Ernst. W. Mayr, Friedhelm Meyer auf der Heide
(editors)

# Parallel and Distributed Algorithms

Dagstuhl-Seminar-Report; 8
4.3.1991 - 8.3.1991 (9110)

ISSN 0940-1121


Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.



Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: dagstuhl@dag.uni-sb.de

INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK

Schloß Dagstuhl

Tagungsbericht 10/1991

# Parallele und verteilte Algorithmen

4.–8. März 1991

The first workshop on *"Parallel and distributed algorithms"* at the IBFI was organized by Ernst W. Mayr (Frankfurt) and Friedhelm Meyer auf der Heide (Paderborn). There were twenty-six participants from five countries. Regrettably, most of the invitees from the US, Canada, and Israel had to cancel due to travel restrictions because of the Gulf war.

Twenty-four lectures were presented at the workshop, covering a wide range of topics in parallel complexity theory, parallel and distributed algorithms, parallel architectures and models of parallel and distributed computing.

Lectures in parallel complexity theory dealt with communication complexity, depth reduction for unbounded fan-in circuits and shallow circuits for symmetric functions.

A large number of algorithmic problems was covered, including clustering algorithms, parallel algorithms for sorting integers, dynamic programming and context-free recognition. Other talks were concerned with the transformation of parallel algorithms on idealistic machine models to more realistic ones, and with basic implementation issues on such more realistic networks, e.g. distributed load balancing, wait-free parallel algorithms, routing on grids, and the issue of parallel data structures like heaps and priority queues, or distributed dictionaries.

Other, more specific topics arising from architectural problems were the bisection problem for graphs of degree 4, the determination of close upper and lower bounds for the crossing number of hypercubes and cube-connected-cycles, and a time-randomness tradeoff.

Finally, there were several talks on parallel architectures and computation models. On the one hand, the theoretical PRAM model was shown to have a lot of practical appeal after all, given some recent advances in technology, and on the other, even more powerful models of parallel computation, like synchronized alternation, were presented. There were also some musings on the fundamental issues in parallel complexity theory and computation, as well as a survey on the state of the art in neurocomputing.

The setup of IBFI and the workshop provided ample time for discussions among the participants, be it on topics related to the theme of the workshop, or on politics and its fast pace at the moment.

# Participants

Ferri Abolhassan, Saarbrücken
Richard Anderson, Seattle
Elias Dahlhaus, Bonn
Martin Dietzfelbinger, Paderborn
Torben Hagerup, Saarbrücken
Bernd Halstenberg, Darmstadt
Ulrich Hertrampf, Würzburg
Juraj Hromkovič, Paderborn
Jörg Keller, Saarbrücken
Manfred Kunde, München
Mirosław Kutyłowski, Paderborn
Klaus-Jörn Lange, München
Reinhard Lüling, Paderborn
Seshu Madhavapeddy, Paderborn
Ernst Mayr, Frankfurt
Kurt Mehlhorn, Saarbrücken
Friedhelm Meyer auf der Heide, Paderborn
Burkhard Monien, Paderborn
Ian Parberry, Denton
Wolfgang J. Paul, Saarbrücken
Peter Rossmanith, München
Wojciech Rytter, Warszawa
Ondrej Sýkora, Saarbrücken
Thomas Tensi, München
Imrich Vrťo, Saarbrücken
Rolf Wanka, Paderborn
Ingo Wegener, Dortmund
Juraj Wiedermann, Bratislava

# Contents

# Abstracts

### PRAMs — An Efficient Alternative to Distributed Memory Machines

by FERRI ABOLHASSAN (joint work with Jörg Keller, Wolfgang J. Paul)

Todays parallel computers provide good support for problems that can be easily embedded on the machines' topologies with regular and sparse communication patterns. But they show poor performance on problems that do not satisfy these conditions. A general purpose parallel computer should guarantee good performance on most parallelizable problems and should allow users to program without special knowledge about the underlying architecture. Access to memory cells should be fast for local and non local cells and should not depend on the access pattern. A theoretical model that reaches this goal is the PRAM. But it was thought to be very expensive in terms of constant factors.

