Exakte Algorithmen für NP-harte Probleme auf Netzwerken: Design, Analyse und Implementierung

Jochen Alber*
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen

Abstract: Wir befassen uns in der Arbeit mit dem Entwurf von exakten Algorithmen für verschiedene NP-vollständige Optimierungsprobleme auf Graphen, wie beispielsweise Vertex Cover, Independent Set oder Dominating Set. Im Vordergrund der Arbeit stehen exakte Lösungsverfahren mit beweisbaren Laufzeitschranken. Wir verfolgen dabei den jüngst vorgeschlagenen Ansatz sogenannter "parametrisierter Algorithmen".

Dabei untersuchen wir sowohl von theoretischer, als auch von praktischer Seite unterschiedliche Methoden des Algorithmen-Designs: Datenreduktion, beschränkte Suchbäume, Separation von Graphen und das Konzept von Baumzerlegungen. Schließlich stellen wir ein Software-Paket vor, welches im Rahmen dieses Projektes entwickelt wurde und eine Vielzahl der entwickelten Algorithmen implementiert.

1 Einleitung

Ziel der Dissertation [Alb03] ist der Entwurf von effizienten, exakten Algorithmen für NP-vollständige Optimierungsprobleme auf Netzwerken, d.h. Graphprobleme wie etwa VERTEX COVER, INDEPENDENT SET oder DOMINATING SET. Viele praxisbezogene Aufgaben, beispielsweise sogenannte "facility location"-Probleme aus dem Bereich der Entscheidungsanalyse (decision analysis), sind durch entsprechende Netzwerk-Modellierung auf derartige Fragestellungen zurückzuführen. Im Vordergrund der Arbeit stehen Lösungsverfahren mit beweisbaren Laufzeitschranken. Wegen der solchen Problemen inhärenten kombinatorischen Komplexität müssen wir exponentielles Laufzeitverhalten unserer Algorithmen in Kauf nehmen, wollen dieses jedoch kleinstmöglich halten.

Wir verfolgen dabei den Ansatz sogenannter "parametrisierter Algorithmen" [AGN01, DF99]. Vereinfacht gesagt handelt es sich hierbei um eine zweidimensionale Herangehensweise, bei welcher die Laufzeit nicht ausschließlich in der Größe der Eingabeinstanz, sondern überdies auch in der Größe eines sogenannten "Problemparameters" gemessen wird. Problemparameter ist in unserem Zusammenhang die Größe der zu optimierenden Lösung, wie etwa die Größe einer Knotenüberdeckungsmenge bei VERTEX COVER, die Größe einer unabhängigen Knotenmenge in INDEPENDENT SET oder die Größe einer dominierenden Knotenmenge im Fall von DOMINATING SET. Ein parametrisiertes

*Unterstützt durch die Deutsche Forschungsgemeinschaft (DFG), Forschungsprojekt PEAL (Parameterized complexity and Exact ALgorithms), NI 369/1-1,1-2.

Problem[1] heißt *"fixed-parameter tractable"*, falls sich ein Lösungsalgorithmus der Laufzeit $f(k) \cdot n^{O(1)}$ finden lässt, wobei f eine beliebige, den exponentiellen Laufzeitanteil erfassende Funktion sein darf. Hier ist n die Größe der Eingabeinstanz und k der zugehörige Problemparameter. Entsprechend heißt ein solcher Algorithmus dann Fest-Parameter-Algorithmus. FPT bezeichnet die Klasse aller parametrisierten Probleme, die fixed-parameter tractable sind.

Es ist beispielsweise bekannt, dass par-VERTEX COVER in FPT liegt. Das Problem par-INDEPENDENT SET hingegen ist vollständig für W[1], eine Klasse parametrisierter Probleme, die sich wohl nicht mit einem Fest-Parameter-Algorithmus lösen lassen; par-DOMINATING SET ist sogar W[2]-vollständig, in diesem Sinn also parametrisiert noch weniger handhabbar. Schränkt man sich auf planare Graphen ein, so behalten die genannten Probleme zwar ihre NP-Härte, sind jedoch alle in FPT (siehe [DF99] für Details).

Der Entwurf von effizienten Fest-Parameter-Algorithmen für parametrisierte Optimierungsprobleme auf *planaren* Graphen steht im Mittelpunkt der Arbeit. Unter effizient verstehen wir hierbei Algorithmen mit möglichst geringem exponentiellen Laufzeitanteil (gemessen durch die Funktion f in der Definition von FPT). Dabei untersuchen wir sowohl von theoretischer, als auch von praktischer Seite unterschiedliche Methoden des Algorithmen-Designs: Datenreduktion, beschränkte Suchbäume, Separation von Graphen und das Konzept von Baumzerlegungen. Jedem der Verfahren ist ein Kapitel der Arbeit gewidmet.

