Parametrisierte Komplexität –

eine neue Herangehensweise

für schwere Probleme

Alexander Wolff

Überblick

- Neue parametrisierte Algorithmen für Vertex Cover.
- Parametrisierte Komplexität.
- Vergleich mit bisherigen Ansätzen für NP-schwere Probleme.

Vertex Cover — ein Klassiker

Definition. *Gegeben Graph $G = (V, E)$ mit n Knoten und m Kanten.*

$C \subseteq V$ *heißt* Vertex Cover *(VC) von G, falls* $C \cap \{u, v\} \neq \emptyset \quad \forall\, uv \in E$.

k-Vertex Cover

Gegeben:	Graph $G = (V, E)$.
Parameter:	k
Gesucht:	Vertex Cover der Größe $\leq k$ (k-VC).

- Eines der 1. Probleme, dessen NP-Vollständigkeit gezeigt wurde [K 72].
- Eines der „6 basic NP-complete problems“ [GJ 79].
- Es gibt Approximationsalgorithmen, aber...
- Hat wichtige Anwendungen, etwa in der algorithmischen Biochemie.

Komplexitätsstatus von Min-VC

Approximierbarkeit

- Faktor 2 [G 74]
- Faktor $2 - \frac{\log\log n}{2\log n}$ [BE 85, MS 85]

- Faktor $\geq 7/6$ [H 97]
- Faktor $\geq 16/15$ [BGS 95]
- kein PTAS [ALMSS 92]

Nicht-Approximierbarkeit

Algorithmen für k-VC

		Laufzeit $O(\cdot)$	klam value
Fellows & Langston	86	$f(k)\,n^3$	
Johnson	87	$f(k)\,n^2$	
Fellows	88	$2^k\ n$	
Papadimitriou & Yannakakis	93	$m + 3^k\,n$	
Buss	89	$kn + 2^k\,k^{2k+2}$	8
Balasubramanian, Fellows, Raman	92	$kn + 2^k\,k^2$	54
Balasubramanian, Fellows, Raman	98	$kn + 1.325^k\,k^2$	129
Downey, Fellows, Stege	99	$kn + 1.320^k\,k^2$	131
Niedermeier & Rossmanith	99	$kn + 1.292^k\,k^2$	141

k-VC leichter gemacht

Beob. 1 Sei $v \in V$.
Ein VC muß entweder v **oder** v's Nachbarschaft $N(v)$ enthalten.

Beob. 2 Gegeben (G, k).
Jeder Knoten mit Grad $> k$ ist in **jedem** k-VC von G enthalten.

Beob. 3 Falls $\Delta(G) \leq k$ und $|E| > k^2$, so hat G **kein** k-VC.

Der Sam-Buss-Algorithmus

Phase I: Kernbildung

Sei $H = \{v \in V : \deg(v) > k\}$.
Falls $|H| > k$, gib NEIN aus. Fertig.
Sei $G'(V', E') = G|_{V-H}$ und $k' = k - |H|$.
Falls $|E'| > k^2$, gib NEIN aus. Fertig.

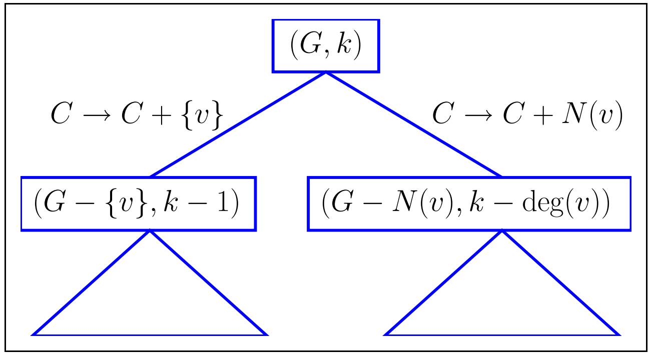
Phase II: Brute Force — Hat G' VC der Größe k'?

Gehe durch alle k'-elementigen Teilmengen C von V'.
Falls C VC von G', gib $H \cup C$ aus. Fertig.
Gib NEIN aus.

Der Suchbaum-Algorithmus

Ziel: Verbessere Phase II.

Idee: Baue rekursiv einen Suchbaum für ein k-VC C.



Verbesserung des Suchbaums

Was wäre, wenn immer $\deg(v) \geq 4$?

Rekursionsgleichung:	$T(k) = T(k-4) + T(k-1) + 1, \qquad T(\leq 0) = 0$
Verzweigungsvektor:	$(4, 1)$
Charakterist. Polynom:	$z^4 = z^{4-4} + z^{4-1} = 1 + z^3$
Verzweigungszahl:	$z \approx 1.38$

Größe des Suchbaums: $T(k) \in O(1.38^k)$

Größenordnung der Verbesserung

	Verzweigungsvektor	Rekursionsgleichung	$O(\cdot)$
vorher	$(1, 1)$	$S(k) = S(k-1) + S(k-1) + 1$	2^k
jetzt	$(4, 1)$	$T(k) = T(k-4) + T(k-1) + 1$	1.38^k

	k	1	4	10	20	50
vorher	$S(k)$	1	15	1023	1.048.575	$\approx 10^{15}$
jetzt	$T(k)$	1	4	35	906	$\approx 10^7$

Bisherige Kernbildung

Eliminiere Knoten mit Grad $> k$.

Regel K.

Eliminiere Knoten mit Grad 0.

Regel 0.

Verbesserte Kernbildung I

Eliminiere Knoten mit Grad 1.

Regel 1.

Eliminiere Knoten mit Grad 2.

Regel 2.