Our goal is to show that the PRAM is a realistic approach for a general purpose architecture for any class of algorithms. To do that we sketch a measure of cost–effectiveness that allows to determine constant factors in costs and speed of machines. This measure is based on the price/performance ratio and can be computed by hand without building a machine. The smaller the measure $M$, the better the machine. We describe a realistic PRAM architecture based on RANADE's Fluent Machine and an architecture of a distributed memory machine (DMM) that reflects our view of existing parallel machines. This architecture is very general, the processor interconnection is given as a parameter. We restrict the architecture to use only neighbour–to–neighbour communications.

To compare these two models with the above measure we construct worst case situations for PRAMs and DMMs. In the first case the global memory is not used at all, in the latter any known algorithm for a DMM is worse than the simulation of a PRAM on that machine. This leads to upper and lower bounds $U = O(\log n)$ and $L = \Omega(1)$ for the term M(PRAM)/M(DMM) as one would expect. The astonishing fact is that the constant factor in $U$ is quite small, the one in $1/L$ however is quite large. We conclude that for reasonable values of $n$ a PRAM cannot be much worse than a DMM but vice versa. We show by examples — matrix multiplication and connected components of a graph — that both bounds are tight.

### Wait-free Parallel Algorithms for the Union-Find Problem

by RICHARD J. ANDERSON (joint work with Heather S. Woll)

We are interested in designing efficient data structures for a shared memory multiprocessor. In this paper we focus on the union-find data structure. Our machine model is asynchronous and allows stopping faults. Thus we require our solutions to the data structure problem have the *wait-free* property, meaning that each thread continues to make progress on its operations, independent of the speeds of the other threads. In this model efficiency is best measured in terms of the total number of instructions used to perform a sequence of data structure operations, the *work* performed by the processors. We give a wait-free implementation of an efficient algorithm for union-find. In addition we show that the worst case performance of the algorithm can be improved by simulating a synchronized algorithm, or by simulating a larger machine if the data structure requests support sufficient parallelism. Our solutions apply to a much more general adversary model than has been considered by other authors.

### Parallel Hierarchical Clustering

by ELIAS DAHLHAUS

The problem of hierarchical clustering is the following:
Given a distance function $d : \Omega^2 \to I\!R^+$. Find a tree $T$ with leaf set $\Omega$ and an age function $a : V_T \to I\!R^+$, such that:

i) $a \restriction \Omega \equiv 0$
ii) $a(parent(x)) > a(x)$

and

iii) $d'(x,y) = a(LCA(x,y))$ is similar to $d$ (Dendrogram).

It is well known that $d' = d$ can be realized if and only if $d$ is an ultrametric, i. e. $d(x,z) \leq \max(d(x,y), d(y,z))$.

In this talk a parallel algorithm for the recognition of an ultrametric and the construction of its dendrogram is presented. The processor number is $O(n^2/\log n)$ and the time is $O(\log n)$.

Also a parallel algorithm for the single linkage heuristics is presented.

### Simulating Restricted PRAMs on Complete Networks

by MARTIN DIETZFELBINGER (joint work with Friedhelm Meyer auf der Heide)

We study the problem of simulating a $p$-processor PRAM on a complete network of $p$ processors. It is known that one PRAM step can be simulated in $O(\log p)$ network steps deterministically and in (expected) $O(\frac{\log p}{\log \log p})$ by a network of randomized processors. We argue that there is little chance for improvement if step-by-step simulation is used and memory cells are distributed by a random hash function among network processors.

Utilizing our previous construction of a distributed dictionary we propose a way of asynchronously simulating a certain subclass of exclusive-read exclusive-write PRAMs faster: We demand that a PRAM processor that reads from a shared memory cell knows (from the part of the computation it has seen so far) which other processor has written to that cell. We demonstrate that such PRAMs can be simulated in $O(\sqrt{\log p})$ steps (expected) on a network. Even more restricted classes of PRAMs (e. g., where in addition each cell is written to only once) can be simulated without time loss.