2 Ergebnisse

In diesem überblicksartigen Artikel werden wir ausgewählte Verfahren und einige Hauptresultate kurz skizzieren. Wohingegen in der Dissertation zahlreiche NP-harte Probleme auf Graphen betrachtet werden, beschränken wir uns in der Ausführung hier in erster Linie auf das par-DOMINATING SET Problem: Zu gegebenem Graph G und Parameter k ist (ggf. konstruktiv) zu entscheiden, ob es in G eine Knotenmenge D (eine sog. "Dominating Set") der Größe maximal k gibt, so dass jeder Knoten in G entweder selbst zu D gehört oder durch mindestens einen Knoten in D dominiert wird, d.h. mindestens einen Nachbar in D besitzt. DOMINATING SET gilt als zentrales Problem der kombinatorischen Optimierung. Es wurden (Stand 1998) in der Literatur über 1200 Publikationen zu DOMINATING SET, bzw. zu einer der 75 bekannten Variationen des Problems, gezählt [HHS98]. In der Praxis finden sich hierzu zahlreiche Einsatzgebiete, welche von "facility location"-Problemen, über Anwendungen in der Algorithmischen Biologie, der Spieltheorie, bis hin zur Analyse sozialer Netzwerke reichen. Aus theoretischer Sicht erweist sich DOMINATING SET als äußerst robust gegenüber verschiedensten Ansätzen wie etwa der Approximationstheorie. Beispielsweise wurde gezeigt, dass das Problem nicht approximierbar mit Faktor $(1-\epsilon)\ln|V|$ für irgendein $\epsilon > 0$ ist, es sei denn NP $\subseteq$ DTIME$(n^{\log\log n})$ [Fei98].

Am Rande berichten wir über Ergebnisse zu par-VERTEX COVER (hier wird eine Knotenmenge der Größe maximal k gesucht, welche alle Kanten des Graphs abdeckt) und zu

[1] Die "kanonische" parametrisierte Variante eines Optimierungsproblems $\mathcal{Q}$, bei welcher der Problemparameter eine Schranke auf die Optimierungskosten ist, bezeichnen wir mit par-$\mathcal{Q}$.

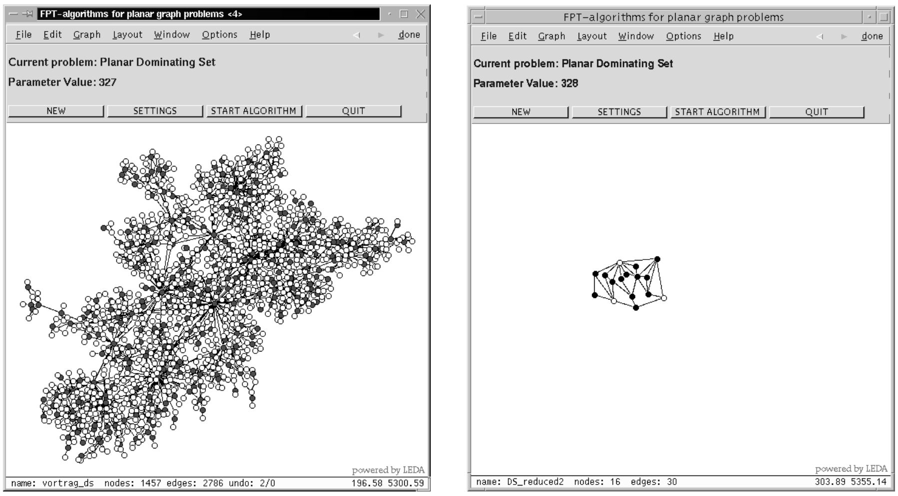

Abbildung 1: Beispiel für die Stärke der Datenreduktion durch Preprocessing im Fall von par-DOMINATING SET. Der planare Graph (mit ca. 1500 Knoten) im linken Bild kann mittels der vorgestellten Polynomzeit- Reduktionsregeln auf einen Graph mit 16 Knoten (siehe rechte Abbildung) reduziert werden. Die eingefärbte optimale Dominating Set im Ausgangsgraph wurde über die Problemkernreduktion in wenigen Sekunden gefunden.

par-INDEPENDENT SET (hier sucht man nach mindestens k unabhängigen Knoten, d.h. solchen Knoten, die paarweise durch keine Kanten verbunden sind).

