Friedrich-Alexander-Universität
Erlangen-Nürnberg

Sammlung der Zusammenfassungen

der Vorträge auf dem

51. Workshop über Datenstrukturen, Effiziente Algorithmen und Komplexitätstheorie

Theorie Tag

15. März 2005

Friedrich-Alexander-Universität

Erlangen-Nürnberg

Rolf Wanka (Hrsg.)

Co-Design-Report 03-2005

Programm

9:00	*Begrüßung*
9:05	**Smoothed Analysis, Permutationen und binäre Suchbäume** *Bodo Manthey, Rüdiger Reischuk,* Universität zu Lübeck
9:30	**Smoothed Analysis of Integer Programming** *Heiko Röglin, Berthold Vöcking,* RWTH Aachen
9:55	**Balanced Allocation and Dictionaries with Tightly Packed Constant Size Bins** *Martin Dietzfelbinger, Christoph Weidling,* TU Ilmenau
10:20	*Kaffeepause*
10:45	**Die Komplexität hybrider Logiken über transitiven Rahmen** *Martin Mundhenk, Thomas Schneider*, Friedrich-Schiller-Universität Jena
11:10	**Kombinatorische Inferenz azyklischer Beziehungstypen im Internet** *Sven Kosub, Moritz G. Maaß, Hanjo Täubig,* TU München
11:35	**Neue untere Schranken für die Beschreibungskomplexität der Erreichbarkeit** *Till Tantau*, TU Berlin
12:00	Mittagspause
14:00	**Komplexität von DNF und Isomorphie für monotone Formeln** *Judy Goldsmith,* University of Kentucky *Matthias Hagen, Martin Mundhenk,* Friedrich-Schiller-Universität Jena
14:25	**Erfüllbarkeitsprobleme für gemischte Horn-Formeln** *Stefan Porschen, Ewald Speckenmeyer*, Universität zu Köln
14:50	**Isomorphic Implication** *Michael Bauland*, Universität Hannover *Edith Hemaspaandra,* Rochester Institute of Technology
15:15	*Kaffeepause*
15:35	**Bounded Degree Closest *k*-Tree Power is NP-Complete** *Michael Dom, Jiong Guo, Rolf Niedermeier,* Friedrich-Schiller-Universität Jena
16:00	**Algorithm Engineering for Optimal Graph Bipartization** *Falk Hüffner*, Friedrich-Schiller-Universität Jena
16:25	**Toleranzbasierende Algorithmen für das Travelling Salesman Problem** *Gerold Jäger*, Martin-Luther-Universität Halle-Wittenberg
16:50	*Kaffeepause*
17:10	**Verhältnis von AvgP und HP zu Error-Klassen** *Arfst Nickelsen, Birgit Schelm,* TU Berlin und Universität Hannover
17:35	**Dynamic Complexity Theorie Revisited** *Volker Weber, Thomas Schwentick*, Philipps-Universität Marburg
18:00	*Ende des Workshops*

Smoothed Analysis, Permutationen und binäre Suchbäume

Bodo Manthey

Universität zu Lübeck
Institut für Theoretische Informatik
Ratzeburger Allee 160, 23538 Lübeck
manthey@tcs.uni-luebeck.de

Smoothed Analysis wurde 2001 von Spielman und Teng (J. ACM 51, pp. 385–463, 2004) eingeführt, um die Diskrepanz zwischen Average-Case- und Worst-Case-Laufzeit des Simplex-Algorithmus zu erklären. Worst-Case-Instanzen sind meist künstlich und tauchen in der Praxis kaum auf. Andererseits ist die Average-Case-Komplexität meist zu optimistisch, da typische Instanzen in der Regel nicht völlig zufällig sind.

Bei der *Smoothed Analysis* werden (Worst-Case-)Instanzen leicht gestört. Sei X die Menge aller Instanzen, C ein Komplexitätsmaß und $\Delta(x,\sigma)$ die Wahrscheinlichkeitsverteilung, bezüglich der $x \in X$ gestört wird. Die geglättete Komplexität (*smoothed complexity*) ist definiert als

$$\max_{x \in X} \mathbb{E}_{y \sim \Delta(x,\sigma)} C(y) \,.$$

Hierbei ist σ ein Maß für die Stärke der Störung: Je kleiner σ ist, desto näher ist die geglättete Komplexität an der Worst-Case-Komplexität.