### Improved Parallel Integer Sorting without Concurrent Writing

by TORBEN HAGERUP (joint work with Susanne Albers)

We show that $n$ integers in the range $1 \ldots n$ can be stably sorted on an EREW PRAM using $O((\log n)^{5/3}(\log \log n)^{1/3})$ time, $O(n(\log n)^{2/3}(\log \log n)^{1/3})$ operations and $O(n)$ space and that $n$ arbitrary integers can be sorted on a randomized CREW PRAM within the same bounds for time, operations and space with high probability. In both cases, our algorithm is closer to optimality than all previous algorithms for the stated problem in the stated model. We also show that $n$ integers in the range $1 \ldots m$ can be sorted on an EREW PRAM with a nonstandard word length of $\Theta((\log n)^2 \log \log n \log m)$ bits using $O((\log n)^2)$ time, $O(n/(\log n)^2)$ processors and $O(n)$ space, thereby greatly improving the upper bound on the word length necessary to sort integers with a linear time-processor product, even sequentially. Our algorithms are inspired by, and in one case directly use, the fusion trees recently introduced by Fredman and Willard.

### Circuits and Communication Complexity

by BERND HALSTENBERG

We consider communication complexity of Boolean functions in the two-processors model at the bit level as introduced by Yao. We show how to interpret certain kinds of communication protocols as Boolean circuits of a special form. Inputs are divided into two groups each of which is processed by one circuit with $2^{polylog}$ outputs. The outputs of these two circuits are inputs two a quite simple regular circuit of constant depth using gates with fan-in $2^{polylog}$. Using this interpretation we study how to replace gates of the simple circuit by different types of gates (e.g. AND/OR by MAJORITY). Other results exhibit non-collapsing hierarchies in the number $k$ of inputs to PARITY-gates for all $k \in I\!N$. This work is based on results from structural complexity theory and an earlier work co-authored by Rüdiger Reischuk.

### Depth Reduction for Circuits of Unbounded Fan-in

by ULRICH HERTRAMPF (joint work with Eric Allender)

We consider circuits of constant depth and unbounded fan-in with AND, OR, and PARITY gates. We define the function $f_d$:

$$f_d(x_0, \ldots, x_{n-1}) := \mathrm{MOD}_3(x_0, \ldots, x_{s-1})$$

where $s := \log^d n$, $d$ some constant positive integer. We show that $f_d \in \mathrm{AC}^0_{2d}$, the class of AND, OR circuits of polynomial size and depth $2d$. By a result of Smolensky (1987) the function $f_{2kr+1}$ cannot be computed in depth $k$ and size $2^{\log^r n}$ by AND, OR, PARITY circuits. Thus we know that optimal simulations of $\mathrm{AC}^0$ in a fixed constant depth with AND, OR, PARITY circuits requires at least size $2^{polylog}$.

Then we define the class

$$\textit{Uniform}\text{-AND, OR, PARITY-}\textit{DEPTH}(k)\textit{SIZE}(2^{O(\log^r n)})$$

and show that it can be simulated probabilistically in depth 2, where at most $O(\log^{3r} n)$ random bits are used. This leads to a simulation in depth 4 by deterministic circuits. Several similar simulations are proved.

Finally, using an approach with the Nisan, Wigderson (1988) pseudorandom generators we show that probabilistic AND, OR circuits of size $2^{polylog}$ (with many random bits) can be simulated in depth 4 by deterministic AND, OR, PARITY circuits.

### Synchronized Alternation

by JURAJ HROMKOVIČ

A survey is presented about new characterizations of fundamental complexity classes by synchronized alternation and globally synchronized alternation. A new description of nondeterministic and deterministic space bounded complexity classes in terms of these two types of generalized alternation. For example, linear nondeterministic (deterministic) space is exactly the family of languages recognized by synchronized (globally deterministic synchronized) alternating finite automata. Thus, many question concerning the relation between nondeterministic and deterministic space can be expressed as the question whether synchronization is more powerful that globally deterministic synchronization for some simple kinds of automata.

### Balanced Routing on Grids of Processors

by MANFRED KUNDE

The problem of packet routing on an $r$-dimensional grid of processors with side length $n$ is studied. Each processor is able to store $rf(n)$ packets, $f(n) < n^{1-1/r}$. The new class of balanced routing problems is introduced which includes such fundamental classes as partial $h-h$ routing. On a 3-dimensional $n \times n \times n$ grid (without wrap-arounds) partial permutation problems and so called $(1, f(n))$-balanced problems can deterministically be solved within at most $3.333n + O(n/f(n)^{1/2})$ steps, which is only slightly larger than the distance bound of $3n - 3$ steps. The algorithm can be extended to the $r$-dimensional case where it needs $(r + (r-2)^{r-2}\sqrt{1/r})n + O(n/f(n)^{1/(r-1)})$ transport steps. Thus the algorithm beats all so far known randomized algorithms as well as deterministic algorithms. The number of steps is reduced to the half if the algorithm is adapted to tori of processors, i. e. grids with wrap-around connections.

