Parameterisierte Algorithmen

WS 2011/12 in Trier

Henning Fernau
fernau@uni-trier.de

Parameterisierte Algorithmen Gesamtübersicht

- Einführung
- Grundbegriffe
- Problemkerne
- Suchbäume
- Graphparameter
- Weitere Methoden
- Komplexitätstheorie—parameterisiert

Planare Graphen

stehen im Mittelpunkt der heutigen Vorlesung.

Wir werden Algorithmen vorstellen, die asymptotisch wesentlich besser sind als alle bisherigen *FPT*-Algorithmen (besser als $\mathcal{O}^*(c^k)$).

Wir werden dazu einen Anschluss an die vorige Vorlesung (Baumzerlegungen) suchen.

Dazu erinnern wir zunächst die entsprechenden Begriffe.

Baumzerlegung: die Definition

Eine Baumzerlegung eines Graphen $G = (V, E)$ ist ein Paar $\langle\{X_i \mid i \in I\}, T\rangle$, wobei jedes $X_i \subseteq V$ Bag heißt, und T ist ein Baum mit den Elementen von I als Knoten. Es gelten folgende Bedingungen:

1. $\bigcup_{i \in I} X_i = V$; alle Knoten
2. $\forall \{u, v\} \in E \exists \in I : \{u, v\} \subseteq X_i$; alle Kanten
3. $\forall i, j, k \in I$: liegt j auf dem Weg von i nach k in T, so gilt $X_i \cap X_k \subseteq X_j$. Algorithmik

$\max\{|X_i| \mid i \in I\} - 1$ ist die Weite der Baumzerlegung.
Die Baumweite $\mathrm{tw}(G)$ von G ist das kleinste k, sodass G eine Baumzerlegung der Weite k besitzt.

Etwas Strukturelles

Satz 1 *Besitzt* $G = (V, E)$ *eine Baumzerlegung der Weite* ℓ*, so auch jeder Teilgraph.*

Konsequenz: Beim Nachweis von Baumweitenschranken kann man sich manchmal auf Graphen mit möglichst vielen Kanten zurückziehen, z.B. auf triangulierte zusammenhängende Graphen im Falle planarer Graphen.

Hübsche Baumzerlegungen

Eine Baumzerlegung $\langle\{X_i \mid i \in I\}, T\rangle$ mit ausgezeichnetem Wurzelknoten r heißt hübsche Baumzerlegung falls gilt:

1. Jeder innere Knoten von T hat höchstens 2 Kinder.

2. Hat n zwei Kinder n' and n'', dann ist $X_n = X_{n'} = X_{n''}$ (join node).

3. Hat n nur ein Kind n', gilt entweder

   (a) $|X_n| = |X_{n'}| + 1$ und $X_{n'} \subset X_n$ (insert node) oder

   (b) $|X_n| = |X_{n'}| - 1$ and $X_n \subset X_{n'}$ (forget node).

Satz 2 *Zu jeder Baumzerlegung kann in Linearzeit eine "äquivalente" hübsche Baumzerlegung desselben Graphens gefunden werden.*

Ist der ursprüngliche Baum ein Pfad, so erhalten wir sogar eine "hübsche Pfadzerlegung."

Konsequenz: Beim Entwickeln von Algorithmen für Graphen mit gegebener Baumzerlegung kann man sich auf hübsche Baumzerlegungen beschränken.

Satz 3 *Kleinstmögliche Knotenüberdeckungen können für Graphen mit gegebener Baumzerlegung der Weite ℓ und einem zugehörigen Baum mit n Knoten in Zeit $\mathcal{O}(2^\ell n)$ berechnet werden.*

Algorithm 1 A schematics for algorithms based on a given tree decomposition.

Input(s): a graph G, together with a tree decomposition
Output(s): Whatever to be solved. . .

Find a nice tree decomposition $\mathbb{TD} = \langle \{X_i \mid i \in I\}, T \rangle$ of G.
Perform a depth-first search along T, choosing an arbitrary root r for T as starting point; i.e., call: dfs$(G, \mathbb{TD}, r)$.
Read off a solution from the table that belongs to r.