Banderier et al. (28th MFCS, LNCS 2747, pp. 198–207, 2003) untersuchten unter anderem Quicksort und die Anzahl der Left-To-Right-Maxima, wobei die Eingabelisten leichten Störungen unterworfen sind. Sie schlugen *p-partielle Permutationen* als Störungsmodell vor: Jedes Element wird mit einer festen Wahrscheinlichkeit p markiert, und anschließend werden alle markierten Elemente zufällig permutiert.

Wir untersuchen die Höhe von binären Suchbäumen unter leichten Störungen. Unter p-partiellen Permutationen hat jeder binäre Suchbaum mit n Elementen erwartete Höhe $O(\sqrt{n/p})$. Diese Schranke ist scharf, d. h. es gibt Bäume, deren erwartete Höhe unter p-partiellen Permutationen $\Omega(\sqrt{n/p})$ ist. Ohne Störungen ist die Höhe im Worst-Case n, bei einer völlig zufälligen Permutation ist die erwartete Höhe $\Theta(\log n)$.

Neben partiellen Permutationen stellen wir zwei weitere Störungsmodelle vor und vergleichen die verschiedenen Modelle. Außerdem untersuchen wir die Stabilität von Störungen, d. h. die Frage, um wie viel die Höhe unter Störung ansteigen kann.

Der Vortrag basiert auf einer Arbeit zusammen mit Rüdiger Reischuk.

Smoothed Analysis of Integer Programming

Heiko Röglin

Department of Computer Science –
RWTH Aachen
roeglin@cs.rwth-aachen.de

We present a probabilistic analysis of integer linear programs (ILPs). More specifically, we study ILPs in a so-called smoothed analysis in which it is assumed that first an adversary specifies the coefficients of an integer program and then (some of) these coefficients are randomly perturbed, e.g., using a Gaussian or a uniform distribution with small standard deviation. In this probabilistic model, we investigate structural properties of ILPs and apply them to the analysis of algorithms. For example, we prove a lower bound on the slack of the optimal solution. As a result of our analysis, we are able to specify the smoothed complexity of classes of ILPs in terms of their worst case complexity. For example, we obtain polynomial smoothed complexity for packing and covering problems with any fixed number of constraints. Previous results of this kind were restricted to the case of binary programs.

Joint work with Berthold Vöcking

Balanced Allocation and Dictionaries with Tightly Packed Constant Size Bins

Martin Dietzfelbinger, Christoph Weidling

Technische Universität Ilmenau

{martin.dietzfelbinger,christoph.weidling}@tu-ilmenau.de

We study a particular aspect of the balanced allocation paradigm (also known as the "two-choices paradigm"): constant sized bins, packed as tightly as possible. Let $d \geq 1$ be fixed, and assume there are m bins of capacity d each. To each of $n \leq dm$ balls two possible bins are assigned at random. How close can $dm/n = 1+\varepsilon$ be to 1 so that with high probability each ball can be put into one of the two bins assigned to it, without any bin overflowing? We show that $\varepsilon > (2/e)^{d-1}$ is sufficient. If a new ball arrives with two new randomly assigned bins, we wish to rearrange some of the balls already present in order to accommodate the new ball. We show that on average it takes constant time to rearrange the balls to achieve this, for $\varepsilon > \gamma \cdot \beta^d$, for some constants $\gamma > 0$, $\beta < 1$. An alternative way to describe the problem is in data structure language. Generalizing cuckoo hashing (Pagh and Rodler, 2001), we consider a hash table with m positions, each representing a bucket of capacity $d \geq 1$. Keys are assigned to buckets by two fully random hash functions. How many keys can be placed in these bins, if key x may go to bin $h_1(x)$ or to bin $h_2(x)$? Our results lead to an implementation of a dynamic dictionary that accommodates n keys in $m = (1+\varepsilon)n/d$ buckets of size $d = O(\log(1/\varepsilon))$, so that key x resides in bucket $h_1(x)$ or $h_2(x)$. If $d \geq 1 + 3.26 \cdot \ln(1/\varepsilon)$, then for a lookup operation only two hash functions have to be evaluated and two contiguous segments of d memory cells have to be inspected. The expected time for inserting a new key is constant, for some $d = O(\log(1/\varepsilon))$.