### Time Optimal Realistic CREW PRAM Algorithms

by MIROSŁAW KUTYŁOWSKI (a report on a joint work with Martin Dietzfelbinger and Rüdiger Reischuk)

It has been proved that the computation time for most Boolean functions of $n$ arguments is at least $\phi(n) \approx 0.72.. \cdot \log_2 n$ on CREW PRAMs with arbitrarily large resources. On the other hand, any Boolean function of $n$ arguments can be computed in $\phi(n)+2 \approx 0.72.. \log_2 n$ steps, but on a CREW PRAM with $n \cdot 2^{n-1}$ processors. We show that for many functions the computation time $\phi(n) + o(\log_2 n)$ can be achieved on "realistic" CREW PRAMs, i. e., with a small number of processors and common memory cells (in most cases $n$) that may store only binary words of a bounded length (in most cases of length 1). Such algorithms are presented, among others, for $PARITY$, for evaluation of many Boolean formulas, for all symmetric functions and some functions over many-valued domain $\{1, 2, \ldots, k\}$ for small $k$. Similar algorithms have been found for adding binary numbers, sorting bit strings or sorting strings of binary numbers.

### Some Results of KLARA

by KLAUS JÖRN LANGE

A short overview of the ongoing work of the project "Classification and Parallelization by Analysis of Reductions" (KLARA) is given. There are three main directions of KLARAs research:

In the part 'inheritance' (Vererbbarkeit) new notions of reducibility are investigated which respect problem properties which are relevant for the implementation on asynchronous distributed memory machines.

The part 'characterization' considers relations between several models of parallel complexity theory like PRAMs, circuits, and Auxiliary Push-Down Automata. These investigations are influenced by the results of 'inheritance'.

The part 'heuristics' deals with the parallel treatment of Boolean formulae. Some techniques known for unquantified formulae are transferred to quantified ones.

### Dynamic Distributed Load Balancing Algorithms

by REINHARD LÜLING

If the computation and/or communication load of a distributed computation are unknown or vary dynamically during runtime it is necessary to use dynamic distributed load balancing strategies to maximize the performance of the distributed computation.

Results were presented about the implementation of a distributed branch and bound algorithm on a 256 processor network connected as a De Bruijn graph. Speedup results of up to 237 for very short parallel computation times were achieved using an adaptive load balancing method.

Simulations on a ring network of 128 processors with a speedup of 112 show that good results can also be achieved on networks with even larger processor numbers.

In Part 2 different load balancing algorithms were compared using synthetic workload generation.

### Optimal Algorithms for Dissemination of Information in Generalized Communication Modes

by SESHU MADHAVAPEDDY (joint work with Rainer Feldmann, Juraj Hromkovič, Burkhard Monien and Peter Mysliwietz)

Some generalized communication modes enabling the dissemination of information among processors of interconnection networks via vertex-disjoint or edge-disjoint paths in one communication step are investigated. A thorough study of these communication modes is presented by giving optimal algorithms for *broadcasting*, *accumulation* and *gossiping* in most of the fundamental parallel architectures (complete graphs, Hypercubes, Cube Connected Cycles, Butterfly networks, complete binary trees, etc.). An interesting consequence of the presented algorithms is the fact that in almost all these interconnection networks the gossip problem cannot be solved in time less than the sum of time complexities of the accumulation problem and the broadcast problem (i. e. for most networks the optimal algorithm for the gossip problem is simply the concatenation of optimal algorithms for accumulation and broadcasting). Moreover, some closed lower and upper bounds on the complexity of broadcasting, accumulation and gossiping are given for two important families of graphs: trees and $k$-degree bounded graphs.