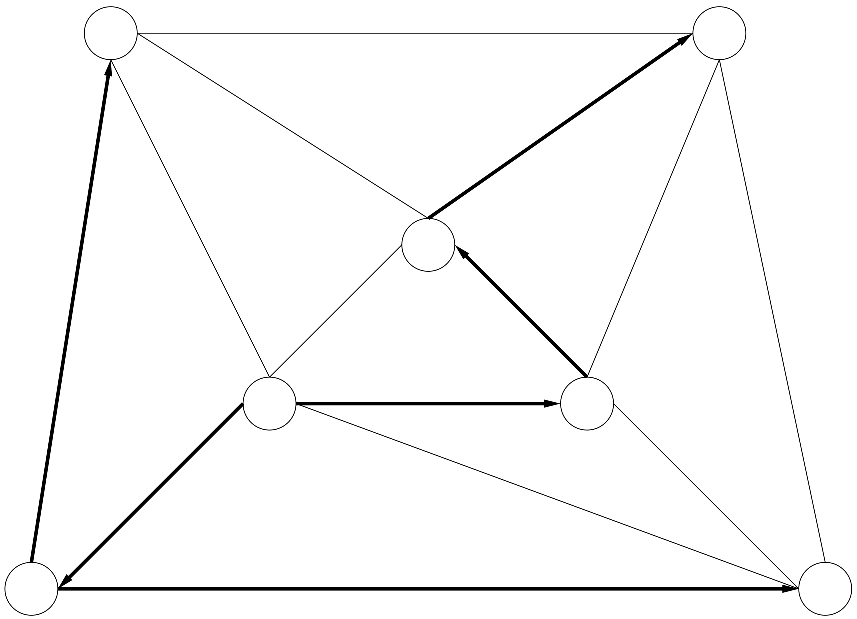
subroutine dfs$(G, \mathbb{TD}, n)$
if current node n is a leaf **then**
    perform leaf node actions and return corresponding table
else if current node n has one child n' **then**
    table$_{n'}$:=dfs$(G, \mathbb{TD}, n')$
    if n is forget node **then**
        perform forget node actions and return corresponding table, based on table$_{n'}$
    else
        perform insert node actions and return corresponding table, based on table$_{n'}$
    end if
else
    {Current node n has two children n' and n''}
    table$_{n'}$:=dfs$(G, \mathbb{TD}, n')$
    table$_{n''}$:=dfs$(G, \mathbb{TD}, n'')$
    perform join node actions and return corresponding table, based on table$_{n'}$ and on table$_{n''}$
end if

Satz 4 *Besitzt ein planarer Graph einen Spannbaum der Höhe ℓ, so ist seine Baumweite durch 3ℓ beschränkt. Eine entsprechende Baumzerlegung kann in Zeit $\mathcal{O}(\ell \cdot n)$ gefunden werden.*

