Seminar-Themen

„Algorithmen und Komplexität“ und „Komplexität“

Wintersemester 2018/19 — Prof. Dr. Georg Schnitger

Im Folgenden finden Sie Vortragsthemen für das Bachelor-Seminar „Algorithmen und Komplexität“ und das Master-Seminar „Komplexität“ im Wintersemester 18/19. Die Vergabe der Themen erfolgt in der Vorbesprechung. Bachelor-Studenten dürfen jedes Thema wählen, das mit einem B markiert ist. Für Master-Studenten kommen nur Themen in Frage, die mit M gekennzeichnet sind. Sind Themen sowohl mit B als auch mit M markiert, stehen sie beiden Studiengängen offen, wobei der geforderte Umfang für Master-Studenten entsprechend größer ist.

Bitte machen Sie sich vorab mit den Themen vertraut, sodass Sie in der Vorbesprechung bereits wissen, welche Themen für Sie in Frage kommen! Wenn Sie ein eigenes Thema vorschlagen wollen, setzen Sie sich bitte rechtzeitig vor der Vorbesprechung mit uns in Verbindung.

Inhaltsverzeichnis

Algorithmen

① The Hungarian Method for the Assignment Problem — B

Katta G. Murty
Network Programming (Internet Edition), Kapitel 3.1. http://www-personal.umich.edu/~murty/books/network_programming/

Kurzbeschreibung: Das *Assignment Problem* ist die Frage nach einem schwerstmöglichen Matching in einem gewichteten bipartiten Graphen. Die sogenannte „Ungarische Methode“ (Kuhn-Munkres-Algorithmus 1955, 1957) bildet die Grundlage für primal-duale Algorithmen zur Lösung von ganzzahligen linearen Programmen (ILPs).

Nützliche Vorkenntnisse: Lineare Programmierung, Dualität

② Matching Is As Easy As Matrix Inversion — B

Ketan Mulmuley, Umesh V. Vazirani, Vijay V. Vazirani
Combinatorica 7(1): 105-113 (1987). http://dblp.org/rec/html/journals/combinatorica/MulmuleyVV87

Kurzbeschreibung: Mithilfe eines Isolation-Lemmas wird ein randomisierter, parallelisierbarer Algorithmus zur Berechnung eines perfekten Matchings vorgestellt. Der „schwierigste“ Schritt des Algorithmus ist das Invertieren einer ganzzahligen Matrix.

Nützliche Vorkenntnisse: Lineare Algebra, elementare Stochastik, parallele Algorithmen

③ Auction algorithms for network flow problems: A tutorial introduction — B

Dimitri P. Bertsekas
Comp. Opt. and Appl. 1(1): 7-66 (1992). https://dblp.org/rec/journals/coap/Bertsekas92

Kurzbeschreibung: Auktionsalgorithmen sind für viele Graphprobleme wie kürzeste Wege oder Max-Flow bekannt. Der „Klassiker“ ist der Auktionsalgorithmus für gewichtetes bipartites Matching. Dabei wird die eine Seite der Knoten zu *Bietern*, die andere Seite zu *Gegenständen* mit anfänglichem Preis 0. Jeder Bieter wählt nun den Gegenstand mit dem für ihn besten Profit (Kantengewicht minus Preis). In mehreren Runden passt der Algorithmus die Preise an, bis keine zwei Bieter mehr denselben Gegenstand wählen und somit ein Matching gefunden ist.

Bemerkungen: Nur die Abschnitte 1 bis 3 müssen bearbeitet werden.

Nützliche Vorkenntnisse: Lineare Programmierung, Dualität

④ The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals — B, M

Jon Feldman, Matthias Ruhl
SIAM J. Comput. 36(2): 543-561 (2006). https://dblp.org/rec/journals/siamcomp/FeldmanR06

Kurzbeschreibung: Im Directed Steiner Network Problem sind ein gerichteter Graph mit Kantengewichten, sowie p Quellen $s_1, \ldots, s_p$ und p Senken $t_1, \ldots, t_p$ gegeben. Gefunden werden soll ein leichtester gerichteter Wald (d. h. eine leichteste Vereinigung von gerichteten Pfaden) sodass es von jeder Quelle s_i einen Weg zur Senke t_i gibt. Während das Problem für allgemeine p als NP-vollständig nachgewiesen wurde, lässt es sich für beschränkte p in Polynomialzeit lösen.