### A Time-Randomness Tradeoff in Communication Complexity

by KURT MEHLHORN (joint work with Rudolf Fleischer and Hermann Jung)

Let $x_i, y_i \in \{0,1\}^n$ for $1 \leq i \leq n$ and define

$$f(x_1 \ldots x_n, y_1 \ldots y_n) = \begin{cases} 1 & \exists i : x_i = y_i \\ 0 & \forall i : x_i \neq y_i \end{cases}$$

We consider the communication complexity of $f$ in the Las Vegas model. For an algorithm $A$, let $T_A$ be the expected number of bits exchanged and let $R_A$ be the number of random bits consumed.

L1: For all algorithms $A$: $R_A \geq \log(n^2/T_A)$.

L2: There is an algorithm $A$: $R_A \leq (1 + o(1)) \log(n^2/T_A)$

L1 plus L2 together give a tight trade-off between communication time and the amount of randomness used.

### The Bisection Problem for Graphs of Degree 4

by BURKHARD MONIEN (joint work with Juraj Hromkovič)

We study the bisection problem where for a given graph $G$ a balanced partition with a minimal number of crossing edges has to be found. We are interested in the question how large the minimal bisection of $k$-degree bounded graphs may be. We present two results:

- Every $k$-regular graph with $n$ nodes, $n = d \cdot p$, can be partitioned into $d$ components, each of $p$ vertices, by removing at most $\frac{k-2}{2} \cdot \frac{d-1}{d} \cdot n + o(n)$ edges.
- Every 4-regular graph with $n$ nodes, $n$ even, $n > 512$, has a partition with at most $n/2 + 2$ crossing edges.

Our results have applications in optimizing the configuration hardware of Transputer systems.

### Ragged Heaps, Parallel Priority Queues, and Halving Networks

by IAN PARBERRY

A *ragged heap* is an $n$-node heap with $O(\log n)$ empty nodes, and at most two extra values per level. Using this data structure, a sequence of priority queue operations can be processed with $O(1)$ delay per operation by a constant degree network of $\log n$ processors, where $n$ is the maximal size of the priority queue. These algorithms can be extended to a *parallel priority queue*, in which $p$ operations of the same type are to be processed concurrently, with delay $O(\log p)$ per operation on $O(p \log n)$ processors. A more practical algorithm with a smaller constant multiple in the time bound is obtained using halving networks.

### Towards a Theory of Computer Architecture

by WOLFGANG J. PAUL (joint work with S. M. Müller)

We treat computer architecture as a formal optimization problem. We view an architecture as a 4-tuple $(H, IS, C_0, L)$ where $IS$ is an instruction set, $H$ is hardware supporting $IS$, $L$ is a high level language and $C_0$ is a compiler from $L$ to $IS$. Our hardware model is an extension of switching circuits. It contains some extra components like registers, drivers, busses, RAMs and ROMs. It also contains some parameters of technology. These include memory access times (measured in gate delays) and packing densities for various types of of circuits. We model workload by benchmarks. Here we use dhrystone. In this framework we compare RISC and CISC architectures. It turns out, that both architectures are good solutions to the optimization problem: optimize price/performance on dhrystone for the actual technology parameters at the time of invention of these architectures.

### PRAMs and Circuits with Bounded Fan-out

by PETER ROSSMANITH (joint work with Inga Niepel)

Relations between PRAMs with several memory access restrictions and other computational models are given. In particular all XRYW-PRAMs, $X, Y \in \{E, C, O\}$, are considered.

1. A circuit characterization of EREW-PRAMs is given. Furthermore this leads to circuit characterizations of CRCW-, CREW-, ERCW-, and EREW-PRAMs in a uniform way. For this purpose bounded fan-out circuit which include SELECT-gates are introduced.
2. It holds: $\text{EREW}^k = \text{OREW}^k$ and $\text{ERCW}^k = \text{ORCW}^k$.
3. OROW-PRAMs are introduced as a PRAM-model with restricted communication ability. OROW-PRAMs are essentially complete processor networks without shared memory. Nevertheless they are very powerful: $LOGSPACE \subseteq \text{OROW}^1$.

### Sublinear Time Parallel Computations of Dynamic Programming Recurrences and Context-free Recognition

by WOJCIECH RYTTER (joint work with L. Larmore)

We are given dynamic programming recurrences

$$\begin{aligned} \text{cost}(i,j) &= \min\{\text{cost}(i,k) + \text{cost}(k,j) + f(i,j,k) \mid i < k < j\} \text{ for } i < j-1, \\ \text{cost}(i,i+1) &= \text{init}(i), \end{aligned}$$

$0 \le i < j \le n$, $n$-size. The values of $f(i,j,k)$ and $\text{init}(i)$ are given. We present a sublinear time parallel algorithm computing costs with total work $O(n^3)$. This matches the best sequential algorithm.