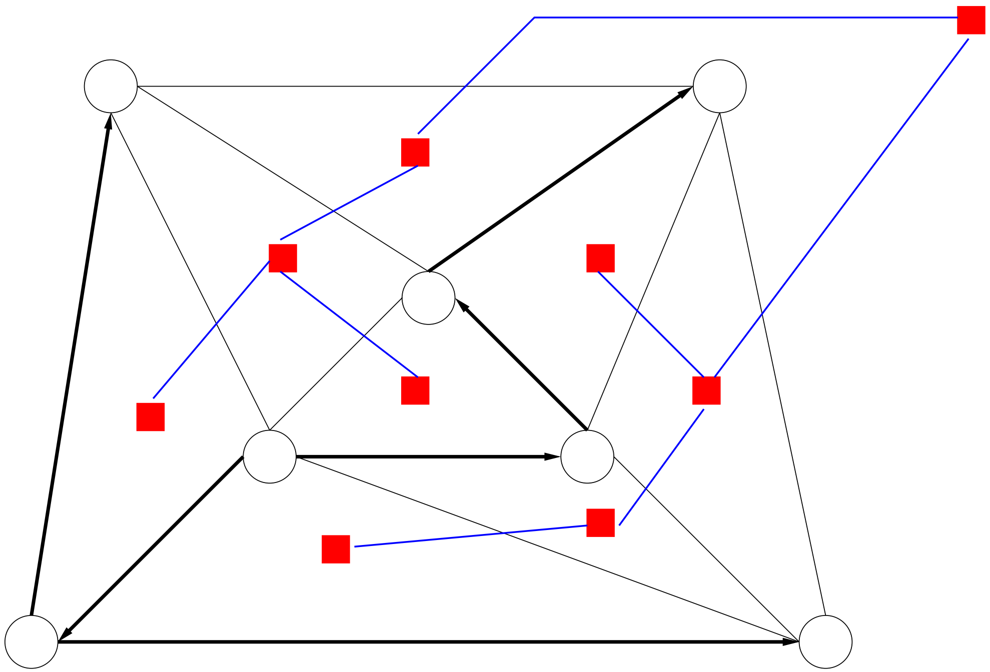
Wir können davon ausgehen, dass der eingebettete Graph $G = (V, E)$ trianguliert ist, d.h., alle Facetten sind Dreiecke.
Betrachte einen Spannbaum B der Höhe ℓ.
Wir definieren einen weiteren Graphen T, den Facettenbaum, wie folgt:
Die Knotenmenge von T sind die Facetten von G.
Zwischen zwei Knoten f, f' von T ist eine Kante gdw. die Berandungen von den Facetten f und f' eine gemeinsame Kante enthalten, die *nicht* in B liegt.

Beweisplan

Achtung: Die äußere Facette ist hier nicht trianguliert.



Ein Spannbaum B



und der sich ergebene Facettenbaum T

Beh. 1: T ist kreisfrei.

Andernfalls gäbe es einen Kreis $f_1, \ldots, f_s, f_{s+1} = f_1$ in T, bezeugt durch Kanten $e_1, \ldots, e_s$, wobei e_i auf dem Rand von f_i und f_{i+1} liegt, aber eben nicht in B.

Wähle von jeder Kante e_i einen Punkt a_i, der auf dem Inneren der e_i darstellenden Kurve liegt.

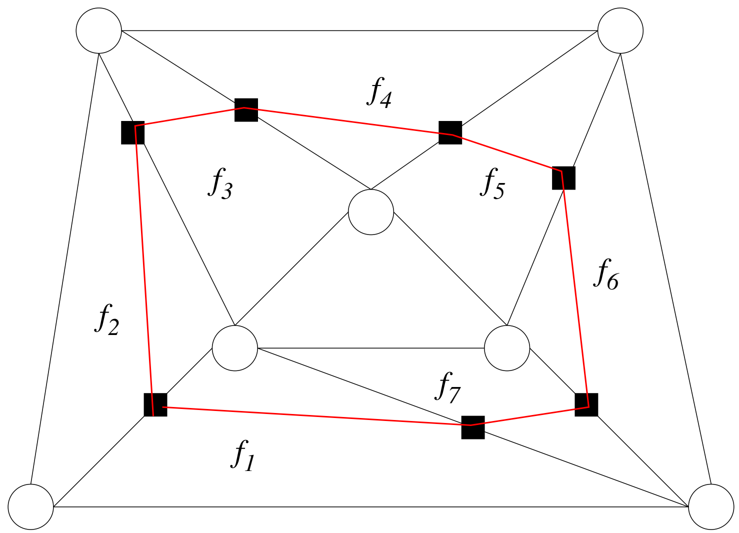
Es sei Γ_i eine Kurve, deren Inneres im Inneren von f_i liegt und die a_i und a_{i+1} verbindet.

Dann ist $\Gamma := \Gamma_1 \cup \cdots \cup \Gamma_s$ ein geschlossenes Polygon in der Ebene.

Jordanscher Kurvensatz $\rightsquigarrow$ Unterteilung der Ebene in zwei Gebiete (und damit nicht-triviale "Aufspaltung" des eingebetteten Graphen).

Da keine Kante von B das Polygon Γ schneidet, kann B kein Spannbaum sein.

Wir sehen später: T ist zusammenhängend, also ein Baum.



Ein Kreis in T mit zugehörigem Polygonzug (rot) gemäß Jordanschem Kurvensatz.