2.1 Kleine Problemkerne

Datenreduktion gilt als eine zentrale Technik beim Design parametrisierter Algorithmen. Dabei wird eine Eingabeinstanz in deterministisch polynomieller Zeit in eine — bezüglich der Problemstellung äquivalente — neue Instanz kleinerer Größe, den sogenannten Problemkern, transformiert. In der Theorie der parametrisierten Algorithmen hat die Problemkernreduktion eine zentrale Bedeutung aufgrund des folgenden Zusammenhangs: Es lässt sich zeigen, dass ein parametrisiertes Problem genau dann in FPT liegt, wenn es eine Reduktion auf einen Problemkern gibt, dessen Größe nur vom Problemparameter — und nicht von der Größe der Ausgangsinstanz — abhängt. Aus praktischer Sicht sind kleine Problemkerne besonders interessant. Ein bekanntes Theorem von Nemhauser und Trotter liefert etwa einen sogenannten linearen Problemkern für par-VERTEX COVER. Außerdem zeigen wir, wie aus dem Vierfärbbarkeitssatz auf einfache Weise ein linearer Problemkern für par-INDEPENDENT SET auf planaren Graphen folgt. Hauptresultat der Dissertation [Alb03, AFNar] ist in diesem Zusammenhang der Nachweis eines linearen Problemkerns für par-DOMINATING SET auf planaren Graphen, womit wir eine seit längerem offene Frage positiv beantworten können.

Theorem 1 (Problem Kernel Reduction): *Es gibt eine Polynomzeit-Transformation, die einen planaren Graph G in einen planaren Graph G' überführt, so dass Folgendes gilt:*

1) *G hat eine Dominating Set der Größe $k \Leftrightarrow G'$ hat eine Dominating Set der Größe k.*
2) *Die Größe von G' ist durch $c \cdot k$ beschränkt, wobei c eine Konstante ist.*

Die Problemkernreduktion hierfür basiert auf zwei sehr einfachen und leicht zu implementierenden Reduktionsregeln; der Beweis für die Linearität des Problemkerns ist jedoch recht aufwändig. Durch Anwendung der Reduktionsregeln werden bereits Knoten für eine optimale Dominating Set ermittelt.

Abbildung 1 zeigt ein Beispiel für die Problemkernreduktion. Der links dargestellte Graph wurde vermöge der Reduktionsregeln auf den Problemkern reduziert. Auf der verbleibenden kleinen Instanz wird auf naive Weise eine Dominating Set bestimmt. Zusammen mit den während der Reduktion ermittelten Knoten ergibt sich eine optimale Dominating Set im Ausgangsgraph.

Die Implementierung der Reduktionsregeln und eine Reihe von empirischen Studien zeigen die eindrucksvolle Stärke der Problemkernreduktion. Nachfolgende Tabelle gibt Aufschluss über Reduktionsgeschwindigkeit und die Prozentsätze der Reduktionsleistungen von kombinatorisch generierten Zufallsgraphen unter Einsatz jeweils nur einer der beiden Reduktionsregeln, der Kombination von beiden Reduktionsregeln und der Hinzunahme von weiteren, einfachen heuristischen Regeln.

	Rule 1	Rule 2	Rules 1+2	+Heuristics
edges/vertices removed	78.7%	81.1%	85.4%	99.8%
dominating set vertices found	81.9%	75.4%	89.1%	99.7%
time (in sec.)	$0.2 - 14.7$	$0.3 - 35.9$	$0.6 - 33.2$	$0.3 - 6.9$

2.2 Suchbaumalgorithmen

Die wohl am häufigsten angewandte Methode im Entwurf von Fest-Parameter-Algorithmen ist der Einsatz *beschränkter Suchbäume*, mit deren Hilfe sich eine systematische, erschöpfende Suche realisieren lässt. In der Literatur findet man etwa zum par-VERTEX COVER Problem eine Reihe von Arbeiten mit zunehmender Verbesserung der Suchbaumgröße.

Wir betrachten einen speziellen Ansatz für kombinatorische Graphprobleme, die sogenannte "degree-branching"-Technik, bei welcher ein Suchbaum sukzessive aufgrund lokaler Nachbarschaftsstrukturen aufgebaut wird. Die Anwendung dieser Methode für par-DOMINATING SET auf planaren Graphen liefert Folgendes [Alb03, AFF+01].