For three subclasses of context-free languages (unambiguous, deterministic, linear), we give sublinear time algorithm with total work $\text{Seq}(n) \cdot n^\alpha$, $0 < \alpha \le 0.5$, where $\text{Seq}(n)$ is the best time complexity of a known sequential algorithm for a given class ($\text{Seq}(n) \approx n^2$ for unambiguous and linear cfl's, and $\text{Seq}(n) \approx n$ for deterministic cfl's).

### On the Crossing Number of the Hypercube and the Cube Connected Cycles

by ONDREJ SÝKORA (joint work with Imrich Vrťo)

The $n$-dimensional hypercube graph $Q_n$ is defined as the 1-skeleton of the $n$-dimensional cube. Hypercubes have been much studied in the graph theory. Interest in hypercubes has been increased by the recent advent of massively parallel computers. The cube connected cycles graph ($CCC_n$) is obtained from $Q_n$ by replacing each vertex by the $n$-vertex cycle. Computers with $CCC_n$ architecture have a similar computing power as those with $Q_n$ structure.

The crossing number $cr(G)$ of a graph $G$ is defined as the least number of crossings of its edges when $G$ is drawn in a plane. According to the survey paper by Harary, Hayes and Wu, all that is known on the exact value of $cr(Q_n)$ is $cr(Q_3) = 0$, $cr(Q_4) = 8$ and $cr(Q_5) \le 56$. In addition, Eggleton and Guy announced that

$$cr(Q_n) \le \frac{5}{32} 4^n - \left\lfloor \frac{n^2+1}{2} \right\rfloor 2^{n-1} \tag{1}$$

but a gap has been found in the description of the construction. Anyway, Erdős and Guy conjectured equality in (1).

In this paper we prove the following bounds for $cr(Q_n)$ and $cr(CCC_n)$ :

$$\frac{4^n}{20} - (n+1)2^{n-2} < cr(Q_n) < \frac{4^n - (3n^2+4)2^{n-2}}{6}$$

$$\frac{4^n}{20} - 3(n+1)2^{n-2} < cr(CCC_n) < \frac{4^n + (9n^2 - 36n - 8)2^{n-2}}{6}.$$

In practice, crossing numbers appear in the fabrication of VLSI circuits. The crossing number of the graph corresponding to the VLSI circuit has strong influence on the area of the layout as well as on the number of wire – contact cuts that should be as small as possible. Leighton was the first who pointed out the importance of crossing numbers in VLSI complexity theory. He showed that the crossing number serves as a good area lower bound argument. Thus our lower bounds for $cr(Q_n)$ and $cr(CCC_n)$ give immediately an alternative proof that the area complexity of hypercube and $CCC$ computers realized on VLSI circuits is $A = \Omega(4^n)$.

**Acknowledgment.** The authors thank to Professor Kurt Mehlhorn, Max-Planck Institut für Informatik, Saarbrücken and Alexander von Humboldt Foundation, Bonn, Germany, who supported this research.

### Oblivious Restricted Message Routing on Grids and Offline-Routing on Grids

by THOMAS TENSI (partially joint work with Manfred Kunde)

The problem of oblivious message routing on grids of processors is studied. A model of restricted message routing is introduced which automatically avoids buffer overflow. We show that on an $n \times n$ grid the simple row-column greedy strategy is deadlock-free and has a lower bound of $\Omega(n^2 m)$ steps when each packet is divided into $m$ flits. This bound is valid for several tie-breaking rules even in the case where a kind of time-fairness is provided. On $r$-dimensional grids with sidelength $n$ a lower bound of $\Omega(n^{\lceil (r+1)/2 \rceil} m)$ and an upper bound of $O(n^r(m+r))$ can be proved. Furthermore we demonstrate that techniques which help to improve store and forward algorithms fail in this context and may lead to deadlocks.

