04091 Abstracts Collection
Data Structures
— Dagstuhl Seminar —

S. Albers[1], R. Sedgewick[2] and D. Wagner[3]

[1] Univ. Freiburg, DE
salbers@informatik.uni-freiburg.de
[2] Princeton, US
rs@cs.princeton.edu
[3] Univ. Karlsruhe, DE
dwagner@ira.uka.de

Abstract. From 22.02. to 27.02.2004, the Dagstuhl Seminar 04091 "Data Structures" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar are put together in this paper. The first section describes the seminar topics and goals in general.

Keywords. Cache oblivious algorithms; cell probe model; computational geometry; data compression; dictionaries; finger search; hashing; heaps; I/O efficiency; lower bounds.

Seminar Summary

The design and analysis of algorithms is a fundamental area in computer science. This also involves the development of suitable methods for structuring the data to be manipulated by these algorithms. Hence, algorithms and data structures form a unit, and the right choice of algorithms and data structures is a crucial step in the solution of many problems. For this reason, the design, analysis and implementation of data structures form a classical field of computer science, both in research and teaching.

The Dagstuhl Seminar on Data Structures in 2004 reported on ongoing research on classical data structuring problems as well as classical application areas such a computational geometry. Furthermore many contributions investigated new algorithmic and data structuring problems arising in large networks or in the maintenance of large data sets. As in previous meeting, there was some shift of interest away from theory, e.g., the classical analysis of asymptotic behavior of algorithms, to more practical issues, such as implementation problems and the usefulness of algorithms in practical applications. This is motivated by the

Dagstuhl Seminar Proceedings 04091
Data Structures
http://drops.dagstuhl.de/opus/volltexte/2005/175

fact that more and more researchers in computer science also want to make their results available in form of programs or software packages.

With 48 participants, the attendence was even higher than in previous meetingson the topic. In addition to scientific talks, there were fruitful and stimulating discussions throughout the meeting. We thank the team of Schloss Dagstuhl for their hospitality and support making this successful workshop possible.

Joint work of: S. Albers, R. Sedgewick, D. Wagner

The Cell Probe Complexity of Succinct Data Structures

Peter Bro Miltersen (Aarhus University, DK)

We prove lower bounds on the redundancy-query time tradeoff for succinct data structures in the cell probe model.

Joint work of: Gal, Anna; Bro Miltersen, Peter;

RAMBO Processor

Andrej Brodnik (IMFM Ljubiljana, SLO)

Fredman and Saks in 1988 defined a new machine model called RAMBO – RAM with Bytes Overlapping. In this model the memory consists of words, which, however, can share bits; i.e. the same bit can appear in several words which consequently means that if we change that bit one word, the content of some word is also changed. Brodnik in 1995 presented a real example of use of such memory. The memory was also implemented in hardware in 2000 by Brodnik et. al. and used to solve priority queue problem in a constant time.

In this talk we use RAMBO in two separate occasions. First we approach interval based problems. These are problems, where we store integer values in an array. The operations on these values are an update of the value in the array (change(i, delta)) and query on the values of the array in a portion of the array (query(i, j) –> value). We study two versions of the query. First we look at a query "sum" which returns a partial sum of the values in the interval. The addition forms a group. We show how to solve this problem in O(log n) time on standard cell probe model, which is also optimal (Demaine 04). We also show how to solve the problem on RAMBO in O(1) time. We use an Yggdrasil topology of RAMBO.

The second problem we study is a query "max". This operatoin does not form a group, but a monoid. We also show how to solve this problem in an optimal O(log n) time, but we can not find an O(1) solution for RAMBO. It remains an open problem if one exists. A hypothesis is that it does not.

In the second part of the talk we show how to use RAMBO as a processor which permits us to perform bitwise operation on a word: Boolean operations, shifts and rotations. All of them in constant time. Here we use several topologies of RAMBO: Twin, Circle, Line, and Tail.

Keywords: Data structures, RAMBO, machine model, partial sum

Joint work of: Brodnik, Andrej; Karlsson, Johan; Nilsson, Andreas;

Applications and Hardness of Link-Eval on a Path

Adam Buchsbaum (AT&T Research - Florham Park, USA)

We study the special case of the link-eval data structure operations [Tarjan, 1979] applied to simple paths. This case arises in linear-time algorithms for minimum spanning tree (MST) verification and computing dominators in flowgraphs. We show special cases for which this application of link-eval can be performed in linear time (on RAMs and pointer machines), thereby fixing the linear-time pointer machine algorithms for MST verification and dominators previously suggested by Buchsbaum et al. [1998]. The dominators resolution also depends on results by Georgiadis and Tarjan [2004].

