

**Sammlung der Zusammenfassungen  
der Vorträge auf dem**

**23. Workshop  
über Komplexitätstheorie,  
Datenstrukturen und  
effiziente Algorithmen**

**am 31. Mai 1994**

**in Paderborn**

Organisation: Rolf Wanka

# Datenstrukturen und effiziente Algorithmen

Paderborn, 31. Mai 1994

## Programm

- ab 8:45 IMBISS
- 9:25 – 9:30 BEGRÜSSUNG
- 9:30 – 9:55 Richard Beigel (New Haven), Martin Kummer, Frank Stephan (Karlsruhe):  
*Approximierbare Mengen sind nicht btt-hart für NP*
- 9:55 – 10:20 Thomas Hofmeister, Hanno Lefmann (Dortmund):  
*Computing Sparse Approximations Deterministically*
- 10:20 – 10:45 Ralf Diekmann, Burkhard Monien, Robert Preis (Paderborn):  
*Effiziente Graph-Partitionierung mit Hilfe der HS-Heuristik*
- 10:45 – 11:00 PAUSE
- 11:00 – 11:25 Martin Middendorf (Karlsruhe):  
*Zur Approximierbarkeit verschiedener Super- und Subsequences*
- 11:25 – 11:50 Klaus Jansen (Trier), Sabine Öhring (Denton):  
*Approximation Algorithms for Time Constrained Scheduling*
- 11:50 – 12:15 Joan Boyar, Rolf Fagerberg, Kim Skak Larsen (Odense):  
*Chromatic Priority Queues*
- 12:15 – 13:30 MITTAGSPAUSE
- 13:30 – 13:55 Ralf Diekmann, Jörn Gehring, Reinhard Lüling, Burkhard Monien, Markus Nübel, Rolf Wanka (Paderborn):  
*Sortieren großer Datenmengen auf massiv parallelen Systemen*
- 13:55 – 14:20 Michael Kaufmann (Tübingen), Heiko Schröder (Newcastle), Jop F. Sibeyn (Saarbrücken):  
*Asymptotically Optimal and Practical Routing on the Reconfigurable Mesh*
- 14:20 – 14:45 Przemysława Kanarek, Krzysztof Lorys (Wrocław):  
*Shifts by Powers of 2 Are Easy for Switching Networks*
- 14:45 – 15:00 PAUSE
- 15:00 – 15:25 Beate Bollig, Martin Hühne, Stefan Pölt, Petr Savický (Dortmund):  
*Über den erwarteten Delay von Schaltkreisen für die Disjunktion*
- 15:25 – 15:50 Harry Hengster, Rolf Drechsler, Bernd Becker (Frankfurt/Main):  
*On the Application of Local Circuit Transformations with Special Emphasis on PDF Testability*
- 15:50 – 16:15 Petr Savický (Prag), Ingo Wegener (Dortmund):  
*Efficient Construction of Reduced  $\pi$ OBDDs from FBDDs and  $\pi'$ OBDDs*
- 16:15 – 16:30 PAUSE
- 16:30 – 16:55 Hans-Jörg Burtchick, Wolfgang Lindner (Berlin):  
*Turing-Reduzierbarkeit auf p-selektive Mengen*
- 16:55 – 17:20 Werner Stein (Kaiserslautern):  
*Exploring the Sources of Computational Inefficiency in Inductive Inference*
- 17:20 – 17:45 Peter L. Bartlett (Canberra), Paul Fischer, Klaus-Uwe Höffgen (Dortmund):  
*Lernen aus Random Walks*



# Approximierbare Mengen sind nicht btt-hart für NP

Richard Beigel

Department of Computer Science,  
P.O. Box 2158, Yale Station, New Haven, CT 06520-2158, USA  
e-mail: `beigel@cs.yale.edu`

Martin Kummer    Frank Stephan\*

Institut für Logik, Komplexität und Deduktionssysteme,  
Universität Karlsruhe, 76128 Karlsruhe, Germany  
e-mail: `{kummer,fstephan}@ira.uka.de`

Eine Menge  $A$  heißt approximierbar, wenn es ein  $b$  gibt und eine Funktion  $f \in \text{FP}$ , welche für die Eingabe  $(x_1, \dots, x_b)$  einen  $b$ -Bit-Vektor  $(y_1, \dots, y_b) \neq (A(x_1), \dots, A(x_b))$  berechnet. Beispiele für approximierbare Mengen sind  $p$ -selektive Mengen.

Ausgehend von der Frage, ob es btt-harte  $p$ -selektive Mengen gibt, haben die Autoren (gleichzeitig mit Agrawal, Arvind und Ogihara) gezeigt, daß es keine btt-harten approximierbaren Mengen für NP gibt, außer im Fall  $P = NP$ .

Der Beweis zeigt, daß jeder approximierbare  $d$ -selbstreduzierbare  $d$ -Zylinder in  $P$  ist. Daher sind auch GI (Graph-Isomorphismus-Problem) und GA (Graph-Automorphismus-Problem) in  $P$ , wenn sie approximierbar sind.

Bei den Versuchen, die Ergebnisse auf tt-harte Mengen zu übertragen, wurde nur ein Teilerfolg erzielt: Wenn  $P \neq NP$  und SAT auf eine approximierbare Menge tt-reduzierbar ist, dann ist die Anzahl der Anfragen mindestens  $n^c$  für ein  $c > 0$ .

Für bestimmte Verschärfungen des Begriffs der Approximierbarkeit wie etwa easily countable konnte gezeigt werden, daß es keine Turing-harten easily countable Mengen für NP gibt, außer im Fall  $P = NP$ .

