

Platzeffiziente Euler-Partitionierung und bipartite Kantenfärbung

MORITZ LAUDAHN

Universität Augsburg, Lehrstuhl für Theoretische Informatik
moritz.laudahn@informatik.uni-augsburg.de

Zusammenfassung: Der Vortrag behandelt platzeffiziente Algorithmen für zwei Probleme auf ungerichteten Multigraphen, nämlich Euler-Partitionierung (die Partitionierung der Kantenmenge in eine minimale Anzahl von Pfaden) und bipartite Kantenfärbung. Im Fall der Euler-Partitionierung gelingt es, den Speicherbedarf unter Beibehaltung der linearen Laufzeit um einen logarithmischen Faktor zu reduzieren. Für die bipartite Kantenfärbung erreichen wir eine Laufzeit von $O(n + m \min\{\Delta, \log m\})$ unter Nutzung von $O(n + m)$ Bits an Arbeitsspeicher, wobei n , m und Δ die Anzahl der Knoten, die Anzahl der Kanten sowie den maximalen Grad des Multigraphen bezeichnen. Der Vortrag beruht auf gemeinsamer Arbeit mit Torben Hagerup und Frank Kammer.

Edge Hop

MORITZ GOBBERT

Universität Trier, Abteilung Informatikwissenschaften
s4mogobb@uni-trier.de

Zusammenfassung: In meinem Vortrag werde ich ein Spiel namens *Edge Hop* vorstellen, bei dem es darum geht einen markierten Spielstein auf einem Graphen von einem Startknoten zu einem Zielknoten zu bewegen. Hierbei hat der Graph bestimmte Eigenschaften. Beispielsweise befinden sich auf ihm zusätzliche Spielsteine und jeder Knoten kann nur eine bestimmte Anzahl an Steinen aufnehmen. Weiter werde ich zeigen, dass *EDGEHOP*, also die Frage, ob der markierte Spielstein den Zielknoten erreichen kann, *NP-vollständig* ist. Für die Reduktion werde ich verschiedene Gadgets konstruieren, die ein Gerüst für die Komplexitätsanalyse anderer (kombinatorischer) Spiele bzw. ähnlicher Fragestellungen bieten.

Exploration von Temporalen Graphen

F. KAMMER

Universität Augsburg, Lehrstuhl für Theoretische Informatik
kammer@informatik.uni-augsburg.de

Zusammenfassung: Ein temporaler Graph ist ein Graph, dessen Kantenmenge sich in jedem Zeitschritt verändern kann. Eine Exploration in einem Graphen G ist ein Pfad, der alle Knoten von G mindestens einmal enthält. Die Suche nach einer Exploration in einem temporalen Graphen ist die Suche nach einem Zeitplan, der für jeden Zeitschritt vorschreibt, ob man am aktuellen Knoten verbleibt oder ihn über eine momentan vorhandene Kante verläßt. *TEXP* ist das Problem, einen solchen Zeitplan zu finden, der möglichst wenige Zeitschritte hat. Der Vortrag zeigt, dass das Finden einer $O(n^{1-\epsilon})$ -approximativen Lösung für *TEXP* NP-hart ist, und skizziert Ideen für spezielle temporale Graphen. Die Ergebnisse sind gemeinsame Arbeit mit T. Erlebach und M. Hoffmann.

Clustering mit unteren Schranken

K. CASEL

Universität Trier, Abteilung Informatikwissenschaften

casel@uni-trier.de

Zusammenfassung: Klassische Clustering-Probleme wie k -Center oder k -Means betrachten n Elemente eines metrischen Raumes und suchen nach einer optimalen Partition dieser in exakt k nichtleere Teilmengen. Optimalität in Bezug auf die gegebene Metrik kann dabei auf unterschiedliche Art definiert werden.

Anders verhält sich die Suche nach einer optimalen Partition, bei der jede Teilmenge mindestens r Elemente enthält *ohne* die Anzahl der Teilmengen k zu fixieren. Auf diese Art lässt sich z.B. eine Variante von Facility Location modellieren, bei der eine Filiale nur eröffnet wird, wenn sie mindestens r Kunden bedient. Wir diskutieren die Eigenschaften dieses Partitionsproblems für unterschiedliche Optimalitätskriterien insbesondere für kleine Werte von k .

On Finding Points in General Position

A. NICHTERLEIN

TU Berlin, Institut für Softwaretechnik und Theoretische Informatik

andre.nichterlein@tu-berlin.de

Zusammenfassung: We study computational aspects of the General Position Subset Selection problem defined as follows: Given a set of points in the plane, find a maximum-cardinality subset of points in general position. We prove that General Position Subset Selection is NP-hard, APX-hard, and give several fixed-parameter tractability results.

This is joint work with Vincent Froese (TU Berlin), Iyad Kanj (DePaul University, Chicago), and Rolf Niedermeier (TU Berlin).