Joint work of: Buchsbaum, Adam

Logarithmic Lower Bounds in the Cell-Probe Model

Erik Demaine (MIT - Cambridge, USA)

We introduce a new information-theoretic technique for proving amortized lower bounds on data structures in the cell-probe model. In particular, this technique allows us to prove the first logarithmic lower bounds in this model, proving the optimality of e.g. Sleator and Tarjan's link-cut trees. We also obtain tight bounds on the partial-sums problem even under various parameterizations. These results appear in SODA 2004 and STOC 2004.

Joint work of: Demaine, Erik; Patrascu, Mihai

Oracles for distances avoiding a link failure

Camil Demetrescu (Università di Roma I, I)

We consider the problem of preprocessing an edge-weighted directed graph G to answer queries that ask for the shortest distance from any given node x to any other node y avoiding an arbitrary failed node or link. We describe an oracle (i.e, a simple data structure) for such queries that can be stored in $O(n^2 logn)$ space, and which allows queries to be answered in $O(1)$ time, where n is the number of nodes in G.

Keywords: DISTANCES, LINK FAILURES, SHORTEST PATHS

Joint work of: Chowdhury, R. A.; Demetrescu, Camil; Ramachandran, Vijaya; Thorup, Mikkel;

Cache-Friendly Dictionary Implementations with Constant Lookup Time and Small Space Overhead

Martin Dietzfelbinger (TU Ilmenau, D)

We study implementations of dictionaries (insertions, deletions, lookups) for keys from a universe U in a hash table.

Generalizing the cuckoo hashing scheme of Pagh and Rodler (2001) in a way different from the “d-ary cuckoo hashing” approach of Fotakis et al. (2003) we propose using a table that consists of $m = (1+\epsilon)n/d$ buckets, each of which can store up to d words from U .

Two hash functions h_1 and h_2 are suitable for a set S of n keys if each key x from S can be stored in bucket $B_{h_1(x)}$ or in bucket $B_{h_2(x)}$ without any bucket overflowing.

Lookups take time $O(d)$; only two buckets have to be accessed: this makes the scheme well suited for cache architectures.

We show that an arbitrarily small space overhead $\epsilon > 0$ can be achieved with $d = O(log(1/\epsilon))$.

Keywords: Dictionaries, Hashing, Cuckoo Hashing, Cache-oriented, Worst-Case Constant Lookup Time, Simulation of Uniform Hashing

Joint work of: Dietzfelbinger, Martin; Weidling, Christoph;

Wavelength Conversion in Optical Networks with Shortest-Path Routing

Thomas Erlebach (ETH Zürich, CH)

We consider all-optical networks with shortest-path routing that use wavelength-division multiplexing and employ wavelength conversion at specific nodes in order to maximize their capacity usage.

We present efficient algorithms for deciding whether a placement of wavelength converters allows the network to run at maximum capacity, and for finding an optimal wavelength assignment when such a placement of converters is known. Our algorithms apply to both undirected and directed networks. Furthermore, we show that the problem of designing such networks, i.e., finding an optimal placement of converters, is MAXSNP-hard in both the undirected and the directed case.

Finally, we give a linear-time algorithm for finding an optimal placement of converters in undirected triangle-free networks, and show that the problem remains NP-hard in bidirected triangle-free planar networks.

Joint work of: Erlebach, Thomas; Stefanakos, Stamatis;

The Cost of Cache-Oblivious Searching

Rolf Fagerberg (Aarhus University, DK)

Tight bounds on the cost of cache-oblivious searching are proved.

It is shown that no cache-oblivious search structure can guarantee that a search performs fewer than $\lg e \log_B N$ block transfers between any two levels of the memory hierarchy. This lower bound holds even if all of the block sizes are limited to be powers of 2. A modified version of the van Emde Boas layout is proposed, whose expected block transfers between any two levels of the memory hierarchy arbitrarily close to $[\lg e + O(\lg\lg B/\lg B)]\log_B N + O(1)$. This factor approaches $\lg e \approx 1.443$ as B increases. The expectation is taken over the random placement of the first element of the structure in memory.