Theorem 2 (Search Tree Algorithm): *Für einen planaren Graph mit n Knoten können in Laufzeit $O(8^k\, n)$ alle Dominating Sets der Größe höchstens k gefunden werden.*

Die Analyse des zugehörigen Suchbaums ist sehr technisch und umfangreich. Nachdem ein früheres $O(11^k\, n)$ Resultat von Downey und Fellows [DF99, Theorem 3.1] einen (of-

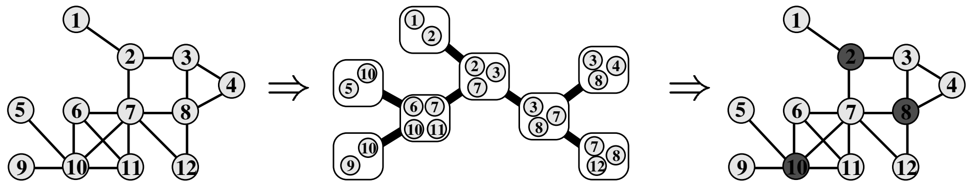

Abbildung 2: Typischer 2-phasiger baumzerlegungsbasierter Algorithmus: In einer ersten Phase wird zu dem Graph eine Baumzerlegung möglichst kleiner Baumweite erstellt. Auf dieser Baumzerlegung wird das Optimierungsproblem durch dynamisches Programmieren für den Ausgangsgraph gelöst.

fensichtlichen) Fehler enthält, liefert unser Ergebnis — unseres Wissens nach — den ersten (korrekten) Fest-Parameter-Algorithmus der Laufzeit $O(c^k\, n)$ für par-DOMINATING SET auf planaren Graphen.

Zentrale Beobachtung für die Verzweigung innerhalb des Suchbaums ist Folgende: Für eine zu konstruierende optimalen Dominating Set muss zu gegebenem Knoten v mindestens einer der Knoten aus der geschlossenen Nachbarschaft $N[v]$ von v gehören. Verzweigt wird jeweils nach einem möglichst kleingradigen, im bisherigen Verlauf des Algorithmus noch nicht dominierten Knoten. Der Beweis für die Existenz eines solchen kleingradigen Knotens ist der zentrale Kern für die Analyse der Suchbaumgröße. Im übrigen lässt sich nachweisen, dass die Schranke auf den maximalen Verzweigungsgrad 8 scharf ist.

In der Praxis zeigt sich, dass die Suchbaumgröße, d.h. die tatsächlich auftretenden Verzweigungsgrade des Suchbaums, deutlich niedriger zu sein scheint als dies die Worst-Case-Analyse vorgibt.

2.3 Algorithmen basierend auf Baumzerlegungen

In diesem Abschnitt wenden wir uns dem Konzept von *Baumzerlegungen* zu. Wir stellen eine Methode vor, mit welcher man für parametrisierte Probleme unter gewissen, leicht zu verifizierenden Voraussetzungen Fest-Parameter-Algorithmen der Laufzeit $2^{O(\sqrt{k})} + n^{O(1)}$ gewinnen kann [ABF+02]. Hierunter fallen u.a. par-VERTEX COVER, par-INDEPENDENT SET und par-DOMINATING SET auf planaren Graphen. Unserer Erkenntnis nach sind dies die ersten Fest-Parameter-Algorithmen "sublinear-exponentieller" Laufzeit (d.h. die Funktion f hat einen sublinearen Exponenten) überhaupt in der Literatur. Darüber hinaus zeigen wir, dass diese Algorithmen asymptotisch optimal sind: Genauer gesagt würde ein Algorithmus der Laufzeit $2^{o(\sqrt{k})}\, n^{O(1)}$ für eines dieser Probleme bereits 3 SAT $\in$ DTIME$(2^{o(n)})$ implizieren (n ist hierbei die Anzahl der Variablen einer 3 SAT Formel), was in der (klassischen) Komplexitätstheorie als eher unwahrscheinlich angesehen wird.

Grundidee. Baumzerlegungen, bzw. der *Baumweite*-Begriff haben ihren Ursprung in der Theorie der Graph-Minor von Robertson und Seymour. Aus algorithmischer Sicht stellen Baumzerlegungen ein interessantes Konzept im Design von Fest-Parameter-Algorithmen dar. Baumzerlegungsbasierte Algorithmen setzen sich typischerweise aus zwei Phasen zusammen (siehe Abbildung 2):

Phase 1: Finde eine Baumzerlegung (möglichst kleiner Weite) für den Graph.

Phase 2: Löse das Problem durch dynamisches Programmieren auf der Baumzerlegung.

Da die Weite der Baumzerlegung meist exponentiell in die Laufzeit der zweiten Phase eingeht, sind kleine Baumweiten wünschenswert. Die Arbeit liefert neue Ergebnisse bezüglich beider Phasen der oben genannten Methode.

Informell gesprochen ist die Baumweite ein Maß für die Baumähnlichkeit eines Graphs. Formal liegt folgende Definition zugrunde (siehe [Bod98] für Details zu Baumzerlegungen).