Die Komplexität hybrider Logiken über transitiven Rahmen

Martin Mundhenk, Thomas Schneider

Institut für Informatik,
Friedrich-Schiller-Universität Jena
{mundhenk,schneider}@cs.uni-jena.de

Modale Logik ist eine ausdrucksstarke Erweiterung der klassischen Aussagenlogik. Eine Ausprägung davon ist temporale Logik, mit deren Hilfe man das zeitliche Verhalten von Dingen beschreiben kann. Dort kann man zusätzlich zur Sprache der klassischen Aussagenlogik über Operatoren verfügen, die besagen, „es gibt einen Zeitpunkt in der Vergangenheit/Zukunft, zu dem φ wahr ist“ (Future/Past) oder „in der Zukunft wird φ wahr sein, und bis dahin gilt immer ψ“ (Until bzw. Since).

Temporale Logik wird über Rahmen interpretiert. Das sind Strukturen, die aus einer Menge von Zeitpunkten und einer binären Relation („später als“) bestehen. Ohne Zweifel ist die Klasse aller transitiven Rahmen eine der wichtigsten Klassen für temporale Anwendungen.

Will man Zeitpunkte direkt benennen, so reicht das Ausdrucksvermögen der „einfachen“ temporalen Logik nicht aus. Hybride Logik besitzt zusätzlich sprachliche Mittel, die das leisten. So kann man Zeitpunkten Namen zuweisen (mittels Nominals), zu benannten Zeitpunkten springen (mittels des @-Operators) und Variablen an Zeitpunkte binden (mittels des $\downarrow$-Operators).

Über die Komplexität des Erfüllbarkeitsproblems für hybride Sprachen ist bereits einiges bekannt: Je nach Umfang der Sprache und zugrunde liegender Klasse von Rahmen (der Strukturen, über denen die jeweilige Logik interpretiert wird) sind NP-, PSPACE-, EXPTIME- und Unentscheidbarkeitsresultate bekannt. Allerdings gibt es Lücken für bestimmte Sprachen über eingeschränkten Klassen von Rahmen, von denen wir zwei schließen konnten. Wir zeigen, dass das Erfüllbarkeitsproblem über transitiven Rahmen für die hybride Sprache mit Since und Until EXPTIME-vollständig ist. Dies ist dieselbe Komplexität wie für diese Sprache über allgemeinen Rahmen. Wir beweisen außerdem, dass das Erfüllbarkeitsproblem über transitiven Rahmen für Fragmente zweier hybrider $\downarrow$-Sprachen unentscheidbar ist.

[1] M. Mundhenk, T. Schneider: “New Pages for the Road-Map: Complexity of Hybrid Logics over Transitive Frames.”
http://www.minet.uni-jena.de/~thschnei/publ/index.html

Kombinatorische Inferenz azyklischer Beziehungstypen im Internet

Sven Kosub Moritz G. Maaß Hanjo Täubig

Fakultät für Informatik, Technische Universität München
Boltzmannstraße 3, 85748 Garching b. München
{kosub,maass,taeubig}@in.tum.de

Die Gewährleistung von Routingstabilität im Internet ist eine Herausforderung für jede Netzwerkadministration. Ist Routing innerhalb eines *autonomous system* noch unter weitgehend vollständiger Kontrolle der Administratoren, so basiert *interdomain routing* vor allem auf zwischen unabhängigen Internetteilnehmern vereinbarten vertraglichen Beziehungen, die ihren Niederschlag in unterschiedlichen Import- und Exportregeln für BGP-Daten und damit in eingeschränkter Erreichbarkeit finden. Solche Regeln aus beobachtbaren Daten abzuleiten ist von großer Bedeutung für effizientes *traffic engineering*.

