bipartiten Graphen auch mit Hilfe des Algorithmus von Ford-Fulkerson zur maximalen Flußbestimmung gelöst werden. Zu diesem Zweck wird der ungerichtete bipartite Graph $G=(X \cup Y, E)$ folgendermaßen verändert:

Erweitere G zu $G'=(X \cup Y \cup \{q, s\}, E')$ mit $E' = E \cup \{(q,x) \mid x \in X\} \cup \{(y,s) \mid y \in Y\}$ und ordne jeder Kante die Kapazität 1 zu.

Dadurch, dass wir jeder Kante die Kapazität 1 zuordnen, stellen wir sicher, dass

zu jedem Knoten höchstens eine adjazente Kante im Matching liegt, so dass nach dem Algorithmus ein maximales Matching vorliegt.

Optimales Matching

Bis jetzt haben wir uns nur auf ungewichtete bipartite Graphen beschränkt. In der Praxis werden aber auch Wertungen zwischen den einzelnen Objekten benötigt (z.B. die Effektivität eines Arbeiters, eine bestimmte Aufgabe zu erfüllen). Dies führt uns zu folgender Definition:

Sei $G=(X \cup Y, E)$ ein bewerteter, bipartiter Graph mit $|X| = |Y|$ und einer Gewichtsfunktion $w: E \rightarrow \mathbb{R}_+$. Das maximal-bewertete perfekte Matching M bezeichnet man als **optimales Matching**.

Zur Bestimmung eines optimalen Matchings ordnet man zunächst jedem Knoten eine zulässige Kennzeichnung zu und bestimmt danach den Gleichheitsuntergraphen:

Def. zulässige Knotenkennzeichnung:

Eine Knotenkennzeichnung, die jedem Knoten v in einen bipartiten, bewerteten Graphen $G=(X \cup Y, E)$ eine reelle Zahl $\lambda(v)$ zuordnet, heißt zulässige Knotenkennzeichnung, wenn für alle x_i in X und alle y_k in Y gilt:

$$\lambda(x_i) + \lambda(y_k) \geq w(i,k)$$

Def. Gleichheitsuntergraph

Sei $E_\lambda = \{x_i y_k : \lambda(x_i) + \lambda(y_k) = w(i,k)\}$.

Der Untergraph von G mit der Kantenmenge E_λ heißt der λ entsprechende Gleichheitsuntergraph und wird mit G_λ bezeichnet.

Es ergibt sich nun folgender Zusammenhang zwischen dem optimalen Matching und dem Gleichheitsuntergraphen:

Satz:

Wenn der Gleichheitsuntergraph G_λ ein perfektes Matching M enthält, dann ist M ein optimales Matching von G .

Man wendet nun den ungarischen Algorithmus auf den Gleichheitsuntergraphen an. Findet man kein perfektes Matching in diesen Graphen, dann gibt es Mengen S und T mit $N(S) = T$.

Mit $d_\lambda = \min\{\lambda(x) + \lambda(y) - w(x,y) : x \in S, y \notin T\}$ kann eine neue zulässige Knotenkennzeichnung $\lambda'(v)$ für alle $v \in G$ berechnet werden, wobei

$$\lambda'(v) = \begin{cases} \lambda(v) - d_\lambda & , \text{ falls } v \in S \\ \lambda(v) + d_\lambda & , \text{ falls } v \in T \\ \lambda(v) & , \text{ falls } v \notin S \cup T \end{cases}$$

Mit dieser neuen zulässigen Knotenkennzeichnung wird dann ein neuer Gleichheitsuntergraph $G_{\lambda'}$ erzeugt, der alle Kanten von G_{λ} und noch ein paar zusätzliche enthält, und der ungarische Algorithmus kann mit seiner Suche nach einem perfekten Matching fortfahren.

Zusammengefasst ergibt sich folgender Algorithmus:

1. Starte mit einer zulässigen Knotenkennzeichnung λ für G , bestimme den Gleichheitsuntergraphen G_{λ} , und bestimme ein willkürliches Matching M in G_{λ}

2. Ist M ein perfektes Matching?

ja -> M ist optimales Matching

nein -> Wende ungarische Methode auf G_{λ} an.

Falls $N(S) = T$ berechne Defekt d_{λ} und berechne

dann die neue zulässige Knotenkennzeichnung λ' .

Setze $\lambda = \lambda'$ und $G_{\lambda} = G_{\lambda'}$ und fahre mit der ungarischen Methode fort.

Falls ein M -erweiternder Weg gefunden wird, gehe zu 2.

Literaturliste:

A. Aho; J.E. Hopcroft und J.D. Ullman

Data structures and algorithms. Addison-Wesley Publishing Company, Reading Massachusetts, 1983.

(Für dieses Thema weniger geeignet)

Gary Chartrand, Ortrud R. Oellermann

Applied and algorithmic graph theory. Addison-Wesley, Reading Massachusetts, 1974

(ausführlich, gut)

Claude Berge

Graphs. Elsevier Science Publishers B.V., Amsterdam 1985

(sehr detailliert, aber schwierig)

John Clark, Derek A. Holton

Graphentheorie: Grundlagen und Anwendungen. Spektrum Akad. Verlag Heidelberg, 1994

(sehr ausführlich, gute Beispiele)

T.H. Cormen, C.E. Leiserson, R.L. Rivest

Introduction to Algorithms. The MIT Press, Cambridge Mass. , 1990

(Für dieses Thema weniger geeignet)

Dan Gusfield, Robert W. Irving

The stable marriage problem: structure and algorithm. The MIT Press, Cambridge Massachusetts, 1990

(sehr ausführlich)

T. Ottmann, P. Widmayer

Algorithmen und Datenstrukturen. Spektrum Verlag Heidelberg, 1990

(bedingt geeignet, da hauptsächlich Matchings in allgemeinen Graphen behandelt werden)

R. Sedgewick

Algorithmen. Addison-Wesley, Bonn, München, 1991

(bedingt geeignet, da nur das Heiratsproblem behandelt wird)

U. Schöning

Algorithmen – kurz gefasst. McGraw Hill, New York, 1993

(oberflächliche Darstellung)

L. Volkmann

Fundamente der Graphentheorie. Springer Verlag, 1996

(ausführlich, gut)