Definition 3 *Sei* $G = (V, E)$ *ein Graph. Eine* Baumzerlegung $\mathcal{X}$ *von* G *ist ein Paar* $\langle \{ X_i \mid i \in V(T) \}, T \rangle$*, wobei* T *ein Baum ist, und die* X_i *Knotenteilmengen von* V *(sog.* Bags*) sind. Dabei müssen folgende 3 Bedingungen erfüllt sein.*

(i) *Jeder Graph-Knoten muss in der Baumzerlegung auftauchen:* $\bigcup_{i \in I} X_i = V$*;*

(ii) *Jede Graph-Kante muss in der Baumzerlegung auftauchen: für jede Kante* $\{u, v\} \in E$*, gibt es ein* $i \in V(T)$*, so dass* $\{u, v\} \subseteq X_i$*;*

(iii) *Die Baumzerlegung muss* konsistent *sein: falls* j *auf dem Pfad von* i *nach* k *in* T *liegt* $(i, j, k \in V(T))$*, dann gilt* $X_i \cap X_k \subseteq X_j$.

Die Weite $\mathrm{tw}(\mathcal{X})$ *von* $\mathcal{X}$ *ist definiert als* $\mathrm{tw}(\mathcal{X}) := \max\{ |X_i| \mid i \in V(T) \} - 1$. *Die* Baumweite $\mathrm{tw}(G)$ *des Graphs* G *ist das kleinste* ℓ*, so dass* G *eine Baumzerlegung der Weite* ℓ *zulässt.*

Als Beispiel für eine Baumzerlegung betrachte Abbildung 2: Die mittlere Abbildung liefert eine Baumzerlegung der Weite 3 für den Graph in der linken Abbildung.

Phase 1: Konstruktion von Baumzerlegungen. Das Problem, zu gegebenem Graph G und einer Konstante ℓ zu bestimmen, ob G eine Baumzerlegung der Weite ℓ zulässt und diese gegebenenfalls zu konstruieren, ist NP-hart. Hält man ℓ fest, so lässt sich dieses Problem sogar in Linearzeit lösen. Der beste Algorithmus geht auf ein bekanntes Resultat von Bodlaender zurück und hat Laufzeit $O(2^{\Theta(\ell^3)}\, n)$ — selbst für kleine Baumweiten ist dieser Algorithmus praktisch nahezu unbrauchbar [Bod98].

In der Dissertation [Alb03] widmen wir uns der Konstruktion von Baumzerlegungen unter der Voraussetzung, dass gewisse Problemparameter (etwa die "vertex cover number" oder die "domination number") beschränkt sind. Eines der Hauptresultate der Dissertation [Alb03] ist in diesem Zusammenhang der Nachweis von konstruktiven oberen Schranken für die Baumweite $\mathrm{tw}(G)$ eines planaren Graphs G. Fundamental hierbei ist der von uns eingeführte Begriff der "Layerwise Separation Property" (LSP). Es wird gezeigt, dass für jede "Ja"-Instanz (G, k) eines parametrisierten LSP-Problems eine Baumzerlegung der Weite höchstens $c\sqrt{k}$ effizient in Zeit $O(\sqrt{k}\, n)$ konstruiert werden kann. Aus diesem allgemeinen Resultat folgen hieraus speziell für das par-VERTEX COVER bzw. das par-DOMINATING SET Problem folgende Schranken.

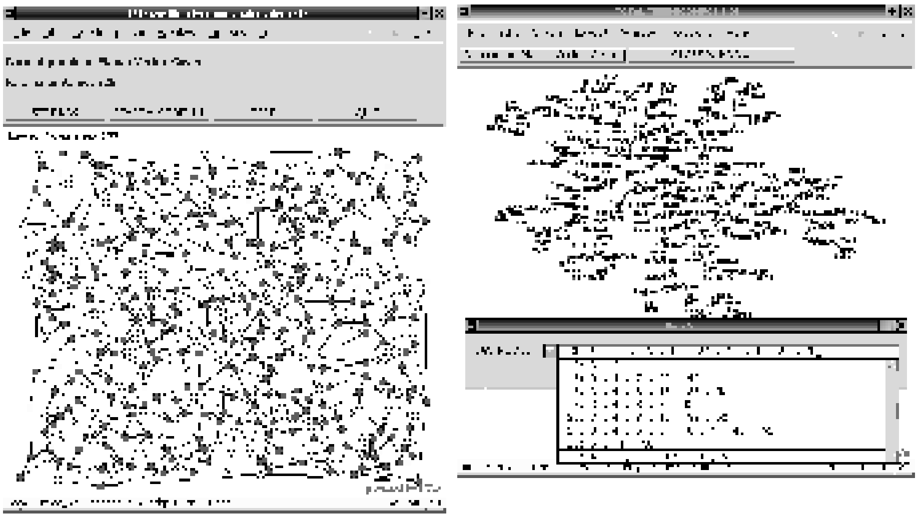

Abbildung 3: In der “FPT-Toolbox” ist u.a. der baumzerlegungsbasierte Ansatz implementiert. Die Lösung des par-VERTEX COVER Problems (im linken Bild) wurde vermöge der zunächst errechneten Baumzerlegung (s. rechtes Bild) gefunden.