⑤ SAT-Algorithmen: Schöning-Algorithmus — B

Uwe Schöning, Jacobo Torán
Das Erfüllbarkeitsproblem SAT – Algorithmen und Analysen. Mathematik für Anwendungen 1, Lehmann 2012. http://dblp.org/rec/books/daglib/0028796

Kurzbeschreibung: Im SAT-Problem ist eine KNF-Formel gegeben und auf Erfüllbarkeit zu prüfen. Da das Problem NP-vollständig ist, sind hier keine Wunder zu erwarten, aber man möchte die exponentielle Laufzeit so niedrig wie möglich drücken. Der randomisierte Schöning-WalkSAT-Algorithmus durchsucht mittels einer Markov-Kette den Belegungsraum und erreicht die Laufzeit $\mathcal{O}\big(\mathrm{poly}(n) \cdot (\frac{4}{3})^n\big)$ für 3-SAT.

Nützliche Vorkenntnisse: Elementare Stochastik, Markov-Ketten

⑥ SAT-Algorithmen: Moser-Scheder-Algorithmus — B, M

Uwe Schöning, Jacobo Torán
Das Erfüllbarkeitsproblem SAT – Algorithmen und Analysen. Mathematik für Anwendungen 1, Lehmann 2012. http://dblp.org/rec/books/daglib/0028796

Kurzbeschreibung: Der Moser-Scheder-Algorithmus erreicht dieselbe Laufzeit wie der oben genannte Schöning-Algorithmus sogar deterministisch mithilfe von Überdeckungscodes und lokaler Suche.

Bemerkungen: Das Thema baut auf dem obigen auf.

⑦ Time Bounds for Selection und die Akra-Bazzi-Methode — B

Bemerkung: Sowohl der Algorithmus als auch die Akra-Bazzi-Methode sollen bearbeitet werden.

Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, Robert E. Tarjan: J. Comput. Syst. Sci. 7(4): 448-461 (1973). https://dblp.org/rec/journals/jcss/BlumFPRT73

Kurzbeschreibung: Aus n gegebenen Zahlen soll die k-größte gefunden werden. Wie lässt sich das schnell bewerkstelligen, ohne die Eingabe vollständig zu sortieren? Ein einfacher deterministischer Linearzeit-Algorithmus wird präsentiert.

Mohamad A. Akra, Louay Bazzi: On the Solution of Linear Recurrence Equations. Comp. Opt. and Appl. 10(2): 195-210 (1998). https://dblp.org/rec/journals/coap/AkraB98

Kurzbeschreibung: Die Laufzeit des obigen Algorithmus kann mit der Akra-Bazzi-Methode analysiert werden – ein Mastertheorem für Rekursionsgleichungen der Form $T(n) = T(an) + T(bn) + f(n)$.

⑧ A Universal Algorithm for Sequential Data Compression — B, M

Jacob Ziv, Abraham Lempel
IEEE Trans. Information Theory 23(3): 337-343 (1977). https://dblp.org/rec/journals/tit/ZivL77

Kurzbeschreibung: Der Lempel-Ziv-Algorithmus (LZ77 bzw. LZ1) steht am Anfang einer Reihe von technischen Weiterentwicklungen bei der Kompression von sequentiellen Datenströmen. Er bildet die Basis für Grafikformate wie GIF oder PNG und den Lempel-Ziv-Markov-Algorithmus (LZMA), der bei verschiedenen Linux-Distributionen genutzt wird.

Bemerkungen: Als Masterthema ist neben LZ77 auch LZMA (siehe https://www.7-zip.org/sdk.html) zu bearbeiten. Insbesondere soll dargestellt werden, auf welchen theoretischen Grundlagen neuere Varianten des Lempel-Ziv-Algorithmus fußen.