Verbesserte Kernbildung II

Eliminiere Knoten v mit $N(v) \cup \{v\} \supseteq N(u)$ für ein $u \in N(v)$

Regel E.

Eliminiere Knoten mit Grad 3.

Regel 3.1

Der Grad-4-Algorithmus

Wende die verbesserte Kernbildung in jedem Suchbaumknoten auf G an.

$\Rightarrow$ Bei jeder Verzweigung hat G einen Knoten mit Grad ≥ 4.

$\Rightarrow$ Verzweige mit $(4, 1) \,/\, 1.38$

$\Rightarrow$ Größe des Suchbaums $O(1.38^k)$

$\Rightarrow$ Laufzeit $O(nk + 1.38^k\, k^2)$

Aufbau der neuen k-VC-Algorithmen

Phase I. *Kernbildung*: Reduktion auf den harten Kern einer Instanz.

Phase II. *Suchbaum*: in der verkleinerten Instanz eine Lösung finden.

alternativ:

Phase II'. *Heuristik:* schnell durch einen Teil des Suchbaums gehen.

Suchbaum weiter verbessern

Ziel: besser als mit $(4,1)\,/\,1.38$ verzweigen!

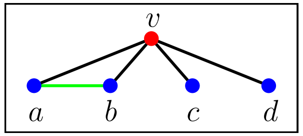
Beob. 4: Sei $N(v) = \{a, b, c, d\}$ in G.

Dann hat G ein minimales VC C mit $|C \cap N(v)| \neq 3$.

Fallunterscheidung

nach Anzahl der Kanten zwischen a, b, c, d.

Fall 1: $G|_{N(v)}$ hat ≥ 1 Kante, sagen wir ab:



Dann hat G ein minimales VC C mit

$\{a,b,c,d\} \subseteq C$ **oder** $N(c) \subseteq C$ **oder** $\{c\} \cup N(d) \subseteq C$.

$\Rightarrow$ verzweige mit $(4,4,5)\,/\,1.29$

Fall 0: $G|_{N(v)}$ hat keine Kante.

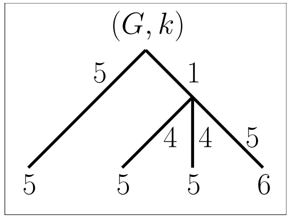
$\Rightarrow$ 3 Unterfälle...

Verzweigungen fest im Blick

	Verzweigungsvektor	Verzweigungszahl
1. Suchbaum-Algo	(1,1)	2
Grad-4-Algo	(1,4)	1.38
Fall 1	(4,4,5)	1.29
Fall 0.1	(4,2)	1.27
Fall 0.2	(4,4,5,9)	1.317
Fall 0.3	(4,7)	1.14
aber	(5,1)	1.324 [BFR 98]
während	(6,1)	1.285

Der Downey-Fellows-Stege-Algorithmus

- Kernbildung vor jeder Verzweigung $\Rightarrow$ G hat Minimalgrad 4.
- Solange es Knoten mit Grad ≥ 6 gibt, verzweige mit $(6,1)\,/\,1.285$.
- Behandle alle Knoten mit Grad 4; schlechtester Fall: $(4,4,5,9)\,/\,1.317$.
- Falls G 5-regulär, entferne 1 Grad-5-Knoten, dann sofort 1 mit Grad 4:



$$(5,1) \oplus \boxed{\begin{array}{l}(4,4,5)\\(4,2)\\(4,4,5,9)\\(4,7)\end{array}} = \boxed{\begin{array}{l}(5,5,5,6)\\(5,5,3)\\(5,5,6,10)\\—\end{array}} \,/\, \boxed{\begin{array}{l}1.304\\1.298\\1.320\\—\end{array}}$$

$\Rightarrow$ Größe des Suchbaums $O(1.320^k)$ $\Rightarrow$ Laufzeit $O(nk + 1.320^k\,k^2)$

Parametrisierte Komplexität

Definition.
Ein parametrisiertes Problem $L \subseteq \Sigma^* \times \mathbb{N}$ ist fest-Parameter-berechenbar*, wenn es einen Algorithmus gibt, der für jede Instanz $(x, k) \in \Sigma \times \mathbb{N}$ in Zeit $f(k)|x|^c$ entscheidet, ob $(x, k) \in L$, wobei*

- *$f : \mathbb{N} \to \mathbb{N}$ eine beliebige Funktion und*
- *c eine von k unabhängige Konstante ist.*

$\mathcal{FPT}$ ist die Menge der fest-Parameter-berechenbaren Probleme.

Anwendungen

- VLSI-Layout
- Biochemie
- Numerische Analysis
- Compilerbau
- Hardware-Beschränkungen
- Linguistik

Methoden

- Suchbäume beschränkter Größe.
- Reduktion auf den Problemkern.
- Quasi-Wohlordnungen.
- Beschränkte Baum- und Pfadweite.
- Farbkodierung / Hashing.

Ansätze für NP-schwere Probleme

- Approximative Lösungen
- Average-Case- statt Worst-Case-Analyse
- Randomisierung
- Quantencomputer
- Empirische Untersuchung von Heuristiken auf Benchmarks
- Entwurf von parametrisierten Algorithmen

Zusammenfassung

- Parametrisierte Komplexität $=$ neuer Werkzeugkasten.
- k-VC kann in $O(nk + 1.3^k\, k^2)$ Zeit gelöst werden.
- Es ist immer sinnvoll, beschränkte Parameter zu identifizieren.
  FPT nutzt sie!
- Hoffnung: „natürliches Problem“ $P \in \mathcal{FPT} \;\Rightarrow\; f(k)$ erträglich.