Theorem 4 *Für die Baumweite* $\mathrm{tw}(G)$ *eines planaren Graphs* G *gilt:*

$$\mathrm{tw}(G) \leq 4\sqrt{3\,\mathrm{vc}(G)} + 5 \quad \textit{und} \quad \mathrm{tw}(G) \leq 6\sqrt{34\,\mathrm{ds}(G)} + 8,$$

wobei $\mathrm{vc}(G)$ *die Größe einer optimalen Knotenüberdeckung (vertex cover number) bzw.* $\mathrm{ds}(G)$ *die Größe einer optimalen dominierenden Knotenmenge (domination number) bezeichnen. Beide Schranken sind asymptotisch optimal.*

Eine solche Baumzerlegung kann in Zeit $O(\mathrm{vc}(G)\,n)$ *bzw.* $O(\mathrm{ds}(G)\,n)$ *konstruiert werden.*

Die asymptotische Optimalität dieser Schranken erkennt man durch Betrachtung des planaren $m \times m$-Gittergraphs G_m mit $n = m^2$ Knoten. Hierfür gilt $\mathrm{tw}(G_n) = m$ und $\mathrm{vc}(G_n) = \lfloor \frac{m^2}{2} \rfloor$ bzw. $ds(G_n) = \frac{1}{5}(m^2 + m - 3)$. Wohingegen für nicht-planare Graphen G noch die scharfe Abschätzung $\mathrm{tw}(G) \leq \mathrm{vc}(G)$ Gültigkeit hat, ist im Fall von DOMINATING SET keine Schranke der Form $\mathrm{tw}(G) \leq f(\mathrm{ds}(G))$ für eine beliebige Funktion f möglich: Eine Clique K_n mit n Knoten hat Baumweite $n - 1$ und kann durch einen einzigen Knoten dominiert werden.

Im übrigen lässt sich aus obigem Hauptresultat leicht folgende Aussage beweisen.

Corollary 5 *Für einen planaren Graph* G *gilt:*

$$\mathrm{tw}(G) \leq 2\sqrt{6n} + 2 \approx 4.90\sqrt{n} + 2$$

Dies verbessert — im planaren Fall — eine zuvor von Alon, Seymour und Thomas nachgewiesene Schranke von $\mathrm{tw}(G) \leq h^{\frac{3}{2}}\sqrt{n}$ für K_h-Minor freie Graphen, welche (für $h = 5$) im planaren Fall $\mathrm{tw}(G) \leq 11.18\sqrt{n}$ liefert.

Phase 2: Dynamisches Programmieren auf Baumzerlegungen. Eine Vielzahl von NP-harten Problemen auf Graphen kann bei gegebener Baumzerlegung mit beschränkter Baumweite in polynomieller Laufzeit (gemessen in der Graphgröße) gelöst werden. In der Regel verwendet man hier einen dynamischen Programmieransatz. Dabei geht die Baumweite der Zerlegung meist exponentiell in die Laufzeit ein.

In der Arbeit [Alb03, AN02] werden neue Resultate zum dynamischen Programmieren auf Baumzerlegungen für eine Reihe von "Dominierungs"-Problemen vorgestellt. Durch ein Konzept, das auf einer Art "Monotonieverhalten" innerhalb des dynamischen Programmierens beruht, erhalten wir folgendes Resultat.

Theorem 6 DOMINATING SET *kann für Graphen mit gegebener Baumzerlegung in Laufzeit* $O(4^{\ell}\, N)$ *gelöst werden. Hierbei ist* ℓ *die Weite der gegebenen Baumzerlegung und* N *die Anzahl der Knoten im Baum.*

Der zuvor beste bekannte Algorithmus hierzu hatte Laufzeit $O(9^{\ell}\, N)$ [TP97] und konnte somit erheblich verbessert werden. Ähnliche Verbesserungen erzielen wir beispielsweise für VERTEX COVER, INDEPENDENT DOMINATING SET, TOTAL DOMINATING SET, PERFECT DOMINATING SET, oder PERFECT CODE. Die nachfolgende Tabelle skizziert kurz, wie sich die Laufzeitverbesserung unseres $O(4^{\ell}\, N)$-Algorithmus gegenüber dem bisher bekannten $O(9^{\ell}\, N)$-Algorithmus für verschiedene Baumweiten praktisch auswirkt. Die Grundannahme hier ist eine Baumzerlegung mit $N = 1000$ Knoten und eine Maschine mit 10^9 Instructions per Second.