Nützliche Vorkenntnisse: String-Algorithmen, Lesen von technischen Spezifikationen und Quellcode

⑨ Fast Distributed PageRank Computation — B

Atish Das Sarma, Anisur Rahaman Molla, Gopal Pandurangan, Eli Upfal
Theor. Comput. Sci. 561: 113-121 (2015). https://dblp.org/rec/journals/tcs/SarmaMPU15

Kurzbeschreibung: In dieser Arbeit werden parallele Algorithmen zur effizienten Bestimmung des PageRanks mittels Random Walks vorgestellt, bei denen die Prozessoren jeweils nur eine kleine Anzahl von Bits austauschen. Die Skalierbarkeit der Algorithmen ist daher gewährleistet.

Nützliche Vorkenntnisse: Elementare Stochastik (insb. Markov-Ketten), parallele Algorithmen

⑩ The Small-World Phenomenon: An Algorithmic Perspective — B

Jon Kleinberg
STOC, 2000. https://dl.acm.org/citation.cfm?id=335325

Kurzbeschreibung: In sozialen Netzwerken tritt das sogenannte Kleine-Welt-Phänomen auf: Je zwei Personen sind über eine sehr kurze Kette von Kontakten miteinander verbunden. Diese Arbeit beschäftigt sich mit den algorithmischen Grundlagen von Milgrams Experiment „übermittle einen Brief an eine Zielperson, von der du nur den Wohnort kennst, indem du ihn an einen deiner Bekannten weitergibst, von dem du vermutest, dass er der Zielperson näher steht als du“.

Nützliche Vorkenntnisse: parallele Algorithmen

⑪ **Low-Density Parity-Check Codes** B, M

David J.C. MacKay
Information Theory, Inference, and Learning Algorithms, Kapitel 47. http://www.inference.phy.cam.ac.uk/itprnn/book.pdf

Kurzbeschreibung: LDPC-Codes sind fehlerkorrigierende lineare Blockcodes, die z. B. für WLAN, digitales Fernsehen und für andere Datenübertragungstechniken eingesetzt werden. Die Dekodierung von Codewörtern ist jedoch NP-vollständig. Als Dekodierungsheuristik wird der Belief-Propagation-Algorithmus verwendet. Neben Anwendungen im maschinellen Lernen und in der kombinatorischen Optimierung, ist die Dekodierung von LDPC-Codes damit eines der herausragenden Anwendungsbeispiele von Belief Propagation.

Nützliche Vorkenntnisse: Elementare Stochastik, lineare Blockcodes

Komplexitätstheorie

⑫ **A Personal View of Average-Case Complexity** M

Russell Impagliazzo
Structure in Complexity Theory Conference 1995: 134-147. http://dblp.org/rec/conf/coco/Impagliazzo95

Kurzbeschreibung: Wie sähe unser Leben aus, wenn Hypothesen aus der Komplexitätstheorie wie bspw. „P = NP“ gälte oder Einwegfunktionen existierten? Auf umgangssprachliche Weise werden fünf mögliche Welten beschrieben: Algorithmica, Heuristica, Pessiland, Minicrypt und Cryptomania.

Bemerkungen: Diese Arbeit ist als Ausgangspunkt für eine Formalisierung der beschriebenen Welten (Definitionen, Sätze, Beweisideen) anhand der dort zitierten Literatur aufzufassen.
Es können bis zu zwei Vorträge zu dieser Arbeit vergeben werden. Dabei sollte der eine im Themengebiet Average-Case-Complexity (die ersten drei Welten) und der andere im Themengebiet One-Way-Functions (die letzten drei Welten) angesiedelt werden.

Nützliche Vorkenntnisse: Grundkenntnisse in Kryptographie

⑬ **Interactive Proofs and Graph Isomorphism** B, M

Kapitel 8 bis inkl. Abschnitt 8.3 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Gegeben seien eine Sprache L und eine Eingabe x. In einem Interactive Proof (IP) müssen zwei Personen kooperieren, um die Frage $x \in L$? zu beantworten. Der *Prover* betrachtet die Eingabe, rechnet beliebig lange und schickt einen „Beweis“ an den *Verifier*. Dieser muss dann mithilfe des Beweises in Polynomialzeit entscheiden, ob $x \in L$ gilt. Ein prominentes Problem im Kontext der IPs ist Graph Isomorphism (GI). Es wird einerseits gezeigt, dass GI vermutlich nicht NP-vollständig ist – sonst würde die Polynomialzeithierarchie kollabieren –, andererseits ist auch kein Polynomialzeitalgorithmus für GI bekannt. Es wird somit vermutet, das GI „zwischen“ P und NP-vollständig liegt.