See <http://arxiv.org/abs/1508.01097> for the the arXiv version.

Pseudoarboricity or Lowest Maximum Indegree Orientations

M. BLUMENSTOCK

Johannes Gutenberg-Universität Mainz, Institut für Informatik

mablumen@uni-mainz.de

Zusammenfassung: The densest subgraph problem, which asks for the subgraph with the highest edge-to-vertex ratio d^* , is a problem solvable in polynomial time. We discuss algorithms for this problem and the computation of an orientation of the graph with the lowest maximum indegree, which is equal to $\lceil d^* \rceil$. This value also equals the pseudoarboricity of the graph. We prove that it can be computed faster than previously known if $|E| \in \mathcal{O}(|V|^{4/3})$ or $d^* \in \mathcal{O}(|V|^{2/3})$. Our analyses include a generalization of unit capacity flow networks to flow networks where only source and sink arcs have non-unit capacities. We also compare several algorithms experimentally.

A New Parameterisation for Consensus Strings Problems

M. L. SCHMID

Universität Trier, Abteilung Informatikwissenschaften

mschmid@uni-trier.de

Zusammenfassung: The parameterised complexity of the CLOSEST SUBSTRING PROBLEM and the CONSENSUS PATTERNS PROBLEM with respect to the parameter $(\ell - m)$ is investigated, where ℓ is the maximum length of the input strings and m is the length of the solution string. We present an exact exponential time algorithm for both problems, which is based on an alphabet reduction. Furthermore, it is shown that for most combinations of $(\ell - m)$ and one of the classical parameters (m, ℓ , number of input strings k , distance d), we obtain fixed-parameter tractability, but even for constant $(\ell - m)$ and constant alphabet size, both problems are NP-hard.

Antimaterie in Komplexitätsklassen von Membransystemen

R. FREUND

TU Wien, Institut für Computersprachen

rudi@emcc.de

Zusammenfassung: Bestimmte Klassen von Membransystemen beschreiben wohlbekannte Komplexitätsklassen. Im Speziellen betrachten wir P-Systeme mit aktiven Membranen, welche nur die Teilung von elementaren Membranen erlauben und die polynomielle Komplexitätsklasse **P** charakterisieren. Erweitert man dieses Modell jedoch mit Materie-Antimaterie-Annihilationsregeln, so erhält man zumindest **NP** - wir zeigen wie **SAT** von einer Familie von P-Systemen mit aktiven Membranen und der Teilung von elementaren Membranen sowie Materie-Antimaterie-Annihilationsregeln gelöst werden kann.

Der GapSet Algorithmus

A. JAAX

Universität Trier, Abteilung Informatikwissenschaften

jaax@uni-trier.de

Zusammenfassung: Der GapSet Algorithmus ist ein neuer Algorithmus zur Berechnung minimaler Schnitte in Graphen. Er stellt eine Verallgemeinerung des Algorithmus von Hao und Orlin dar und besitzt ebenfalls die Komplexität $\mathcal{O}(n^2\sqrt{m})$. Der flussbasierte Minimum Cut Algorithmus baut auf drei Charakteristika auf: der Informations-Wiederverwendung, der Partitionierung der Knotenmenge in GapSets und der Technik Flussverdopplung. Die neuen Erkenntnisse der Entstehung von GapSets und deren Eigenschaften, sowie die Technik Flussverdopplung bieten ein besseres Verständnis dieser Algorithmen und die Möglichkeit einer einfachen Parallelisierung.

A Simple and Fast Linear-Time Algorithm for Proportional Apportionment

R. REITZIG

TU Kaiserslautern, Fachbereich Informatik

reitzig@cs.uni-kl.de

Zusammenfassung: Cheng and Eppstein describe a linear-time algorithm for computing highest-average allocations in proportional apportionment scenarios, for instance assigning seats to parties in parliament so that the distribution of seats resembles the vote tally as well as possible.

We propose another linear-time algorithm that consists of only one call to a rank selection algorithm after elementary preprocessing. Our algorithm is conceptually simpler and faster in practice than the one by Cheng and Eppstein.

News on Quicksort

S. WILD

TU Kaiserslautern, Fachbereich Informatik

wild@cs.uni-kl.de

Abstract: One might think that the solutions to the most basic problems in computer science are settled by now; at least, we know how to sort efficiently, don't we? Fact: Within the last decade, *all* sorting algorithms in the Java runtime library have been entirely rewritten! I will introduce the new dual-pivot Quicksort that is nowadays used to sort arrays of primitive types in Java, and present some theoretical analyses of this algorithm that offer a possible and, in my opinion, plausible explanation why (a) dual-pivot Quicksort is faster than the previously used (classic) Quicksort and (b) why this improvement was not already found much earlier.