---

\*Unterstützt von der Deutschen Forschungsgemeinschaft (DFG) mittels Grant Me 672/4-1

# Computing Sparse Approximations Deterministically

Thomas Hofmeister      Hanno Lefmann

Lehrstuhl Informatik II

Universität Dortmund, D-44221 Dortmund, Germany

e-mail: {hofmeist,lefmann}@ls2.informatik.uni-dortmund.de

It is known [A] that for every  $n \times m$ -matrix  $A$  with entries taken from the interval  $[0, 1]$  and for every probability vector  $\underline{p}$ , there is a sparse probability vector  $\underline{q}$  with only  $O(\ln n/\varepsilon^2)$  non-zero entries such that every component of the vector  $A \cdot \underline{q}$  differs from every component of  $A \cdot \underline{p}$  in absolute value by at most  $\varepsilon$ .

The existence of such a vector is proved by a probabilistic argument. It was stated in [A] as an open problem whether there is an efficient, i. e. polynomial-time, deterministic algorithm which actually constructs such a vector  $\underline{q}$ .

In this talk, we provide an algorithm which computes such a vector  $\underline{q}$  and which takes time polynomial in  $n$ ,  $m$ , and  $\varepsilon$ . The algorithm is based on the method of “pessimistic estimators”.

The approximation result was crucial in some applications to matrix games. In [LY], applications of the result to Complexity Theory are given.

## References

- [A] I. Althöfer, *On Sparse Approximations to Randomized Strategies and Convex Combinations*, Linear Algebra and its Applications 199, 339-355 (1994).
- [LY] R. J. Lipton and N. E. Young, *Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory*, to appear in: Proc. of the 26th Ann. ACM Symposium on Theory of Computing, 1994.

# Effiziente Graph-Partitionierung mit Hilfe der HS-Heuristik

Ralf Diekmann    Burkhard Monien    Robert Preis

Fachbereich Mathematik / Informatik  
Universität-GH Paderborn, D-33095 Paderborn  
e-mail: {diek,bm,robsy}@uni-paderborn.de

Eine bedeutende Aufgabe im Bereich der Parallelen Programmierung besteht in der Zuordnung großer Prozeß- oder Daten-Netzwerke zu einem realen Multiprozessoren-System. Das zugehörige Mapping-Problem ist  $\mathcal{NP}$ -vollständig.

Wegen der fortschreitenden Entwicklung von Verbindungsnetzwerken mit unabhängigen Routingsystemen ist es sinnvoll, das Mapping-Problem auf ein Partitionierungsproblem zu reduzieren. Hierbei wird der Prozeßgraph in mehrere gleichgroße Partitionen aufgeteilt. Die Anzahl der Partitionen entspricht den vorhandenen Prozessoren. Als Optimierungsfunktion gilt die Anzahl der zwischen den Partitionen verlaufenden Kanten, die möglichst gering gehalten werden soll. Selbst das Bisektionsproblem ist  $\mathcal{NP}$ -vollständig. Die meisten verwendeten Heuristiken führen eine allgemeine Partitionierung durch rekursive Bisektion aus.

Es existieren einige gut bekannte Bisektionsalgorithmen. Am meisten benutzt werden die Kernighan-Lin-Heuristik (KL) mit Verbesserungen von Fiduccia und Mattheyses, Spektrale Methoden, Simulated Annealing und auch einige geometrische Heuristiken wie z. B. die Inertial-Methode.

Der Mapping-Prozeß selber ist nur ein Preprocessing-Schritt für die eigentliche parallele Applikation. Deshalb muß er möglichst sicher, schnell und mit geringem Speicherbedarf ausgeführt werden.

Mit der **Helpful-Set** Heuristik (HS) führen wir eine neue Methode zur Verbesserung einer Graphpartitionierung ein, die effizienter als die meisten z. Z. benutzten Bisektions-Heuristiken im Hinblick auf die erzielte Lösung, den Zeitaufwand und den benötigten Speicher ist. Ähnlich wie der KL-Algorithmus basiert das Verfahren auf lokaler Suche, die auf eine globale initiale Bisektion aufsetzt. Hierfür können einfache Verfahren gewählt werden, die unter Mißachtung der Optimierungsfunktion sehr schnell eine balancierte initiale Bisektion erzeugen. Ausgehend von dieser wird in mehreren Runden versucht, die Anzahl der Schnittkanten schrittweise zu verringern. In jeder Runde werden gleichgroße Mengen in beiden Bisektionshälften gesucht, die durch einen Wechsel der Seiten eine Verbesserung bewirken. Das Hauptproblem besteht im Finden solcher Mengen.

Die Knoten im Graph werden hinsichtlich ihrer Anzahl an internen und externen Kanten (Kanten, die über den Schnitt reichen) klassifiziert. Mit Hilfe dieser Klassifikation werden  $k$ -hilfreiche Mengen gesucht, die beim Wechsel auf die andere Seite des Schnittes die Anzahl der Schnittkanten um  $k$  verringern. Um die Balance wieder herzustellen, muß nachfolgend in der anderen Partition nach einer Balancierungsmenge gesucht werden, die beim Wechsel auf die andere Seite die Anzahl der Schnittkanten um höchstens  $-k + 1$  erhöht. Falls der Algorithmus solche Mengen findet, werden die entsprechenden Knoten ausgetauscht,

die Partitionierung bleibt balanciert und die Anzahl der Schnittkanten hat sich in dieser Runde um mindestens eine verringert. Anderenfalls wird in der nächsten Runde durch Veränderung des Parameters  $k$  nach Mengen mit geringeren hilfreichen Werten gesucht. Die Idee zu dieser Heuristik basiert auf einer Beweistechnik, die dazu benutzt wurde, obere Schranken für die Bisektionsweite von 4-regulären Graphen zu zeigen.

Wir vergleichen unseren Algorithmus mit Ergebnissen der Spektralen Partitionierung, der Multilevel- und der Inertial-Methode (jeweils auch mit zusätzlicher Verbesserung durch die Kernighan-Lin-Methode), die wir mit der *Chaco Partitioning Library* ermittelt haben, und einer eigenen Implementation des KL-Algorithmus mit den Verbesserungen durch Fiduccia und Mattheyses. Als Testgraphen benutzen wir einige selbstgenerierte Netzwerke (z. B. DeBruijn-Netzwerke und randomisierte, reguläre Netzwerke), einige Graphen aus der *Harwell-Boeing Collection* und einige Finite-Elemente-Graphen aus unterschiedlichen praktischen Anwendungen.