In part II we discuss several ideas for offline algorithms for routing on grids. Unfortunately several strategies useful for online algorithms (colouring, exploiting uniaxiality) together with powerful offline methods (like matching to distribute packets uniformly) don't seem to work. Some rescheduling of greedy schedules by introduction of delays seems promising but very costly offline.

### Optimal Depth, Very Small Size Circuits for Symmetric Functions in $AC^0$

by INGO WEGENER (joint work with Johan Håstad, Norbert Wurm and Sang-Zin Yi)

It is well-known which symmetric Boolean functions can be computed by constant depth, polynomial size, unbounded fan-in circuits, i.e. which are contained in the complexity class $AC^0$. This result is sharpened. Symmetric functions in $AC^0$ can be computed by unbounded fan-in circuits with the following properties. If the optimal depth of $AC^0$-circuits is $d$, the depth is almost $d+1$, the number of wires ist almost linear, namely $n \log^{O(1)} n$, and the number of gates is subpolynomial (but superpolylogarithmic), namely $2^{O(\log^\delta n)}$ for some $\delta < 1$. For the monotone symmetric functions, i.e. the threshold functions, our circuits are monotone.

### Complexity Issues in Discrete Neurocomputing

by JURAJ WIEDERMANN

An overview of the basic results in complexity theory of discrete neural computations is presented. Especially, the computational power and efficiency of single neurons, neural circuits, symmetric neural networks (Hopfield model), and of Boltzmann machines is investigated and characterized. Corresponding intractability results are mentioned as well. The evidence is presented why discrete neural networks (inclusively Boltzmann machines) are not to be expected to solve intractable problems more efficiently than other conventional models of computing.

Collected by Rolf Wanka

# E-Mail Adresses


| | |
|---|---|
| Ferri Abolhassan: | ferri@cs.uni-sb.de |
| Richard Anderson: | anderson@cs.washington.edu |
| Elias Dahlhaus: | dahlhaus@theory.informatik.uni-bonn.de |
| Martin Dietzfelbinger: | martind@uni-paderborn.de |
| Torben Hagerup: | torben@cs.uni-sb.de |
| Bernd Halstenberg: | xitibhal@ddathd21.bitnet |
| Ulrich Hertrampf: | hertramp@informatik.uni-wuerzburg.dbp.de |
| Juraj Hromkovic: | juraj@uni-paderborn.de |
| Jörg Keller: | jkeller@cs.uni-sb.de |
| Manfred Kunde: | kunde@informatik.tu-muenchen.de |
| Mirek Kutyłowski: | mirekk@uni-paderborn.de |
| Klaus-Jörn Lange: | lange@informatik.tu-muenchen.de |
| Reinhard Lüling: | rl@uni-paderborn.de |
| Seshu Madhavapeddy: | seshu@uni-paderborn.de |
| Ernst Mayr: | mayr@informatik.uni-frankfurt.de |
| Kurt Mehlhorn: | mehlhorn@cs.uni-sb.de |
| Friedhelm Meyer auf der Heide: | fmadh@uni-paderborn.de |
| Burkhard Monien: | bm@uni-paderborn.de |
| Ian Parberry: | ian@dept.csci.unt.edu |
| Wolfgang Paul: | wjp@cs.uni-sb.de |
| Peter Rossmanith: | rossmani@informatik.tu-muenchen.de |
| Wojciech Rytter: | rytter@plearn.bitnet |
| Ondrej Sýkora: | ondrej@cs.uni-sb.de |
| Thomas Tensi: | tensi@informatik.tu-muenchen.de |
| Imrich Vrťo: | imrich@cs.uni-sb.de |
| Rolf Wanka: | wanka@uni-paderborn.de |
| Ingo Wegener: | wegener@cantor.informatik.uni-dortmund.de |

# Addresses

Ferri Abolhassan
Universität des Saarlandes
Fachbereich Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany

Prof. Dr. Richard Anderson
University of Washington
Dept. of Computer Science & Engineering
Seattle WA 98195
USA

Dr. Elias Dahlhaus
Universität Bonn
Informatik V
Römerstr. 164
W-5300 Bonn 1
Germany

Dr. Martin Dietzfelbinger
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
Warburger Str. 100
W-4790 Paderborn
Germany

Dr. Torben Hagerup
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany

Bernd Halstenberg
TH Darmstadt
Institut für Theoretische Informatik
Alexanderstr. 10
W-6100 Darmstadt
Germany