In diesem Vortrag werden wir dahin gehend das Problem betrachten, aus Informationen, die in den BGP-Routing-Tabellen enthalten sind, auf die vertraglichen Beziehungen (*customer-to-provider*, *peering-to-peering* und *sibling-to-sibling*) zwischen den autonomen Systemen zu schließen. Dazu geben wir graphentheoretische Charakterisierungen für dieses Problem an und diskutieren algorithmische Lösungen für darauf aufbauende kombinatorische Optimierungsprobleme. Von besonderem Interesse bei der Modellierung der Probleme ist hierbei die Annahme wirtschaftlicher Rationalität bei den Internetteilnehmern.

Neue untere Schranken für die Beschreibungskomplexität der Erreichbarkeit

Till Tantau

Fachgebiet Theoretische Informatik – Algorithmik und Logik
Fakultät für Elektrotechnik und Informatik
TU Berlin
`tantau@cs.tu-berlin.de`

In der deskriptiven Komplexitätstheorie versucht man, möglichst kleine oder einfache Formeln zu finden, die Probleme beschreiben. Ein wichtiges Maß ist die Quantorenkomplexität, welche angibt, wie viele verschachtelte Quantoren benötigt werden, um ein Problem zu beschreiben. Es ist schon lange bekannt, dass man zur Beschreibung des Erreichbarkeitsproblem für gerichteten Graphen genau $\log d$ erststufige Quantoren braucht, wobei d der Abstand von Start und Ziel ist. Im Vortrag soll die Beschreibungskomplexität statt in Abhängigkeit vom Abstand der Knoten, in Abhängigkeit von der Unabhängigkeitszahl α des Graphen betrachtet werden. Für $\alpha \leq 3$ soll gezeigt werden, dass man genau $\alpha + 3$ geschachtelte Quantoren braucht, um Erreichbarkeit in Graphen mit Unabhängigkeitszahl höchstens α zu beschreiben.

Komplexität von DNF und Isomorphie für monotone Formeln

Judy Goldsmith
University of Kentucky
goldsmit@cs.uky.edu

Matthias Hagen Martin Mundhenk
Friedrich-Schiller-Universität Jena
{hagen,mundhenk}@informatik.uni-jena.de

Die exakte Komplexität der Berechnung einer (minimalen) Disjunktiven Normalform (DNF) für aussagenlogischen Formeln ist unbekannt. Sie ist offensichtlich nicht in Polynomialzeit möglich, da es Formeln mit exponentiell langen DNFen gibt. Wir betrachten die Komplexität des Problems für monotone Formeln – aussagenlogische Formeln mit den Verknüpfungszeichen $\wedge$ (Konjunktion) und $\vee$ (Disjunktion), aber ohne $\neg$ (Negation). Viele Probleme (z.B. Erfüllbarkeit) sind für monotone Formeln einfacher als für beliebige Formeln. Es gibt aber auch Probleme (z.B. Äquivalenz), die für beide Formelarten gleich schwer sind. Wir zeigen, dass die Berechnung der Größe einer minimalen DNF für monotone Formeln $\#\mathsf{P}$-vollständig ist. Die Berechnung einer minimalen DNF für monotone Formeln liegt offensichtlich zwischen Polynomial- und Exponentialzeit. Ein Algorithmus hat Ausgabe-Polynomialzeit (Papadimitriou 1997), wenn seine Rechenzeit durch ein Polynom in der Summe von Ein- und Ausgabelänge beschränkt werden kann. Solche Algorithmen haben also nur für lange Ausgaben lange Rechenzeit. Wir zeigen, dass die minimale DNF von monotonen Formeln in Ausgabe-Polynomialzeit berechenbar ist genau dann, wenn $\mathsf{P} = \mathsf{NP}$. Minimale DNFen bestehen aus Primimplikanten. Wir zeigen, dass das Erkennen eines Primimplikanten für eine (nicht-monotone) Formel DP-vollständig ist (DP liegt zwischen NP und Σ_2^p). Für monotone Formeln liegt das Problem dagegen in L.

Schließlich betrachten wir noch die Isomorphie von Formeln. Zwei Formeln sind isomorph, wenn man sie durch Umbenennung der Variablen äquivalent machen kann. Wir zeigen, dass das Isomorphie-Problem für monotone Formeln genau so schwer ist wie für nicht-monotone Formeln.