Nützliche Vorkenntnisse: Komplexitätsklassen wie P, NP und PSPACE, Elementare Stochastik

⑭ **Lower Bounds for Monotone Circuits** B, M

Abschnitt 14.3 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Ein monotoner Schaltkreis besitzt OR- und AND-Gatter, aber keine NOT-Gatter. Eine Technik zur Herleitung unterer Schranken für die Größe monotoner Schaltkreise für Entscheidungsprobleme wird vorgestellt. Beispielsweise benötigt das Clique-Problem monotone Schaltkreise mindestens exponentieller Größe. Könnte man die Einschränkung der Monotonie weglassen, so wäre $\mathsf{P} \neq \mathsf{NP}$ gezeigt, doch leider benötigen monotone Schaltkreise auch exponentielle Größe für die Lösung des Matching-Problems.

Nützliche Vorkenntnisse: Schaltkreise, Elementare Stochastik

⑮ Complexity of Counting and #P-Completeness — B

Kapitel 17 bis inkl. Abschnitt 17.3.1 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Im Gegensatz zu Entscheidungsproblemen, wo nach der *Existenz* einer Lösung gefragt ist, wird in Zählproblemen nach der *Anzahl* der Lösungen gefragt. Viele solcher Zählprobleme, beispielsweise das Zählen perfekter Matchings oder einfacher Kreise in einem Graphen, haben sich als sehr schwierig herausgestellt. Diese Schwierigkeit wird durch das Konzept der #P-Härte analog zur NP-Härte formalisiert.

Nützliche Vorkenntnisse: Komplexitätsklassen, Polynomialzeitreduktionen

⑯ Algebraic Computation Models — B, M

Kapitel 16 exkl. Abschnitt 16.2 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Hier werden Berechnungsmodelle untersucht, die nicht auf Bits, sondern auf komplexen Zahlen rechnen. Die Klassen $\mathsf{P}_{\mathbb{C}}$ und $\mathsf{NP}_{\mathbb{C}}$ werden analog zu P und NP eingeführt. Das Entscheidungsproblem des Hilbertschen Nullstellensatzes („Gegeben m Polynome $p_1, \ldots, p_m$ vom Grad d, besitzen diese eine gemeinsame Nullstelle?“) wird als $\mathsf{NP}_{\mathbb{C}}$-vollständig nachgewiesen – selbst wenn die Eingabe aus konventionellen Bits besteht.

Nützliche Vorkenntnisse: Schaltkreise, Komplexitätsklassen

⑰ On the Optimality of Bellman–Ford–Moore Shortest Path Algorithm — B, M

Stasys Jukna, Georg Schnitger
Theor. Comput. Sci. 628: 101-109 (2016). https://dblp.org/rec/journals/tcs/JuknaS16

Kurzbeschreibung: Der Bellman-Ford-Algorithmus findet mithilfe dynamischer Programmierung einen kürzesten Weg mit höchstens k Kanten in Zeit $\mathcal{O}(kn^2)$. Geht es schneller? Mithilfe des Modells der Switching Networks wird gezeigt, dass Bellman-Ford mehr oder weniger optimal ist.

Nützliche Vorkenntnisse: dynamische Programmierung

⑱ The Complexity of Escaping Labyrinths and Enchanted Forests — B

Florian D. Schwahn, Clemens Thielen
FUN 2018: 30:1-30:13. https://dblp.org/rec/conf/fun/SchwahnT18

Kurzbeschreibung: Die Komplexität der bekannten Brettspiele *Das verrückte Labyrinth* (engl. *The aMAZEing Labyrinth*) und *Sagaland* (engl. *Enchanted Forest*) wird analysiert. Das erste stellt sich als NP-vollständig heraus, sofern man alleine spielt, bzw. als PSPACE-vollständig, wenn man einen Gegenspieler hat. Das zweite hingegen ist in Polynomialzeit lösbar.