Der (gerichtete) Spannbaum B definiert für jeden Knoten x die Menge $A_B(x)$ der Vorgängerknoten von x in B (einschl. x selbst).
Wir betrachten T als Grundstruktur der anzugebenen Baumzerlegung und ordnen jedem Knoten f von T die Gesamtmenge der Vorgängerknoten $A_B(V(f))$ als Bag zu.
Da B Höhe ℓ hat und G triaguliert, folgt mit der Beobachtung, dass wenigstens die Wurzel allen Vorgängermengen $A_B(v)$ mit $v \in V(f)$ gemein ist, sofort:
Beh. 2: Jeder Bag enthält höchstens $3\ell + 1$ Elemente.
Zu jeder Kante gibt es wenigstens eine Facette und somit gilt:
Beh. 3: Jede Kante des Graphen findet sich in einem Bag wieder.

Mit einigem Aufwand (Induktion über die Höhe ℓ des Spannbaums B) kann man zeigen:
Beh. 4: Für jeden Knoten v ist die Menge der Knoten in T, deren Bags v enthalten, eine zusammenhängende Menge in T.
Da nach Def. die Wurzel von B in jedem Bag von T vorkommt, folgt sofort:
Beh. 5: T ist zusammenhängend.
Daher liefert T die gesuchte Baumzerlegung.

Anwendungen

Wir diskutieren einen kürzesten Weg p maximaler Länge ℓ in einem planaren zusammenhängenden Graphen (ℓ heißt auch Durchmesser).
Seien u, v die Endpunkte von p.
Ausgehend von p kann man auch einen Spannbaum der Höhe ℓ konstruieren.
Ist D dominierende Menge, so dominiert jeder $x \in D$ höchstens drei Knoten auf p, denn sonst gäbe es einen kürzeren Weg von u nach v. Gilt $\ell > 3k$, so haben wir eine NEIN-Instanz; andernfalls eine Baumweite von G von höchstens $9k$.
Jeder zweite Knoten auf p in bildet unabhängige Menge, denn sonst gäbe es einen kürzeren Weg von u nach v. Gilt daher $\ell \geq 2k$, so haben wir JA-Instanz; andernfalls eine Baumweite von G von höchstens $6k$.

Folgerung 5 *Auf planaren Graphen sind die Probleme, dominierende Mengen und unabhängige Mengen aufzufinden, jeweils natürlich parameterisiert, in FPT.*

Separatoren

Eine Knotenmenge S eines Graphen $G = (V, E)$ heißt Separator gdw. es gibt zwei disjunkte, nichtleere Mengen A und B von Knoten, sodass

(1) $A \cup B \cup S = V$,

(2) $A \cap S = \emptyset$, $B \cap S = \emptyset$,

(3) Jeder Weg von A nach B enthält einen Knoten aus S.

$(S; A, B)$ heißt auch Separation von G.

Lemma 6 *Ist S ein Separator, so zerfällt $G \setminus S$ in mindestens zwei Komponenten.*

Separatoren: eine algorithmische Sicht

Betrachte Problem, größtmögliche unabhängige Mengen zu finden.
Gibt es in Graph G einen "kleinen" Separator S, so könnten wir auf ihm alle Möglichkeiten durchprobieren (im Wesentlichen in Zeit $2^{|S|}$).
Haben wir auf diese Weise eine Menge $U \subseteq S$ festgelegt von Knoten, welche in die zu konstruierenden unabhängige Menge hineingehen sollen, so prüfe zunächst Konsistenz innerhalb von $G[S]$. Ist diese gegeben, so löschen wir S und alle Nachbarn von U in G und lösen das Problem (rekursiv) auf den entstehenden (mindestens zwei) Komponenten getrennt.

Was bedeutet dies für die Gesamtlaufzeit so eines Algorithmus ?
Funktioniert "Teile und herrsche" auch rekursiv ?