Satisfiability of Mixed Horn Formulas

Stefan Porschen   Ewald Speckenmeyer

Institut für Informatik,
Universität zu Köln,
50969 Köln, Germany.
{porschen,esp}@informatik.uni-koeln.de

In recent time the interest in designing exact algorithms providing better upper time bounds than the trivial ones for NP-complete problems and their NP-hard optimization counterparts has increased. Of particular interest in this context is the investigation of exact algorithms for testing the satisfiability (SAT) of propositional formulas in conjunctive normal form (CNF). This interest stems from the fact that SAT is well known to be a fundamental NP-complete problem appearing naturally or via reduction as the abstract core of many application-relevant problems. Not only the whole class CNF is of interest in this context. In several applications subclasses of CNF are of importance for which SAT unfortunately remains NP-complete. Nevertheless, it is often possible by exploiting the specific structure of such formulas to design fast exact algorithms for their solution.

In this talk we discuss such a class of formulas, called *mixed Horn formulas (MHF's)*. Roughly speaking, a MHF M consists of a positive monotone 2-CNF formula P (containing only 2-clauses) and a Horn formula H, i.e., $M = H \wedge P$. The introduction and investigation of MHF's is by no means artificial. Well known problems for level graphs, like level-planarity test or the NP-hard crossing-minimization problem, can be formulated conveniently in terms of MHF's. This was our motivation for considering MHF's. Also graph colorability naturally leads to MHF's.

It is well known that 2-SAT and Horn-SAT are solvable in linear time, but SAT for MHF's (shortly MHF-SAT) remains NP-complete. A closely related class generalizing the classes of Horn and quadratic formulas is the class of so-called q-Horn formulas introduced by Boros et al., for which SAT can be solved in linear time also. A q-Horn formula can be considered as a specific mixed Horn formula. The class of q-Horn formulas probably is the largest subclass of MHF that is SAT-solvable in polynomial time.

The main purpose of this talk is to prove a non-trivial worst case upper time bound for solving MHF-SAT, namely $O(2^{0.5284n})$ where n is the number of variables in the input formula. Moreover, we obtain a fixed-parameter tractability classification of SAT restricted to MHF's $M = P \wedge H$ where P has a fixed number k of different variables, provided by the polynomial bound $O(\|M\| 2^{0.5284k})$, where $\|M\|$ is the length of M.

We also analyse the connection of MHF-SAT to unrestricted SAT. Specifically we show that each CNF formula C with n different variables can be transformed in polynomial time into a MHF $M = P \wedge H$, such that P has $k \leq 2n$ different variables. Then C is satisfiable if and only if M is satisfiable, and the question, whether $M \in$ SAT, can be answered in time $O(\|C\| 2^{k/2})$. Hence, if there is an $\alpha < \frac{1}{2}$ such that every MHF $M = P \wedge H$ can be solved in time $O(\|C\| 2^{\alpha k})$, then there is $\beta \leq 2\alpha < 1$ such that SAT for an arbitrary CNF-formula C can be decided in time $O(\|C\| 2^{\beta n})$. The MHF-formulation of a CNF-formula C yields a partition of all variables in C into the *essential* variables (variables occuring in P) and the remaining ones.

We finally show that the NP-hard optimization problem minimum weight SAT for mixed Horn formulas can be solved in time $O(2^{0.5284n})$ if non-negative weights are assigned to the variables.

Isomorphic Implication[1]

Michael Bauland [2]

Theoretische Informatik,
Universität Hannover,
Appelstr. 4, 30167 Hannover, Germany,
bauland@thi.uni-hannover.de.

Edith Hemaspaandra [3]

Department of Computer Science,
Rochester Institute of Technology,
Rochester, NY 14623, U.S.A.,
eh@cs.rit.edu.

Wir untersuchen das Problem der isomorphen Implikation für Boolesche Constraints. Wir zeigen, dass dieses ein natürliches Analog zum Subgraph-Isomorphieproblem ist. Wir beweisen, dass, abhängig von der Menge der Constraints, das Problem in P, NP-vollständig oder NP-hart, coNP-hart und in $\mathrm{P}_{||}^{\mathrm{NP}}$ ist. Wir zeigen, wie man für einige Fälle die NP- und coNP-Härte zu einer $\mathrm{P}_{||}^{\mathrm{NP}}$-Härte erweitern kann und vermuten, dass dies in allen Fällen gemacht werden kann.