As searching in the Disk Access Model (DAM) can be performed in $\log_B N+1$ block transfers, this result shows a separation between the 2-level DAM and cache-oblivious memory-hierarchy models. By extending the DAM model to k levels, multilevel memory hierarchies can be modelled. It is shown that as k grows, the search costs of the optimal k-level DAM search structure and of the optimal cache-oblivious search structure rapidly converge.

This demonstrates that for a multilevel memory hierarchy, a simple cache-oblivious structure almost replicates the performance of an optimal parameterized k-level DAM structure.

Joint work of: Bender, Michael A.; Stølting Brodal, Gerth; Fagerberg, Rolf; Ge, Dongdong; He, Simai; Hu, Haodong; Iacono, John; Lopez-Ortiz, Alejandro;

Compression boosting in optimal linear time using the Burrows-Wheeler Trasform

Paolo Ferragina (Università di Pisa, I)

We provide a general boosting technique for Textual Data Compression. Qualitatively, it takes a good compression algorithm and turns it into an algorithm with a better compression performance guarantee. It displays the following remarkable properties: (a) it can turn *any memoryless* compressor into a compression algorithm that uses the "best possible" contexts; (b) it is very simple and *optimal* in terms of time; (c) it admits a decompression algorithm again optimal in time.

To the best of our knowledge, this is the first boosting technique displaying these properties.

Technically, our boosting technique builds upon three main ingredients: the Burrows-Wheeler Transform, the Suffix Tree data structure, and a greedy algorithm to process them. Specifically we show that there exists a proper partition of the Burrows-Wheeler Transform of a string s that shows a deep combinatorial relation with the k-th order entropy of s. That partition can be identified via a greedy processing of the suffix tree of s with the aim of minimizing a proper objective function over its nodes. The final compressed string is then obtained by compressing individually each substring of the partition by means of the base compressor we wish to boost.

Our boosting technique is inherently combinatorial because it does not need to assume any prior probabilistic model about the source emitting s, and it does not deploy any training, parameter estimation and learning. Various corollaries are derived from this main achievement. Among the others, we show analytically that using our booster we get better compression algorithms than some of the best existing ones, i.e., LZ77, LZ78, PPMC and the ones derived from the Burrows-Wheeler Transform. Further, we settle analytically some long standing open problems about the algorithmic structure and the performance of BWT-based compressors. Namely, we provide the first family of BWT algorithms that do not use Move-To-Front or Symbol Ranking as a part of the compression process.

Heaps with Mass Deletions

Martin Fürer (Pennsylvania State University, USA)

It is shown that the cost for an uninterrupted sequence of q deletions in a Fibonacci heap of size n is only $O(q \log \frac{n}{q})$. Here, "uninterrupted" means that no element that is compared during this sequence, is also subject to a Delete operation. Note that a Delete-Min operation does not interrupt a sequence.

As a consequence, a much simplified $O(|E| \log \log^* |V|)$ time Minimum Spanning Tree algorithm is obtained.

Keywords: MST, Minimum weight spanning trees, Fibonacci heaps

An approximation algorithm for MAX2SAT with cardinality constraint

Thomas Hofmeister (Universität Dortmund, D)

We consider the MAX2SAT problem with cardinality constraint, i.e., the assignment one is looking for is required to contain a certain number of ones. We show how one can obtain an approximation algorithm with a ratio of 3/4.

Retroactive Data Structures

John Iacono (Polytechnic Univ. - New York, USA)

We introduce a new data structuring paradigm in which operations can be performed on a data structure not only in the present but also in the past. In this new paradigm, called retroactive data structures, the historical sequence of operations performed on the data structure is not fixed. The data structure allows arbitrary insertion and deletion of operations at arbitrary times, subject only to consistency requirements. We initiate the study of retroactive data structures by formally defining the model and its variants. We prove that, unlike persistence, efficient retroactivity is not always achievable, so we go on to present several specific retroactive data structures.

Joint work of: Demaine, Erik; Iacono, John; Langerman, Stefan;

Geometric Data Structures with Sub-linear Space

Piotr Indyk (MIT - Cambridge, USA)

The vast majority of known data structures use space at least linear in the number of objects stored. Are there any interesting problems that can be solved in sub-linear space ? In this talk we present several new results of this type, in the context of geometric data. Our data structures support insertions and deletions of points, as well as (approximate) queries about geometric properties of the data, such as the value of the minimum cost tree spanning the points, the value of the minimum weight matching of the points, etc.