Separatoren: gleichmäßige Aufteilung

Ein $f(\cdot)$-*Separator-Theorem (mit Konstanten* $\alpha < 1$*,* $\beta > 0$*)* für eine teilgraphen-abgeschlossene Klasse $\mathbb{G}$ von Graphen ist ein Satz der folgenden Form:
Ist G irgendein n-Knoten-Graph aus $\mathbb{G}$, so gibt es eine Separation $(S; A_1, S, A_2)$ von G mit:
(a) $|A_1| \leq \alpha \cdot n$, $|A_2| \leq \alpha \cdot n$,
(b) $|S| \leq \beta \cdot f(n)$

Lipton und Tarjan (1979) bewiesen ein planares $\sqrt{\cdot}$-Separator-Theorem mit Konstanten $\alpha = 2/3$ und $\beta = 2\sqrt{2}$. Djidjev (1982) zeigte für planare Graphen sogar ein $\sqrt{\cdot}$-Separator-Theorem mit $\alpha = 2/3$ und $\beta = \sqrt{6}$ und konnte es (1997) auf $\beta \approx 1.97$ verbessern.

Separatoren: parameterisiert

Erinnerung: IS hat einen $4k$-Kern auf planaren Graphen.

1. Finde kleinen Separator (Größe höchstens $2\beta \cdot \sqrt{k}$).
2. Teste alle Möglichkeiten (rekursiv)
3. Die entstehenden Teilgraphen haben eine Größe von höchstens $(2/3)4k$.
4. In der Rekursion findet man daher Separatoren der Größe höchstens $2\beta\sqrt{(2/3)k}$ usw.

Auf einem Rekursionsast sind daher Separatoren der Größe höchstens $2\beta \cdot (\sum_i (2/3)^{i/2})\sqrt{k} \leq 2\beta\frac{1}{1-\sqrt{2/3}}\sqrt{k}$ zu testen.

Satz 7 *Unabhängige Mengen der Größe mindestens k können auf planaren Graphen in Zeit $\mathcal{O}^*(c^{\sqrt{k}})$ für eine Konstante c gefunden werden.*

Separatoren: woher nehmen ?

Kleine Separatoren in Graphen lassen sich mit Flussalgorithmen finden.

Es gibt aber noch eine andere Idee:

Lemma 8 *Jeder "Bag" einer Baumzerlegung ist ein Separator.*

Versuchen wir also, die baumzerlegungsbasierten Algorithmen mit der Separatoridee zu verknüpfen.

Ein Schichtenmodell

Planare (zshg.) Graphen lassen sich schön im Schichtenmodell veranschaulichen:

$L_0 = \{v\}$ (v sei willkürlich gewählt)

L_i (für $i > 0$) enthalte alle Nachbarn von Knoten der Schicht L_{i-1}, die nicht bereits in einer der Schichten $L_0, \ldots, L_{i-1}$ enthalten sind.

Jede Schicht L_i (außer L_0 und der äußersten) ist ein Separator.

Solch einem Schichtenmodell entspricht in natürlicher Weise ein Spannbaum.

Ein Schichtenmodell für planare zusammenhängende Graphen (Forts.)

Planare Graphen lassen sich schön im Schichtenmodell veranschaulichen:
$L_0, L_1, L_2, \ldots$ seien die “benachbarten Schichten”.
Dazu “gehört” Spannbaum der Tiefe ℓ.
Gilt $\ell \in \mathcal{O}(\sqrt{k})$, so gibt es eine Baumzerlegung der Weite $\mathcal{O}(\sqrt{k})$.
Andernfalls: Gibt es nur $c\dot{k}$ viele Knoten im Graphen, so muss nach dem Schubfachprinzip eine der ersten $\sqrt{k}$ vielen Schichten höchstens $\mathcal{O}(\sqrt{k})$ viele Knoten enthalten (Schichtseparator).
Für die Komponente(n) der ersten $\sqrt{k}$ Schichten gibt es eine Baumzerlegung der Weite $\mathcal{O}(\sqrt{k})$. Zu den jeweiligen “Bags” fügen wir den Schichtseparator hinzu.
Dann gucken wir uns die nächsten $\sqrt{k}$ vielen Schichten an usw.