[1] Teilweise unterstüzt durch NSF-CCR-0311021 and DFG VO 630/5-1.

[2] Die Arbeit wurde teilweise während eines Besuchs des Laboratory for Applied Computing am Rochester Institute of Technology erstellt.

[3] Die Arbeit wurde teilweise während eines Sabbaticals an der University of Rochester erstellt.

Bounded Degree Closest k-Tree Power is NP-Complete

Michael Dom, Jiong Guo, Rolf Niedermeier

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
Institut für Informatik, 07740 Jena

An undirected graph $G = (V,E)$ is the k-power of an undirected tree $T = (V,E')$ if $(u,v) \in E$ iff u and v are connected by a path of length at most k in T. Tree T is called the tree root. Tree powers can be recognized in polynomial time. The thus naturally arising question is whether a graph G can be modified by adding or deleting a specified number of edges such that G becomes a tree power. This problem becomes NP-complete for $k \geq 2$. Strengthening this result, we answer the main open question of Tsukiji and Chen [COCOON 2004] by showing that the problem remains NP-complete when additionally demanding that the tree roots must have bounded degree.

Algorithm Engineering for Optimal Graph Bipartization

Falk Hüffner[1]

Institut für Informatik,
Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2,
07743 Jena,
`hueffner@minet.uni-jena.de`.

We examine exact algorithms for the NP-complete GRAPH BIPARTIZATION problem, also known as MAXIMUM BIPARTITE SUBGRAPH or ODD CYCLE TRANSVERSAL.

> GRAPH BIPARTIZATION
>
> **Input:** An undirected graph $G = (V, E)$ and a nonnegative integer k.
>
> **Task:** Find a subset $C \subseteq V$ of vertices with $|C| \leq k$ such that each odd cycle in G contains at least one vertex from C, that is, the removal of all vertices in C from G results in a bipartite graph.

This problem is NP-complete and MaxSNP-hard; the best known approximation is by a factor of $\log |V|$. It has numerous applications, for example in VLSI design, computational biology, and register allocation.

In a recent breakthrough paper, Reed, Smith, and Vetta proved that the GRAPH BIPARTIZATION problem on a graph with n vertices and m edges is solvable in $O(4^k \cdot kmn)$ time, where k is the number of vertices to delete. The basic idea is to construct size-k solutions from already known size-$(k+1)$ solutions, the so-called *iterative compression*.

In this work we present new algorithms based on iterative compression and demonstrate that they are a worthwhile alternative for solving GRAPH BIPARTIZATION in practice. The worst-case time complexity is improved to $O(3^k \cdot mn)$. We present experimental results with real-world data, synthetic application data, and random graphs. The implementation can solve all problems from a testbed from computational biology within minutes, whereas established methods based on Integer Linear Programming are only able to solve about half of the problems within reasonable time. A heuristic yielding optimal solutions performs very well on dense graphs. These results makes the practical evaluation of iterative compression for other applications appealing.

[1] Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

Toleranzbasierende Algorithmen für das Travelling Salesman Problem

Gerold Jäger

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
jaegerg@informatik.uni-halle.de

Das Travelling Salesman Problem (TSP) ist das Problem, in einem gewichteten, vollständigen Graphen einen kürzesten, geschlossenen Weg zu finden, der jeden Knoten genau einmal durchläuft. TSP ist ein NP-schwieriges Problem. Schon für kleine n ist die triviale Methode, alle Wege durchzuprobieren ($(n-1)!/2$ Schritte) zu aufwendig.

Wegen seiner einfachen Formulierung ist TSP ein Problem, das dazu benutzt wird, verschiedene Methoden miteinander zu vergleichen. Dies wird insbesondere mit Hilfe einer im Internet veröffentlichen Datenbank (TSPLIB) ermöglicht. Diese Datenbank enthält TSP-Instanzen von 14 Knoten bis zu 1.904.711 Knoten. Den meisten Problemen liegen reale Städteprobleme zugrunde. Die Probleme geben sehr gut die historische Entwicklung des TSP wieder.