Algorithm	$w = 5$	$w = 10$	$w = 15$	$w = 20$
$9^w \cdot N$	0.05 sec	1 hour	6.5 years	$3.9 \cdot 10^5$ years
$4^w \cdot N$	0.001 sec	1 sec	18 minutes	13 days

Fest-Parameter-Algorithmen mit subexponentieller Laufzeit. Die Verschmelzung der Ergebnisse zu den beiden Phasen führt zu einem universellen Schema, mit welchem, unter gewissen, leicht zu verifizierenden Voraussetzungen, Fest-Parameter-Algorithmen mit Laufzeitverhalten $O(2^{O(\sqrt{k})}\, n)$ oder $2^{O(\sqrt{k})} + n^{O(1)}$ (d.h. Algorithmen mit sublinearem Term im exponentiellen Laufzeitanteil) für viele parametrisierte LSP-Probleme gewonnen werden können [Alb03, AFN01]. Insbesondere erhalten wir:

Theorem 7 *Für einen planaren Graph der Größe* n *lässt sich* par-VERTEX COVER *in Laufzeit* $O(2^{4\sqrt{3k}}\, k + kn)$, par-INDEPENDENT SET *in Laufzeit* $O(2^{4\sqrt{6k}}\, k + n^2)$ *und* par-DOMINATING SET *in Laufzeit* $O(2^{12\sqrt{17\,\log(3)\,k}}\, k + n^3)$ *lösen. Hierbei bezeichnet jeweils* k *den Problemparameter.*

Vergleichbare Ergebnisse gelten für eine Reihe von Variationen dieser Probleme, u.a. für sogenannte par-DOMINATING SET WITH PROPERTY P Probleme oder für das par-FACE COVER Problem. Abbildung 3 zeigt eine Implementierung dieser baumzerlegungsbasierten Algorithmen im Fall von par-VERTEX COVER.

3 Zusammenfassung und Ausblick

Implementierung und Experimente. In der Arbeit [Alb03] stellen wir abschließend ein Software-Paket, die "FPT-Toolbox", vor, welches im Rahmen dieses Projektes mit LEDA [MN99] entwickelt wurde und eine Vielzahl der genannten Algorithmen implementiert. Abschließend wird über eine Reihe von empirischen Studien zur Auswertung der Praxistauglichkeit dieser Algorithmen berichtet [Alb03, ADNar, ABN03]. Die Leistungstests basieren auf verschiedenen der Praxis entlehnten Netzwerken [ABN03] und kombinatorisch generierten Zufallsgraphinstanzen. In den Tests bestätigt sich, dass unsere theoretische Analyse der Worst-Case-Szenarien oftmals (viel) zu pessimistisch für tatsächliche Aussagen über die praktische Güte der Algorithmen ist. Speziell wollen wir folgende Beobachtungen anbringen: Die in der Praxis erhaltenen Baumzerlegungen hatten eine bei weitem geringere Baumweite, als dies aus der theoretischen Analyse zu erwarten gewesen wäre. Folglich erweist sich das gesamte Konzept der Baumzerlegungen in unseren Tests als sehr vielversprechend und effizient. Schließlich ist die eindrucksvolle Stärke der Datenreduktions-Algorithmen hervorzuheben [Alb03, AFNar]. Wohingegen unsere Tests ergaben, dass die von Nemhauser und Trotter vorgeschlagene Problemkernreduktion für par-VERTEX COVER die Eingabeinstanzen um durchschnittlich ca. 65% verkleinerte, konnten die von uns entwickelten Datenreduktionsregeln für par-DOMINATING SET die Eingabegraphen gar um über 99% reduzieren. Durch Hinzunahme dieser effizienten Problemkernreduktionen als "Preprocessing" konnten demnach die meisten Algorithmen signifikant beschleunigt werden.