Nützliche Vorkenntnisse: Polynomialzeitreduktionen

⑲ Tetris is hard, even to approximate — B

Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogeboom, Walter A. Kosters, David Liben-Nowell
Int. J. Comput. Geometry Appl. 14(1-2): 41-68 (2004) https://dblp.org/rec/journals/ijcga/BreukelaarDHHKL04

Kurzbeschreibung: Der Videospiel-Klassiker Tetris wird hinsichtlich seiner Komplexität untersucht. Zentrale Fragestellungen wie „maximiere die Anzahl der vervollständigten Reihen“ stellen sich als NP-hart heraus.

Nützliche Vorkenntnisse: Polynomialzeitreduktionen

⑳ Smoothed Complexity and Pseudopolynomial-Time Algorithms — B

Tim Roughgarden
Skripte (Lecture 17 und 18) zur Vorlesung CS264 Fall'17 (Beyond Worst-Case Analysis). Stanford University. http://theory.stanford.edu/~tim/w17/w17.html

Kurzbeschreibung: Smoothed Complexity ist ein Hybrid zwischen Worst-Case Complexity und Average-Case Complexity und beantwortet die Frage „Wie hoch ist die erwartete Laufzeit eines Algorithmus, wenn Worst-Case-Eingaben zufällig verrauscht werden?“. Im Rahmen der Vorlesung wird ein zentrales Resultat von Beier und Vöcking präsentiert: Ein binäres Optimierungsproblem hat genau dann polynomielle Smoothed Complexity, wenn es durch einen Algorithmus mit pseudopolynomieller erwarteter Laufzeit gelöst werden kann.

Bemerkungen: Aus Lecture 17 sind nur die Abschnitte 1 und 2 zu bearbeiten.

Nützliche Vorkenntnisse: Laufzeitanalyse, elementare Stochastik, lineare Programmierung

㉑ Not being (super)thin or solid is hard: A study of grid Hamiltonicity — B

Esther M. Arkin, Sándor P. Fekete, Kamrul Islam, Henk Meijer, Joseph S.B. Mitchell, Valentin Polishchuk, David Rappaport, Henry Xiao
Computational Geometry: Theory and Applications, 42(6-7), 582-605. https://www.sciencedirect.com/science/article/pii/S092577210800117X

Kurzbeschreibung: Das Hamiltonkreis-Problem ist eines der klassischen NP-vollständigen Probleme. Sofern $\mathsf{P} \neq \mathsf{NP}$ gilt, ist es im Allgemeinen nicht effizient lösbar. Doch wie sehen möglichst „einfache“ Graphklassen aus, für die das Hamiltonkreis-Problem trotzdem noch NP-vollständig ist? Und welche Graphklassen sind einfach und erlauben eine effiziente Lösung des Hamiltonkreis-Problems? In dieser Arbeit betrachten wir Gitter mit drei-, vier- und sechseckigen Maschen.

Nützliche Vorkenntnisse: Polynomialzeitreduktionen

㉒ Extended Regular Expressions: Succinctness and Decidability — B, M

Dominik D. Freydenberger
28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011). http://drops.dagstuhl.de/opus/volltexte/2011/3039/

Kurzbeschreibung: In der Praxis verbreitete Implementierungen von regulären Ausdrücken erlauben die Verwendung von Variablen bzw. Rückreferenzen. Dieses zusätzliche Feature der sogenannten erweiterten regulären Ausdrücke hat jedoch einen Haken: Der Ausdruck $\alpha = \big((\mathtt{a}|\mathtt{b})^*\big)\%x \cdot x$ beschreibt die (nichtreguläre) Copy-Sprache $L(\alpha) = \{ww : w \in \{a,b\}^*\}$. In dieser Arbeit werden die folgenden Fragestellungen für erweiterte reguläre Ausdrücke untersucht: Minimierbarkeit, Beschreibungslänge im Vergleich zu klassischen regulären Ausdrücken, Unentscheidbarkeit des Universalitäts-, Äquivalenz-, Inklusions-, Regularitäts- und Koendlichkeitsproblems.

Nützliche Vorkenntnisse: Formale Sprachen, Berechenbarkeit, Entscheidbarkeit, Turing-Maschinen