Der erste Durchbruch gelang Dantzig, Fulkerson, Johnson, die 1959 ein Problem mit 49 Städten lösen konnten. Deren Methode bildet immer noch die Grundlage für den zur Zeit besten Algorithmus von Applegate, Bixby, Chvátal, Cook, der 2004 ein Problem mit 24978 Städten in Schweden gelöst hat.

Obwohl es wenig Approximationsschranken für das TSP gibt, gibt es viele gute Heuristiken, die die optimale Lösung ausreichend approximieren. Die derzeit beste Heuristik stammt von Helsgaun aus dem Jahre 1998. Sein Algorithmus kann die meisten der TSP-Instanzen aus TSPLIB lösen, inklusive der Schweden-Tour, ohne allerdings deren Optimalität zu nachzuweisen.

Im ersten Teil des Vortrags geben wir einen Abriß über die Entwicklung des TSP und die derzeit führenden Methoden. Im zweiten Teil stellen wir eine neue Idee für das TSP vor, die auf dem Toleranzbegriff basiert.

Zur Lösung eines TSP muß man sich mit der Frage auseinandersetzen, welche Kanten mit großer Wahrscheinlichkeit in einer optimalen Tour auftauchen und welche nicht. Es ist klar, daß Kanten mit einem großen Gewicht eher unwahrscheinlich sind. Somit ist das Gewicht auf den ersten Blick ein gutes Kriterium. Bedenkt man aber, daß es Knoten gibt, die nur mit schweren Kanten verbunden sind, so relativiert sich dieses Kriterium.

Die *obere Toleranz* einer Kante der optimalen Tour gibt an, wieviel man das Gewicht einer Kante erhöhen muß, damit sie nicht mehr in einer optimalen Tour autaucht. Analog gibt die *untere Toleranz* einer Kante außerhalb einer optimalen Tour an, wieviel man das Gewicht verkleinern muß, damit sie in einer optimalen Tour liegt.

Dieser Begriff ist sicherlich ein sehr gutes Kriterium, allerdings ist die Berechnung der Toleranz einer Kante mindestens so schwer wie die Lösung des gesamten TSP. Deswegen betrachten wir einfacher zu berechende Toleranzen, z.B. von TSP-Heuristiken oder zum TSP verwandten Polynomialzeit-Problemen wie Minimal Spannender Baum, Lineares Assignment-Problem.

Wir stellen verschiedene solcher Toleranzbegriffe vor und zeigen, wie man daraus effiziente TSP-Algorithmen entwickeln kann.

Verhältnis von AvgP und HP zu Error-Klassen

Arfst Nickelsen, Birgit Schelm

TU Berlin, Universität Hannover
{nicke,bts}@cs.tu-berlin.de

Bei der Average-Case-Komplexität betrachtet man die durchschnittliche Laufzeit von Algorithmen (durchschnittlich bezüglich einer Wahrscheinlichkeitsverteilung μ auf den Eingabeinstanzen). Es gibt verschiedene Arten, Polynomialzeit-Average-Case-Klassen zu definieren:

- Eine Sprache A mit Verteilung μ ist in **AvgP** [Levin], falls eine deterministische Turingmaschine M existiert, die A entscheidet und deren Laufzeit im μ-Durchschnitt polynomiell ist.
- Eine Sprache A mit Verteilung μ ist in **HP** [Impagliazzo], wenn eine polynomiell zeitbeschränkte deterministische Turingmaschine M existiert, die bei Eingabe $(x, 1^m)$ einen Wert $\in \{0, 1, ?\}$ ausgibt und folgende Fehlerschranke einhält: $\mathrm{Prob}_\mu \left[M(x, 1^m) \neq \chi_A(x) \right] < 1/m$
- (A, μ) ist in der Error-Klasse $F(n)$**-ErrP** [Jakoby,Schindelhauer], falls ein $S \in \mathrm{P}$ existiert, so dass $\mathrm{Prob}_{\mu_n} \left[x \in A \Delta S \right] \leq F(n)$ für alle n.
  Kann man die Fehlerschranke $F(n)$ unter alle Kehrwerte von Polynomen drücken, so erhält man die Klasse **DistNearlyP** $:= \bigcap_{k \geq 1} \frac{1}{n^k}$-ErrP.