Umfangreiche Messungen zeigen, daß die neue Heuristik eine wesentliche Verbesserung für einige Grapharten darstellt. Dazu gehören z. B. die DeBruijn-Graphen, bei denen die HS-Heuristik bessere Bisektionsergebnisse erzielt als alle anderen Methoden. Wie schon erwähnt, ist die Implementation der HS-Heuristik sehr effizient im Hinblick auf die benötigte Zeit und den benötigten Speicherbedarf. Der z. Z. größte von uns getestete Graph ist ein DeBruijn der Dimension 20 (DB(20)) mit 1M Knoten und 2M Kanten. Basierend auf randomisierten initialen Bisektionen benötigt unsere Heuristik für DB(20) durchschnittlich etwa 6 min. auf einer Sparc Station SS10/30 mit 96MB Speicher. Durch seine Effizienz ist die HS-Heuristik, im Gegensatz zu vielen anderen Heuristiken, besonders für sehr große Graphen geeignet.

# Zur Approximierbarkeit verschiedener Super- und Subsequences

Martin Middendorf

Institut für Angewandte Informatik und Formale Beschreibungsverfahren  
Universität Karlsruhe, D-76128 Karlsruhe  
e-mail: [mimi@aifb.uni-karlsruhe.de](mailto:mimi@aifb.uni-karlsruhe.de)

Optimierungsprobleme im Zusammenhang mit Supersequences und Subsequences kommen z. B. in der Scheduling-Theorie und der Molekularen Biologie vor. Wir untersuchen die Komplexität verschiedener solcher Probleme hinsichtlich optimaler oder näherungsweise optimaler Lösbarkeit.

Eine Supersequence eines Strings  $S$  ist ein String, welchen man durch Einfügen von Buchstaben in  $S$  erhält; eine Subsequence von  $S$  ist ein String, welchen man durch Streichen von Buchstaben aus  $S$  erhält. Eine Non-Supersequence (Non-Subsequence) von  $S$  ist ein String, der keine Supersequence (Subsequence) von  $S$  ist. Eine (Common) Supersequence einer Menge  $L$  von Strings ist eine Supersequence eines jeden Strings in  $L$ . (Common) Subsequences, (Common) Non-Supersequences und (Common) Non-Subsequences werden entsprechend definiert. Eine Supersequence  $S$  von  $L$  heißt minimal, wenn keine echte Subsequence von  $S$  eine Supersequence von  $L$  ist. Eine Subsequence  $S$  von  $L$  heißt maximal, wenn keine echte Supersequence von  $S$  eine Subsequence von  $L$  ist. Maximale Non-Supersequences und Minimale Non-Subsequences werden entsprechend definiert.

Shortest Common Supersequence (SCS), Longest Common Subsequence (LCS), Longest Common Non-Supersequence (LCNS) und Shortest Common Non-Subsequence (SCNS) sind Probleme, die bekanntermaßen bereits über einem Alphabet der Größe 2 NP-vollständig sind. Weiterhin ist bekannt, daß diese Probleme für Alphabete beliebiger Größe MAX SNP-hard sind. Noch ungelöst ist, ob diese Probleme auch über Alphabeten konstanter Größe MAX SNP-hard sind.

Bei allen vier Problemen (SCS, LCS, LCNS und SCNS) geht es darum, eine Sequence  $S$  zu finden, die konsistent im Hinblick auf nur eine gegebene Menge  $L$  von Strings ist (d. h.  $S$  soll z. B. eine Supersequence von  $L$  sein). In [1], [2] und [3] wurde auch das Problem betrachtet, eine Sequence  $S$  zu finden, die konsistent ist zu zwei gegebenen Mengen  $L_1$ ,  $L_2$  von Strings (d. h.  $S$  soll z. B. Supersequence von  $L_1$  und Non-Supersequence von  $L_2$  sein).

Wir zeigen eine Beziehung zwischen dem Problem, eine Sequence zu finden, die konsistent im Hinblick auf zwei gegebene Mengen von Strings ist, zu dem Problem, eine Sequence mit gegebener Buchstabenzusammensetzung zu finden, die konsistent ist im Hinblick auf nur eine gegebene Menge von Strings. Dies führt zu einem Ansatz, mit dem die Komplexität solcher Konsistenzprobleme wesentlich klarer als bisher charakterisiert werden kann. Desweiteren untersuchen wir die Approximierbarkeit von Problemen, die dual sind zu SCS, LCS, LCNS und SCNS. Wir zeigen u. a., daß Longest Minimal Common Supersequence, Shortest Maximal Common Subsequence und Shortest Maximal Common Non-Supersequence bereits über einem Alphabet der Größe 2 MAX SNP-hard sind.



## Literatur

- [1] T. Jiang, M. Li, On the complexity of learning strings and supersequences, *Theoret. Comput. Sci.* **119** (1993) 336-371.
- [2] M. Middendorf, The shortest common nonsubsequence Problem is NP-complete, *Theoret. Comput. Sci.* **108** (1993), 365-369.
- [3] L. Zhang, The Approximation of Longest Common Non-Supersequences and Shortest Common Non-Subsequences is Hard, submitted for publication.

# Approximation Algorithms for Time Constrained Scheduling

Klaus Jansen

FB IV – Mathematik und Informatik,  
Universität Trier, D-54286 Trier, Germany.  
e-mail: [jansen@dm3.Uni-Trier.DE](mailto:jansen@dm3.Uni-Trier.DE)

Sabine Öhring

Department of Computer Science,  
University of North Texas, Denton, TX 76203-3886, USA.  
e-mail: [oehring@ponder.csci.unt.edu](mailto:oehring@ponder.csci.unt.edu)

In this talk we consider the following time constrained scheduling problem. Given a set of jobs  $J$  with execution times  $e(j) \in (0, 1]$  and an undirected graph  $G = (J, E)$ , we consider the problem to find a schedule for the jobs such that adjacent jobs  $(j, j') \in E$  are assigned to different machines and that the total execution time for each machine is at most 1.