To appear in STOC'04.

On the OVSF CodeAssignment Problem

Riko Jacob (ETH Zürich, CH)

Orthogonal variable-spreading-factor (OVSF) codes can be used to distinguish the transmissions from different users in a wireless network. Users request and release codes dynamically, and we want to maintain an assignment of codes to active users that keeps the number of reassignments minimal. This problem can be viewed as maintaining a subset of the nodes of a given complete binary tree of height h under insertions (on a specified level) and deletions.

We give an algorithm that needs at most h reassignments per operation. For the case where at most half the capacity of the tree is in use at any time, we give an algorithm that needs an amortized constant number of reassignments.

We also consider the algorithmic problem of minimizing the number of reassignments for a given situation and a single insertion.

Joint work of: Erlebach, Thomas; Jacob, Riko; Mihalák, Matús; Nunkesser, Marc; Szabó, Gábor; Widmayer, Peter;

On Rectangle Packing: Maximizing Benefits

Klaus Jansen (Universität Kiel, D)

We consider the following rectangle packing problem: Given a set of rectangles, each of which is associated with a profit, we are requested to pack a subset of the rectangles into a bigger rectangle to maximize the total profit of rectangles packed. The rectangles may not overlap and may or may not be rotated. This problem is strongly NP-hard even for packing squares with identical profits. A simple $(3 + \epsilon)$ - approximation algorithm is presented. We further improve the algorithm by showing a worst-case ratio of at most $5/2 + \epsilon$. Finally we devise a $(2+\epsilon)$-approximation algorithm. A number of restricted cases are also considered.

Keywords: Packing, rectangles, approximation algorithms

Joint work of: Jansen, K.; Zhang, G.

Data Structures for Text Editors

Alejandro Lopez-Ortiz (University of Waterloo, CDN)

We consider the problem of finding an efficient data structure for text manipulation and representation supporting

(a) word wrap on fixed length lines with soft breaks, and

(b) the operations

- cursor positioning in time O(log m) where m is the distance in characters from the current position to the new position (in a finger like manner)

- delete, insert, display & cut and paste in O(log n) time where n is the size of the text in characters

We introduce Table Trees, a data structure that supports the operations in the given time under the natural assumption that the length of the line is a constant (usually 80 or 132 characters in most text editing settings).

Joint work of: Lopez-Ortiz, Alejandro; Demaine, Erik; Hagerup, Torben;

Scheduling against an adversarial network

Friedhelm Meyer auf der Heide (Universität Paderborn, D)

Using idle times of the processors of a LAN, a WAN, or the Internet is a well-known approach to run coarse grained parallel algorithms for extremely complex problems. We present on-line algorithms for scheduling the processes of a parallel application on a dynamic network in which the idle times of the processors are dictated by an adversary. We also take communication and synchronization costs into account.

Our first contribution consists of a formal model to restrict the adversary in a reasonable way. We then show a constant factor approximation for the off-line scheduling problem. As this problem has to take communication cost into account, it can be seen as a generalization of many NP-hard parallel machine scheduling problems. Finally, we present on-line algorithms for different models with constant or with "nearly constant" competitive ratio.

Keywords: Dynamic networks, On-line algorithms, parallel computing, approximation algorithms

Joint work of: Meyer auf der Heide, Friedhelm; Leonardi, Stefano; Marchetti-Spaccamela, Alberto;

I/O-Efficient Undirected Shortest Paths

Ulrich Carsten Meyer (MPI für Informatik, D)

We present an I/O-efficient algorithm for the single-source shortest path problem on undirected graphs $G = (V, E)$. Our algorithm performs $O(\sqrt{(VE/B)\log_2(W/w)} + \mathrm{sort}(V + E)\log\log(VB/E))$ I/Os[1], where $w \in \mathbb{R}^+$ and $W \in \mathbb{R}^+$ are the minimal and maximal edge weights in G, respectively. For uniform random edge weights in $(0, 1]$, the expected I/O-complexity of our algorithm is $O(\sqrt{VE/B} + ((V + E)/B)\log_2 B + \mathrm{sort}(V + E))$.