AvgP ist echt enthalten in HP. Wie verhalten sich diese zwei Klassen zu DistNearlyP? Verteilungen μ, für die ein Polynom n^c existiert, so dass $\mu(n) \geq \frac{1}{n^c}$, nennen wir *fair*. Wir können zeigen:

Satz 1: Falls $(A, \mu) \in \mathrm{HP}$ und μ ist fair, so gilt $(A, \mu) \in \mathrm{DistNearlyP}$.

Bei **Satz 1** lässt sich weder die Bedingung an $\mu(n)$ wesentlich lockern, noch die Fehlerschranke $F(n)$ wesentlich verbessern:

Satz 2: Sei $\mu(x) = \frac{1}{g(|x|)} \frac{1}{2^{|x|}}$, wobei g superpolynomiell. Dann gibt es ein A, so dass $(A, \mu) \in \mathrm{HP}$, aber für alle $S \in \mathrm{P}$ existiert ein $n \in \mathbb{N}^+$ mit

$$\mathrm{Prob}_{\mu_n} \left[x \in A \Delta S \right] = 1 .$$

Satz 3: Es gibt ein $(A, \mu) \in \mathrm{AvgP}$ mit μ fair, so dass für alle super-polynomiellen Funktionen $g(n)$ gilt:

$$(A, \mu) \notin \frac{1}{g(n)}\text{-ErrP} .$$

Dynamic Complexity Theory Revisited[1]

Volker Weber, Thomas Schwentick

Philipps-Universität Marburg, FB Mathematik und Informatik
{webervo,tick}@informatik.uni-marburg.de

Abstract. Dynamic complexity asks for the effort needed to maintain the information about properties of a structure under operations changing the structure. This talk introduces a refined notion of dynamic problems which takes the initial structure into account. It develops the basic structural complexity notions accordingly. It also shows that the dynamic version of the LOGCFL-complete problem D_2LREACH(acyclic) can be maintained with first-order updates.

For a set S, the *static decision problem* asks whether a given input I is an element of S. Classical *static complexity theory* studies the inherent computational effort to answer this question. But often one is not interested only once whether $I \in S$ but I undergoes small changes and information about its membership in S should be available after each change. E.g., S could contain all triples (G,s,t), where G is a graph and s and t are nodes such that t is reachable from s in G. Changes could be insertion and deletion of edges. Another typical example is a view in a database with tuple insertions and deletions to the base relations.

Of course, in many cases one can expect that if I' results from I by applying a small change, whether $I' \in S$ might be closely related to whether $I \in S$. In particular, it should often be simpler to *maintain* information about membership in S under small changes than to recompute it from scratch for each new instance. This might involve auxiliary data structures which are updated accordingly.

These considerations are the starting point of *dynamic complexity theory*. This theory has been focusing on two lines of research, structural results about complexity classes of dynamic problems, including suitable reduction concepts and complete problems and upper bounds for concrete problems. Both lines consider problems with a very low update complexity, especially with updates expressible in first-order logic (or, equivalently, by uniform families of constant-depth, polynomial size circuits with unbounded fan-in, aka AC^0).

This talk contributes to both lines described above. First, by taking the complexity of the initial instance I into account in a very simple fashion, we define more refined complexity classes and corresponding reduction concepts. More precisely, our classes are of the form Dyn($\mathscr{C}$,$\mathscr{C}'$), where $\mathscr{C}$ is the complexity of computing the auxiliary data structure for the initial input I and $\mathscr{C}'$ is the complexity of the updates. We show that these classes and reductions behave nicely and that the results of previous work translate in a straightforward way. The new classes allow a more precise classification of problems. We show that most of the problems mentioned above are in the respective class Dyn($\mathscr{C}$,FO), where $\mathscr{C}$ is the complexity of the underlying static problem. Nevertheless, optimality w.r.t. the initial input complexity is not automatic, e.g., it is not clear whether the dynamic reachability problem is in Dyn(NL,TC^0).

The technically most difficult result of this talk contributes to the other line of research. It presents a (non-redundant) LOGCFL-complete problem with first-order updates, more precisely, in Dyn(LOGCFL,FO).

[1] This work was presented at STACS 05. An extended abstract is published in Proceedings of STACS 2005, volume 3404 of LNCS. Springer, 2005.