Ausblick. Die Ergebnisse der vorliegenden Dissertation [Alb03] hatten bereits starken Einfluss auf die Forschungsaktivitäten innerhalb der Parametrisierten Komplexität. Spezielle Beachtung schenkte man dabei den Algorithmen mit (erstmalig) subexponentiellem Laufzeitverhalten (d.h. Laufzeiten der Form $2^{O(\sqrt{k})}n^{O(1)}$). Diese Resultate eröffneten einen neuen Zweig in der zukünftigen Forschung und zählen — mit weit über einem Dutzend Referenzen — bereits jetzt zu den am meisten zitierten Beiträgen in der Parametrisierten Komplexität innerhalb des letzten Jahres.

Literatur

[ABF+02] 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(4):461–493, 2002.

[ABN03] J. Alber, N. Betzler und R. Niedermeier. Computing optimal dominating sets in large networks. In *Proc. INOC 2003*, Seiten 1–6, 2003.

[ADNar] J. Alber, F. Dorn und R. Niedermeier. Experimental evaluation of a tree decomposition based algorithm for vertex cover on planar graphs. *Disc. Appl. Mathematics*, to appear.

[AFF+01] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond und U. Stege. Refined search tree technique for dominating set on planar graphs. In *Proc. 26th MFCS*, LNCS 2136, Seiten 111–122. Springer-Verlag, 2001. Accepted for *JCSS*.

[AFN01] J. Alber, H. Fernau und R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. In *Proc. 28th ICALP*, LNCS 2076, Seiten 261–272. Springer-Verlag, 2001. Accepted for *Journal of Algorithms*.

[AFNar] J. Alber, M. R. Fellows und R. Niedermeier. Polynomial time data reduction for Dominating Set. *Journal of the ACM*, to appear. Extended abstract in *Proc. 8th SWAT*, Springer-Verlag LNCS 2368, Seiten 150–159, 2002.

[AGN01] J. Alber, J. Gramm und R. Niedermeier. Faster exact solutions for hard problems: a parameterized point of view. *Discrete Mathematics*, 229(1):3–27, 2001.

[Alb03] J. Alber. *Exact algorithms for NP-hard Problems on Networks: Design, Analysis, and Implementation.* Dissertation, WSI für Informatik, Universität Tübingen, 2003.

[AN02] J. Alber und R. Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *Proc. 5th LATIN*, LNCS 2286, Seiten 613–627. Springer-Verlag, 2002.

[Bod98] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.

[DF99] R. G. Downey und M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998.

[HHS98] T. W. Haynes, S. T. Hedetniemi und P. J. Slater. *Fundamentals of Domination in Graphs.*, Jgg. 208 of *Monogr. and textbooks in Pure and Appl. Math.* Marcel Dekker, 1998.

[MN99] K. Mehlhorn und S. Näher. *LEDA: A Platform of Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

[TP97] J. A. Telle und A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM Journal on Discrete Mathematics*, 10(4):529–550, 1997.

Werdegang

Jochen Alber wurde am 28.06.1973 in Göppingen geboren. Er besuchte das Raichberg-Gymnasium in Ebersbach, Baden-Württemberg. An diesem Gymnasium war er für mehrere Jahre Schülersprecher. 1993 schloss Jochen Alber mit dem Abitur (Note 1.0) ab; für seine Leistungen erhielt er zahlreiche Preise. Nach seiner Zivildienstzeit in Ulm, studierte er an der Eberhard-Karls-Universität Tübingen und an der University of Washington in Seattle, USA, Mathematik mit Nebenfach Informatik. Jochen Alber war einer der ersten Studenten des Internationalen Studiengangs Mathematik (ISM) und erhielt 1999 nach 8.5 Fachsemestern das Diplom "mit Auszeichnung". Während der Studienzeit war er u.a. bei IBM Global Services in Böblingen beschäftigt. Anschließend promovierte Jochen Alber als wissenschaftlicher Mitarbeiter am WSI für Informatik in Tübingen. Dort reichte er im Oktober 2002 seine Dissertation [Alb03] ein. Im Januar 2003 schloss er die Promotion zum Dr. rer. nat. mit "summa cum laude" ab. Für die Arbeit erhielt er den Promotionspreis 2003 der Universität Tübingen. Seine Forschungsarbeiten haben ihren Schwerpunkt in erster Linie in der Algorithmik, der Optimierungs- und Graphtheorie. In der Folge arbeitete er für die IT-Abteilung "Common and Basic Systems" der Allianz-Lebensversicherungs AG in Stuttgart im Bereich Dokumenten-Management. Seit April 2004 ist Jochen Alber bei der DIgSILENT GmbH in Gomaringen für die Entwicklung von PowerFactory, einer Software zur digitalen Simulation und Berechnung elektrischer (Energieversorgungs-)Netzwerke, zuständig. Er ist dabei für die Bereiche Algorithmik und Numerik und für Optimierungsfragen verantwortlich.