The goal is to find a minimum number of machines to execute all jobs under this time constraint. This scheduling problem is a natural generalization of the classical bin packing problem. We propose and analyse several approximation algorithms with constant absolute worst case ratio for graphs that can be colored in polynomial time.

# Chromatic Priority Queues

Joan Boyar    Rolf Fagerberg    Kim Skak Larsen

Department of Mathematics and Computer Science  
Odense University, Campusvej 55, DK-5230 Odense M, Denmark  
e-mail: {joan,rolf,kslarsen}@imada.ou.dk

We investigate the problem of implementing a priority queue to be used in a parallel environment, where asynchronous processes have access to a shared memory.

Chromatic trees are a generalization of red-black trees appropriate for applications in such an environment. The main idea of chromatic trees is that of the uncoupling of the actual updates from the rebalancing of the structure afterwards. After an update, the structure is simply left as it is, states of unbalance are merely registered, and background processes deal with these problems by doing rebalancing operations, one at a time, in parallel with searches and updates. In this way, the necessary locking of nodes in the structure that a process needs to perform becomes very localized in space as well as in time, allowing for a high degree of parallelism.

We show how an efficient priority queue for a parallel environment, the *chromatic priority queue*, can be obtained via minor modifications of chromatic trees. We furthermore give an amortized analysis, pertaining to chromatic trees as well as chromatic priority queues, showing that starting with an empty structure, only a (small) constant number of rebalancing operations per update is needed to keep the structure balanced.

More precisely, our main result is the following. A parallel priority queue can be implemented such that the delete operation is worst-case constant time, and insert is carried out as a fast search and constant time update, followed by an amortized constant number of rebalancing operations, which can be performed later by other processes, one at a time. If a general delete is desired, it can be implemented as a fast search and constant time update, followed by an amortized constant number of rebalancing operations, which again can be performed later by other processes, one at a time.

Since our proposal differs from previous work on parallel priority queues in that we do not lock the root of the tree before an update, the number of processes which can work on the structure at any one time is not limited by the height of the tree. In fact, parallelism of the order the size of the tree is possible.

We also show that in chromatic priority queues, as well as in chromatic trees, the number of rebalancing operations which can occur at weighted height  $h$  is an exponentially decreasing function of  $h$ , where weighted height is an appropriate notion of height in chromatic structures, closely related to the actual height. This further indicates that chromatic structures are particularly well-suited for parallel environments, since the further down in the structure a lock from a rebalancing operation occurs, the smaller a subtree which cannot be accessed by other operation.

## Sortieren großer Datenmengen auf massiv parallelen Systemen

Ralf Diekmann\*, Jörn Gehring, Reinhard Lüling\*,  
Burkhard Monien\*, Markus Nübel, Rolf Wanka\* †

Fachbereich Mathematik / Informatik  
Universität-GH Paderborn, D-33095 Paderborn  
e-mail: {diek,joern,rl,bm,nuebel,wanka}@uni-paderborn.de

In diesem Vortrag wird eine Performance-Studie weit verbreiteter paralleler Sortieralgorithmen auf dem Parsytec GCel 3/1024 vorgestellt. Sie ist die erste vergleichende Analyse von Sortierverfahren für ein großes MIMD-System. Basierend auf einem theoretischen Modell der Maschine werden präzise Voraussagen der praktischen Laufzeiten gemacht.

*Das Problem.* Eines der grundlegenden Probleme der Informatik ist das Sortieren großer Datenmengen. Durch Verwendung moderner paralleler Systeme und entsprechender Algorithmen ist es möglich, Sortierzeiten wesentlich zu verringern. Dementsprechend beschäftigt sich die Forschung mit der Bearbeitung umfangreicher Sortierprobleme auf parallelen Systemen. Besondere Beachtung gilt dabei, wegen seiner guten technischen Realisierbarkeit, dem zweidimensionalen  $n \times n$ -Gitter.

*Bisherige praktische Ergebnisse.* Im Laufe von Forschungen und Analysen ergaben sich bisher Implementierungen unterschiedlichster Sortieralgorithmen sowohl auf SIMD- (Thinking Machine CM-2 (1024 Prozessoren), MasPar MP-1 (1024 Prozessoren)), als auch auf MIMD-Maschinen mittlerer Größe (AP1000 (128 Prozessoren), iWarp (64 Prozessoren), iPSC/860 System (64 Prozessoren)). Diese Studien kommen zum überwiegenden Teil zu dem Schluß, daß mit einer Samplesort-Implementierung bei sehr großen Datenmengen jeweils die besten Ergebnisse zu erzielen sind. Weiterhin zeigt sich aber auch, daß bei kleineren bis mittleren Datenmengen das bitone Sortieren dem Samplesort vorzuziehen ist.

*Die neuen Ergebnisse.* Unsere Studie präsentiert Analysen und Messungen von Implementierungen der Verfahren Shearsort, Gridsort, Bitonicsort, Setsort, Radixsort und Samplesort auf dem Parsytec GCel 3/1024, einem 1024-Prozessor-MIMD-System, das als  $32 \times 32$ -Gitter verschaltet ist.

Die Implementierungen wurden anhand einer nach dem *NAS Integer Sort Benchmark* erzeugten Folge von zufälligen 32-Bit Zahlen praktisch ausgetestet. Zur Begründung der erzielten Ergebnisse wird das Verhalten der Algorithmen anhand eines theoretischen Modells analysiert, welches Systemeigenschaften allgemein beschreibt und in dem der GCel hinreichend genau modelliert werden kann. Dazu untersuchen wir den Zeitbedarf der Hauptoperationen: Vergleiche, Speicherbewegungen, Kommunikation und Synchronisation. Es ist uns damit möglich, die Gesamtlaufzeiten von Sortierungen sehr genau abzuschätzen, sofern die auftretende Linkauslastung exakt bestimmt werden kann.