Ein Schichtenmodell: konkrete Rechnung bei PIS

Machen wir eine genauere Trade-off-Berechnung:
Angenommen, wir lösen eine Graphinstanz “direkt”, sobald wir einen Spannbaum der Höhe höchstens $c \cdot \sqrt{k}$ gefunden haben. Damit haben wir nach obigem Satz eine Baumzerlegung der Weite $3c\sqrt{k}$ und somit können wir solche Instanzen in Zeit $\mathcal{O}^*(2^{3c\sqrt{k}})$ lösen.
Da PIS einen $4k$-Kern hat, betrachten wir die ersten $c' \cdot \sqrt{k}$ Schichten, so enthält wenigstens eine dieser Schichten höchstens $4/c'\sqrt{k}$ viele Knoten.
Dies führt also zu einer Baumzerlegung der Weite $(8/c' + 3c')\sqrt{k}$.
Diese ist kleinstmöglich für $c' = \sqrt{8/3}$;
also erhalten wir eine Baumweite von $6c'\sqrt{k} = 4\sqrt{6k}$. $\leadsto$

Satz 9 *PIS kann in Zeit $\mathcal{O}^*(16^{\sqrt{6k}})$ gelöst werden.*

Ein Schichtenmodell: konkrete Rechnung bei PIS; noch mehr Einzelheiten:
Gibt es Spannbaum der Höhe höchstens $(4/\sqrt{3}) \cdot \sqrt{2k}$, so löse Instanz direkt in Zeit $\mathcal{O}^*(16^{\sqrt{6k}})$.
Andernfalls betrachte die ersten $\sqrt{8/3}\sqrt{k}+1$ vielen Schichten.
Es gibt darunter eine Schicht L (ungleich der allerersten), die höchstens $(4/c')\sqrt{k}$ viele Knoten enthält.
Betrachte Teilgraph, der alle Schichten bis einschließlich L umfasst.
Dieser Teilgraph hat Baumzerlegung der Weite $\sqrt{24k}$.
Füge L zu allen Bags hinzu sowie eine neue Wurzel mit der Menge L als Bag als Teil der Baumzerlegung des Gesamtgraphen.
Im "Restgraphen" betrachten wir wieder die nächsten $\sqrt{8/3}\sqrt{k}$ vielen Schichten.
Es gibt darunter eine Schicht L', die höchstens $(4/c')\sqrt{k}$ viele Knoten enthält.
Diese (zusammen mit L) benutzen wir zum "Auffüllen" der Bags der Baumzerlegung des Graphen, der aus L und L' und den "Zwischenschichten" besteht.
etc.

Weitere Themen

Berechnung von Minimum Dominierenden Mengen mit und ohne Berücksichtigung des linearen Kerns (der Größe $67k$).

t-außerplanare Graphen haben Baumweite $\leq 3t - 1$ (Bakers Ansatz)

Es gibt alternative Methoden zum Baumzerlegungsansatz:
"Verzweigungsweite" scheint bessere Konstanten zu liefern.

Literatur / Nachweise

Schöne Darstellungen über Baumzerlegungen und planare Graphen finden Sie im Buch von Flum und Grohe.

Genaueres zu dem Hauptthema dieser Vorlesung (in anderer Darstellung) finden Sie in:

J. Alber, H.L. Bodlaender, H. Fernau, T. Kloks und R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica, 33 (2002), pp. 461–493.

J. Alber, H. Fernau und R. Niedermeier. Graph separators: a parameterized view. Journal of Computer and System Sciences, 67 (2003), pp. 808–832.

J. Alber, H. Fernau und R. Niedermeier. Parameterized complexity: exponential speedup for planar graph problems. Journal of Algorithms, 52 (2004), 26–56.

F. V. Fomin und D. M. Thilikos: Dominating Sets in Planar Graphs: Branch-Width and Exponential Speed-Up. SIAM J. Comput. 36(2), 281–309 (2006)

Eine gute Zusammenfassung aktueller(er) Techniken findet man in:
F. Dorn, F. V. Fomin und D. M. Thilikos: Subexponential Parameterized Algorithms. ICALP 2007: 15–27.
Z.B. ist dort als beste aktuelle Schranke zum Lösen des Knotenüberdeckungsproblems auf planaren Graphen $\mathcal{O}^*(2^{3.57\sqrt{k}})$ angegeben.