Dr. Ulrich Hertrampf
Universität Würzburg
Lehrstuhl Theoretische Informatik
Am Hubland
W-8700 Würzburg
Germany

Prof. Dr. Juraj Hromkovič
Universität-GH Paderborn
Fachbereich 17, Mathematik-Informatik
Postfach 1621
W-4790 Paderborn
Germany
Address Institute 2nd:
Comenius University
Department of Computer Science
MFF UK
842 15 Bratislava
ČSFR

Jörg Keller
Universität des Saarlandes
FB Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany

Dr. Manfred Kunde
TU München
Institut für Informatik
Arcisstr. 21
W-8000 München 2
Germany

Dr. Mirosław Kutyłowski
Institute of Computer Science
University of Wrocław
Przesmyckiego 20, 51-151 Wrocław
Poland
Address Institute 2nd:
Universität-GH Paderborn
Fachbereich 17, Mathematik-Informatik
Postfach 1621
W-4790 Paderborn
Germany

Prof. Dr. Klaus-Jörn Lange
TU München
Institut für Informatik
Arcisstr. 21
W-8000 München 2
Germany

Reinhard Lüling
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
Warburger Str. 100
W-4790 Paderborn
Germany

Dr. Seshu Madhavapeddy
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
W-4790 Paderborn
Germany
Address Institute 2nd:
Computer Science Program, MP 3.1
University of Texas at Dallas
Richardson, TX 75083
USA

Prof. Dr. Ernst W. Mayr
Theoret. Inform. Fachbereich Informatik
J.W. Goethe-Universität
Postfach 111 932
Robert-Mayer-Str. 11-15
W-6000 Frankfurt a. M. 11
Germany

Prof. Dr. Kurt Mehlhorn
Max-Planck-Institut
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany

Prof. Dr. Friedhelm Meyer auf der Heide
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
Postfach 1621
W-4790 Paderborn
Germany

Prof. Dr. Burkhard Monien
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
Warburger Str. 100
W-4790 Paderborn
Germany

Prof. Dr. Ian Parberry
Dept. of Computer Science
University of North Texas
P.O. Box 13886
Denton, TX 76203-3886
USA

Wolgang J. Paul
Universität des Saarlandes
Fachbereich Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany

Peter Rossmanith
Technische Universität München
Institut für Informatik
Arcisstr. 21
W-8000 München 2
Germany

Prof. Dr. W. Rytter
Institute of Informatics
Warsaw University
00-913 Warszawa 59
ul. Banacha 2
Poland

Dr. Ondrej Sýkora
z.Zt. Max-Planck-Institut
bei Prof. Dr. Mehlhorn
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany
Adress Institute 2nd:
Computing Centre
Slovak Academy of Sciences
Bratislava
Dubravska 9
842 35 Bratislava
Czechoslovakia

Thomas Tensi
TU München
Institut für Informatik
Arcisstr. 21
W-8000 München 2
Germany

Dr. Imrich Vrťo
z.Zt. Max-Planck-Institut
bei Prof. Dr. Mehlhorn
Im Stadtwald 15
W-6600 Saarbrücken
Germany
Address Institute 2nd:
Computing Centre
Slovak Academy of Sciences
Dubravska 9
842 35 Bratislava
Czechoslovakia

Rolf Wanka
Fachbereich 17, Mathematik-Informatik
Universität-GH Paderborn
W-4790 Paderborn
Germany

Prof. Dr. Ingo Wegener
Universität Dortmund
Informatik II
Postfach 500 500
W-4600 Dortmund 50
Germany

Prof. Dr. Juraj Wiedermann
Institute of Socio-Economic Information
and Automation in Management
842 21 Bratislava
Dubravska 3
ČSFR

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):
Architecture and Performance, Dagstuhl-Seminar-Report; 1, 18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):
Tree Adjoining Grammars, 1st. International Worshop on TAGs: Formal Theory and Application, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):
Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3, 3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):
Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

J. Berstel , J.E. Pin, W. Thomas (editors):
Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):
Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991 (9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):
Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):
Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10, 18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):
Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):
Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, , 20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):
Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):
Query Processing in Object-Oriented, Complex Object, and Nested Relation Databases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Report; 16, 24.-28.6.1991 (9126)