---

\*Gefördert durch die DFG-Forschergruppe "Effiziente Nutzung massiv paralleler Systeme", durch Esprit Basic Research Action Nr. 7141 (ALCOM II), und durch den DFG-Leibniz-Preis.

†Gefördert durch die Volkswagenstiftung.

Unsere Implementierungen und theoretischen Analysen haben zu folgenden Hauptergebnissen geführt:

- Für das gesamte 1024-Prozessor-MIMD-System erzielt das bitone Sortieren die besten Resultate, solange die Anzahl der auf einem Prozessor gespeicherten Schlüssel relativ klein ist ( $< 2048$ ). Für größere Probleme erreicht man mit Samplesort die niedrigsten Laufzeiten.
- Die mit Hilfe des theoretischen Modells vorhergesagten Zeiten weisen eine enge Übereinstimmung mit den gemessenen Werten auf, wodurch wir zu interessanten Aussagen über die Struktur der Algorithmen gelangen. So können wir feststellen, welche Operationen die Gesamtlaufzeiten in welchem Maße beeinflussen.

# Asymptotically Optimal and Practical Routing on the Reconfigurable Mesh

Michael Kaufmann

Fakultät Informatik, Universität Tübingen, Sand 13, 72076 Tübingen, Germany.  
e-mail: [mk@informatik.uni-tuebingen.de](mailto:mk@informatik.uni-tuebingen.de)

Heiko Schröder

Department of E&CE, University of Newcastle, NSW 2308, Australia.  
e-mail: [heiko@faceng.newcastle.edu.au](mailto:heiko@faceng.newcastle.edu.au)

Jop F. Sibeyn\*

Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany  
e-mail: [jopsi@mpi-sb.mpg.de](mailto:jopsi@mpi-sb.mpg.de)

There is increasing interest in reconfigurable meshes for several reasons. One being its constant size diameter, which allows fast execution time for algorithms that are not bound by bisection width. Another being that the hardware cost of reconfigurable meshes is close to the cost of standard SIMD meshes [3].

We present two simple routing algorithms for reconfigurable mesh architectures the first one being asymptotically optimal up to lower order terms, the second one being suboptimal by a factor 3 (for queues of length 4). Both algorithms are worth being presented for different reasons. The first one is very general, i.e. it can be applied to  $k - k$  routing for any  $k$  and runs on reconfigurable arrays of arbitrary dimensions. It is the first algorithm known to be asymptotically optimal up to lower order terms.

The second algorithm is for 1-1 routing on 2D reconfigurable meshes. It is based on the “Move and Smooth” approach used in [1,2]. It is in its simplest version (queue length 4) asymptotically suboptimal by a factor 3 — at the price of drastic increase in queue length and an increase in lower order terms this factor can be reduced close to  $3/2$ . Thus this algorithm is still faster than any other known routing algorithm for reconfigurable meshes. The main justification for the second algorithm is its actual performance on meshes of practical size. We demonstrate that it outperforms the asymptotic faster algorithm by a significant factor for array sizes up to  $1000 \times 1000$ .

## References

- [1] Han, T., Stanat, D.F., ‘ “Move and Smooth” routing algorithms on mesh connected computers’, *Proc. 28th Allerton Conference*, pp 236-245, 1990.

---

\*Partially supported by EC Cooperative Action IC-1000 (Project ALTEC: Algorithms for Future Technologies)

- [2] Kapoor, A., Schröder, H., Beresford-Smith, B., 'Optimal permutation routing on the reconfigurable mesh architecture', *Proc. IPPS 94*, April 1994
- [3] Lie, K.T., Schröder, H., 'A Reconfigurable Mesh Processor Array Utilising Fault-Tolerant Network', *PRFTS 93*, Melbourne, December 1993.

## Shifts by Powers of 2 Are Easy for Switching Networks

Przemysław Kanarek      Krzysztof Loryś

University of Wrocław, PL-51-151 Wrocław, Poland

e-mail:{pka,lorys}@ii.uni.wroc.pl

We consider switching networks that are able to realize every circular shift of  $2^l$  positions for  $0 \leq l \leq \lfloor \log n \rfloor$ . An obvious lower bound on the depth of such a network is  $\Omega(\log \log n)$ . We construct a switching network of depth  $8 \log \log n + 3$  that may perform any of those shifts, thus matching the lower bound up to a constant factor. In fact, the construction is more general and may be used to build depth  $O(\log \log n)$  switching networks for some other families of  $\log n$  shifts that have some regular form.



# Über den erwarteten Delay von Schaltkreisen für die Disjunktion

Beate Bollig\*, Martin Hühne<sup>§</sup>, Stefan Pölt, Petr Savický<sup>¶</sup>

Lehrstuhl Informatik II, Universität Dortmund, 44221 Dortmund  
{bollig, huehne, poelt, savicky}@ls2.informatik.uni-dortmund.de

Bei vielen Algorithmen ist die erwartete Laufzeit, also der Average Case, wesentlich kleiner als der Worst Case. Für Schaltkreise machen Untersuchungen der erwarteten Laufzeit nur dann Sinn, wenn man statt der Tiefe ein anders Maß für die parallele Zeit betrachtet. Ein geeignetes Maß für die Rechenzeit eines Schaltkreises ist sein Delay. Der Delay bei Eingabe  $x$  ist die minimale Zeit, die der Schaltkreis unabhängig von der Initialisierung seiner Bausteine zur Berechnung der Ausgabe benötigt.

Für den Entwurf effizienter Average Case Schaltkreise kann man nun zwei Ansätze verfolgen.

Entweder setzt man voraus, daß die Verteilung der Eingaben im vorhinein bekannt ist. In diesem Fall kann man einen Schaltkreis konstruieren, der für diese spezielle Verteilung im Average Case optimal ist. Jakoby, Reischuk, Schindelhauer und Weis haben dies für eine Reihe grundlegender Funktionen untersucht. Insbesondere geben sie für jede durch einen Tiefe- $d$  Schaltkreis erzeugte Verteilung einen Schaltkreis für die Disjunktion von  $n$  Variablen an, dessen erwarteter Delay  $2^d + 3$  ist, also gar nicht von der Anzahl der Variablen abhängt. Allerdings ist der Worst Case Delay linear in  $n$ .

Im allgemeinen Fall ist die Verteilung der Eingaben nicht genau bekannt. Dann sucht man nach einem einzelnen, „robusten“ Schaltkreis, der für eine weite Klasse von Verteilungen möglichst effizient ist.

Wir zeigen nahezu scharfe obere und untere Schranken für den erwarteten Delay von robusten Schaltkreisen für die Disjunktion. Unsere Verteilungen werden wie bei Jakoby et al. von Schaltkreisen der Tiefe  $d$  erzeugt. Im allgemeinen Fall muß man voraussetzen, daß die Eingaben nicht konstant Null sind.

Wir geben einen robusten Schaltkreis an, dessen erwarteter Delay auf jeder solchen Verteilung höchstens  $2^d + 4d + 4$  ist. Für diese Klasse kann also auch ohne Kenntnis der genauen Verteilung ein fast optimaler Schaltkreis konstruiert werden. Der Worst Case Delay dieses Schaltkreises ist  $\log n + O(\log \log n)$ . Eine zugehörige untere Schranke für den erwarteten Delay robuster Schaltkreise ist  $2^d + d - 2$ .

---

Wir bedanken uns für die freundliche Unterstützung durch:

\*DFG-Projekt We 1066/7-1,

§DFG-Projekt Di 412/2-2, sowie

¶Heinrich-Hertz-Stiftung (B42/44-93).

# On the Application of Local Circuit Transformations with Special Emphasis on PDF Testability\*

Harry Hengster

Rolf Drechsler

Bernd Becker

FB Informatik, J.W.Goethe-Universität, D-60054 Frankfurt/Main

e-mail: `(name)@kea.informatik.uni-frankfurt.de`

A local transformation is the substitution of a subcircuit by a new realization which has better properties with respect to the current goal of optimization. Several types of local transformations and their effect on path delay fault (PDF) testability have been examined theoretically in the literature.

To show that local transformations are practically feasible, we developed SALT (system for application of local transformations), which is a general tool for application of local transformations. To use SALT two circuit descriptions for the definition of each local transformation have to be specified: the search pattern circuit (SPC) and the substitution pattern circuit (SubPC).

Different possibilities are introduced how “related” transformations can be derived from a given transformation. Related transformations are transformations, which adapt the original transformation to specific classes of circuits, like NAND based realizations or two-level circuits, or they use the concept of duality. The computation of dual transformations and the adaptation of a transformation to NAND based realizations can be performed automatically by SALT.

Given the description of a circuit, that is to be transformed, and the description of a local transformation, SALT tries to find a subcircuit which can be matched to the SPC and substitutes this subcircuit by the SubPC. Here, the notion “matched” has to be understood in the following way: During the search phase SALT does not only look for subcircuits which are isomorphic to the SPC but it also searches for patterns that are “pseudo-isomorphic” to the SPC. Inverter shifting and “compatible” modifications at the inputs and outputs of the pattern are used to identify pseudo-isomorphic subcircuits. Using pseudo-isomorphism more subcircuits can be matched to the SPC. On the other hand the substitution process becomes more complicated.

We used SALT to apply testability preserving and testability improving transformations to benchmark circuits. The effect of these transformations (and combinations of different types of transformations) on the size, depth and testability of the transformed circuits is shown by experimental results.

---

\*This work was supported in part by DFG grant Be 1176/4-1.

## Efficient Construction of Reduced $\pi$ OBDDs from FBDDs and $\pi'$ OBDDs

Petr Savický\*

Department of Logic, Faculty of Philosophy,  
Charles University, Prague, Czech Republic  
e-mail: savicky@ls2.informatik.uni-dortmund.de

Ingo Wegener\*\*

FB Informatik, LS II,  
Univ. Dortmund, 44221 Dortmund, Germany  
e-mail: wegener@ls2.informatik.uni-dortmund.de

The problem of converting an FBDD (free binary decision diagram)  $P$  or a  $\pi'$ OBDD (ordered binary decision diagram with respect to the variable ordering  $\pi'$ )  $P$  for the Boolean function  $f$  into a reduced  $\pi$ OBDD  $Q$  for  $f$  is considered. The algorithms run in time  $O(\text{size}(P)\text{size}(Q)\log\text{size}(Q))$  and need space  $O(\text{size}(P)\text{size}(Q))$ , if  $P$  may be an FBDD, or  $O(\text{size}(P) + \text{size}(Q))$ , if  $P$  is known to be an OBDD.

The problem is important for the improvement of given orderings, e.g. by simulated annealing or genetic algorithms, and in the situation, where not compatible representations of functions have to be made compatible.

Moreover, it can be tested in time  $O(\text{size}(P)\text{size}(Q)\log\text{size}(Q))$  whether an FBDD  $P$  and an OBDD  $Q$  represent the same function.

---

\*The work was done while this author was visiting the Univ. Dortmund with a grant of Heinrich-Hertz-Stiftung No. B42/44-93.

\*\*Supported in part by DFG grant We 1066/7-1.

# Turing-Reduzierbarkeit auf p-selektive Mengen

Hans-Jörg Burtschick      Wolfgang Lindner

TU Berlin, Fachbereich Informatik

Franklinstr. 28/29, D-10587 Berlin

e-mail: {hjoerg,linus}@cs.tu-berlin.de

Wir betrachten die nichtuniforme Komplexität von Mengen, die auf p-selektive Mengen Turing-reduzierbar sind. Insbesondere setzen wir die Anzahl der Fragen an ein p-selektives Orakel mit der Advice-Länge in Beziehung.

Mit Hilfe einer Zähltechnik zeigen wir u. a. die folgenden Inklusionen:

- Sei  $q$  ein Polynom. Jede Menge, die mit höchstens  $q(n)$  vielen Fragen an ein p-selektives Orakel auf polynomiell Platz entschieden werden kann, kann auch nichtuniform auf polynomiell Platz und mit Advice-Länge  $O(n + q(n))$  entschieden werden.
- Für alle  $c \in \mathbb{N}$  gibt es  $c' \in \mathbb{N}$ , so daß jede Menge, die mit höchstens  $c \cdot n$  vielen Fragen an ein p-selektives Orakel in polynomieller Zeit entschieden werden kann, auch nichtuniform in Zeit  $2^{c' \cdot n}$  mit Advice-Länge  $c' \cdot n$  entscheidbar ist. (Dies gilt auch für Mengen, die mit höchstens  $c \cdot n$  vielen Fragen an ein p-selektives Orakel in *subexponentieller* Zeit entschieden werden können.)

Es ist wohlbekannt, daß sich  $P/poly$  als die Klasse der Mengen charakterisieren läßt, die auf p-selektive Mengen in polynomieller Zeit Turing-reduzierbar sind. Wir erhalten die folgenden Charakterisierungen der Klassen  $PSPACE/poly$  und  $EXP/poly$ :

- Eine Menge ist genau dann in  $PSPACE/poly$  (bzw.  $EXP/poly$ ), wenn sie mit höchstens polynomiell vielen Fragen an ein p-selektives Orakel auf polynomiell Platz (bzw. in exponentieller Zeit) entschieden werden kann.

Schließlich zeigen wir, daß für beliebige Konstanten  $c, k \in \mathbb{N}$  die (uniformen) Komplexitätsklassen  $E$  und  $EXP$  nicht in  $DTIME(2^{c \cdot n})/O(c \cdot n)$  bzw.  $DTIME(2^{n^k})/n^k$  enthalten sind. Daraus folgt:

- Es gibt eine Menge in  $E$ , die nicht mit  $c \cdot n$  vielen Fragen an ein p-selektives Orakel in polynomieller (subexponentieller) Zeit entschieden werden kann.
- Es gibt eine Menge in  $EXP$ , die nicht mit  $n^k$  vielen Fragen an ein p-selektives Orakel in Zeit  $2^{n^k}$  entschieden werden kann.

# Exploring the Sources of Computational Inefficiency in Inductive Inference

Werner Stein

Universität Kaiserslautern

e-mail: [stein@informatik.uni-kl.de](mailto:stein@informatik.uni-kl.de)

Given a sequence of examples of an unknown object, the goal of inductive inference is to learn a global description for it. We restrict ourselves to the object class of recursive functions. A set of functions is said to be learnable in the limit ( $U \in \text{LIM}$ ) if there is a strategy, such that for each function  $f$  in the set, if feeding the strategy with the sequence  $f(0), f(1), \dots, f(n), \dots$ , it will produce hypotheses for the unknown object which eventually are all one and the same and a description of  $f$ .

It is a well known theorem that every set learnable in the limit, can also be learned by a strategy with linear computational complexity. But the task of learning is not as easy as it seems. In many cases, the strategy has to produce hypotheses fulfilling restrictions and conditions according to the consumers wishes. Examples of what a consumer may require of the strategy are:

- to produce hypotheses consistent with the examples seen so far.
- to produce hypotheses each time closer to the target concept.
- to produce hypotheses always describing a learnable concept.
- always to produce a hypothesis (even if the given function cannot be learned).
- to produce hypotheses having no “loud” mistake.

The classes of sets of functions learnable by such strategies are well explored. But it is an open problem if the linear complexity bound for LIM-strategies is also valid for restricted strategies.

We have shown that **consistency** is a condition having no such fine complexity bound. We will have a look to the reason why there is no upper bound (it turned out that the undecidability of the consistency check is not the only source of inefficiency). We will see that in many cases there is at least a lower complexity bound for learning a class consistently.

Furthermore, we will examine more restrictions and conditions with respect to upper and lower bounds of the computational complexity. We found at last three conditions for strategies such that there is no global upper bound for the general learning task; all other turned out to have at least a linear upper complexity bound. It is still unknown if there are conditions having other worst case bounds.

# Lernen aus Random Walks

Peter L. Bartlett\*

Department of Systems Engineering RSISE,  
Australian National University, Canberra, 0200 Australia  
e-mail: `Peter.Bartlett@anu.edu.au`

Paul Fischer<sup>‡</sup>      Klaus-Uwe Höffgen<sup>§</sup>

Lehrstuhl Informatik II, Universität Dortmund  
D-44221 Dortmund, Germany  
e-mail: `{paulf,hoeffgen}@goedel.informatik.uni-dortmund.de`

In diesem Vortrag stellen wir eine neue Variante für passives Lernen vor. Anders als im „klassischen“ PAC-Modell nehmen wir nicht an, daß die Beispiele unabhängig anhand einer unterliegenden Verteilung gezogen werden. Stattdessen gehen wir davon aus, daß sie von einem zeitabhängigen Prozeß erzeugt werden. Aus der Tatsache, daß aufeinanderfolgende Beispiele abhängig sind, kann man oft zusätzliche Information gewinnen, die effizientes Lernen möglich macht. Wir zeigen, wie man in diesem Modell On-Line Vorhersagealgorithmen entwerfen kann und vergleichen es mit anderen Lernmodellen. Speziell stellen wir effiziente Algorithmen für das Lernen von Booleschen Schwellwertfunktionen, 2-Term-RSE und 2-Term-DNF vor. Der unterliegende Prozeß ist dabei ein Random Walk entlang der Kanten des Booleschen Würfels.

---

\*The author thanks the Australian Telecommunications and Electronics Research Board for their support.

<sup>‡</sup>Unterstützt durch die Deutsche Forschungsgemeinschaft, Beihilfe We 1066/6-1.

<sup>§</sup>Der Autor dankt für die Unterstützung durch das Bundesministerium für Forschung und Technologie, Beihilfe 01IN102C/2.

Der folgende Text stammt aus Trier.

```
#####  
#                                                                 #  
#  WORKSHOP FOR COMPLEXITY THEORY, DATA STRUC- #  
#    TURES AND EFFICIENT ALGORITHMS (CTDSEA) #  
#                                                                 #  
#          GUIDELINES FOR WORKSHOP HOSTS      #  
#                                                                 #  
#####
```

NOTE: Since this workshop took place in german towns only so far,  
we edited these guidelines in German. If you need an english  
Version, please contact the mailinglist maintainer via email  
as workshop-request@ti.uni-trier.de.

```
#####
```

Inhaltsverzeichnis:

1. Die zentrale Adressenliste
2. Wie erreiche ich ...
  - a) ... alle Interessenten
  - b) ... den Listenverwalter
  - c) ... den / andere Veranstalter
  - d) ... einen bestimmten Teil der Interessenten
3. Durchführung des Workshops
  - a) Einladung
  - b) Programm
  - c) Unterlagen
  - d) Adressen-Update

```
#####
```

## 1. DIE ZENTRALE ADRESSENLISTE

Auf dem 22. Workshop in Saarbrücken wurde beschlossen, daß die  
Einladungen ab sofort (wie schon bei den beiden Workshops davor)  
nur noch mit elektronischer Post verschickt werden sollen. Das  
bedeutet, daß nicht mehr - wie bisher - die kompletten Adressen

der Interessenten an die Veranstalter übermittelt werden, damit dieser damit Adreßetiketten bedrucken kann. Die Liste der Interessenten für den Workshop wird in Trier geführt und den Veranstaltern als Mailinglist zur Verfügung gestellt. Zumindest für eine Übergangszeit werden auch die postalischen Adressen der Interessenten gespeichert bleiben.

Die Mailinglist existiert auf einer SUN, also einer Unix-Workstation. Das verwendete primäre Mailsystem ist dementsprechend SMTP. Wir verfügen über (z. Z. funktionierende) Gateways für X.400, UUCP, CSNET und Bitnet.

#####

## 2. WIE ERREICHE ICH ...

A) ... alle Interessenten?

Da die Liste als funktionierende Mailinglist geführt wird, kann man elektronische Post an diese Liste schicken, die dann an alle Interessenten weiterverteilt wird. Falls notwendig, können wir auch die kompletten Daten (Email- und postalische Adresse) wie bisher zur lokalen Verarbeitung zur Verfügung stellen. Wir weisen jedoch darauf hin, daß die Speicherform dieser "Rohdaten" schon mehreren Veranstaltern Schwierigkeiten gemacht hat. Automatische Erstellung von Adreßetiketten (mit LaTeX, das vorhanden sein muß) ist jedoch relativ problemlos.

Die Mailinglist ist zu erreichen unter den folgenden Adressen:

```
Per SMTP:      workshop-list@ti.uni-trier.de
Per X.400:      S=workshop-list;OU=ti;P=uni-trier;A=;C=de
Per UUCP:      ...!ti.uni-trier.de!workshop-list
Per CSNET:      ##### Noch unbekannt #####
Per Bitnet:     ##### Noch unbekannt #####
```

B) ... den Listenverwalter?

Passend zur Mailinglist gibt es ein Alias für den Verwalter der Liste, nämlich workshop-request@ti.uni-trier.de (SMTP Adresse). Die anderen Adressen bilden sich analog zu oben.

C) ... den / andere Veranstalter?



Neben der Liste der Interessenten verwalten wir auch eine Liste, wann der Workshop wo stattfand und welche Email-Adresse dort als Kontakt diente. Diese Adressen werden nicht zu einer Mailinglist gefaßt. Bitte wenden Sie sich an den oben erwähnten Verwalter der Adressenlisten.

D) ... einen bestimmten Teil der Interessenten?

Wir halten keine Teillisten. Wenn Sie eine Unterscheidung der Teilnehmer in spe treffen wollen, müssen Sie die Daten entweder selbst sammeln oder sich die Liste von uns kopieren und dann teilen. Auch hierzu wenden Sie sich bitte an den Listenverwalter.

#####

### 3. DURCHFÜHRUNG DES WORKSHOPS

In der Vergangenheit hat sich das folgende Vorgehen bei der Vorbereitung und Durchführung des Workshops bewährt:

#### A) Einladung

Einladungen zu einem Workshop sollten mindestens sechs Wochen im Voraus an alle Interessenten abgeschickt werden. Achten Sie bitte auf die Bekanntgabe folgender Punkte:

- a) Ort und Datum des Workshops
- b) Aufforderung zur Anmeldung von Vorträgen (z. B. durch Einreichung eines Ein-Seiten-Abstract)
- c) Angabe eines Stichtags für die Einreichung der Abstracts
- d) Angabe einer verlässlichen Email-Kontaktadresse
- e) Eventüll Angabe einer postalische Adresse

#### B) Programm

Programm und Anreisebeschreibung sollten circa vierzehn Tage vor dem Workshop allen Teilnehmern bekanntgegeben werden.

#### C) Unterlagen

Üblicherweise werden den Teilnehmern des Workshops die Ab-

stracts als Tagungsunterlage in Form eines Technical Report zur Verfügung gestellt.

D) Adressen-Update

Um die Liste der Interessenten aktuell zu halten, bitten wir alle Veranstalter, eine Anwesenheitsliste mit Name und zumindest Emailadresse anzufertigen. Bitte senden Sie die gesammelten Daten an den Listenverwalter, der sie mit der Liste abgleicht.

Außerhalb der Workshops kann man sich direkt über den Listenverwalter in die Workshop-Liste aufnehmen lassen.

#####