Joint work of: Meyer, Ulrich Carsten; Zeh, Norbert;

[1] $\mathrm{sort}(N) = \Theta((N/B)\log_{M/B}(N/B))$ is the I/O-complexity of sorting N data items.

Nearly Linear Time Minimum Spanning Tree Maintenance for Transient Node Failures

Guido Proietti (Univ. degli Studi di L'Aquila, I)

Given a 2-node connected, real weighted, and undirected graph $G = (V, E)$, with n nodes and m edges, and given a minimum spanning tree (MST) $T = (V, E_T)$ of G, we study the problem of finding, for every node $v \in V$, a set of replacement edges which can be used for constructing an MST of $G - v$ (i.e., the graph G deprived of v and all its incident edges). We show that this problem can be solved on a pointer machine in $\mathcal{O}(m \cdot \alpha(m, n))$ time and $\mathcal{O}(m)$ space, where α is the functional inverse of the Ackermann's function.

Our solution improves over the previously best known $\mathcal{O}(\min\{m \cdot \alpha(n, n), m + n \log n\})$ time bound, and allows to close the gap existing with the fastest solution for the edge-removal version of the problem (i.e., that of finding, for every edge $e \in E_T$, a replacement edge which can be used for constructing an MST of $G - e = (V, E \setminus \{e\})$).

Our algorithm finds immediate application in maintaining MST-based communication networks undergoing temporary node failures.

Moreover, in a distributed environment in which nodes are managed by selfish agents, it can be used to design an efficient, truthful mechanism for building an MST.

Keywords: Graph Algorithms, Minimum Spanning Tree, Transient Node Failures, Fault Tolerance, Algorithmic Mechanism Design.

Joint work of: Nardelli, Enrico; Proietti, Guido; Widmayer, Peter;

Distance-preserving approximations of polygonal paths

Michiel Smid (Carleton Univ. - Ottawa, CDN)

Given a polygonal path P with vertices $p_1, p_2, \ldots, p_n$ and a real number $t \geq 1$, a path $Q = (p_{i_1}, p_{i_2}, \ldots, p_{i_k})$ is a t-distance-preserving approximation of P if $1 = i_1 < i_2 < \ldots < i_k = n$ and each straight-line edge $(p_{i_j}, p_{i_{j+1}})$ of Q approximates the distance between p_{i_j} and $p_{i_{j+1}}$ along the path P within a factor of t.

We present exact and approximation algorithms that compute such a path Q that minimizes k (when given t) or t (when given k).

Keywords: Computational geometry, polygonal path, approximation

Joint work of: Smid, Michiel; Gudmundsson, Joachim; Narasimhan, Giri;

Space-Efficient Geometric Divide-and-Conquer Algorithms

Jan Vahrenhold (Universität Münster, D)

We present an approach to simulate geometric divide-and-conquer algorithms ina space-effcient way, and illustrate it by giving space-effcient algorithms for the closest-pair, bichromatic closest-pair, and all-nearest neighbors problems.

Joint work of: Bose, Prosenjit; Maheshwari, Anil; Morin, Pat; Morrison, Jason; Smid, Michiel; Vahrenhold, Jan;

Geometric Containers for Shortest Paths

Dorothea Wagner (Universität Karlsruhe, D)

In talk, we consider Dijkstra's algorithm for the single source single target shortest path problem in large sparse graphs. The goal is to reduce the response time for on-line queries by using precomputed information. Due to the size of the graph, preprocessing space requirements can be only linear in the number of nodes. We assume that a layout of the graph is given. In the preprocessing, we determine from this layout a geometric object for each edge containing all nodes that can be reached by a shortest path starting with that edge. Based on these geometric objects, the search space for on-line computation can be reduced significantly. Shortest path queries can then be answered by Dijkstra's algorithm restricted to edges where the corresponding geometric object contains the target.

We present an extensive experimental study comparing the impact of different types of objects. The test data we use are real-world traffic networks, the typical field of application for this scenario.

Furthermore, we present new algorithms as well as an empirical study for the dynamic case of this problem, where edge weights are subject to change and the geometric containers have to be updated. We evaluate the quality and the time for different update strategies that guarantee correct shortest paths.

Part of this work was presented at ESA 2003 and at ATMOS 2003.

Keywords: Shortest paths, online computation, speed-up techniques, route planning, time table information

Joint work of: Wagner, Dorothea; Willhalm, Thomas; Zaroliagis, Christos;

Blind Search for Maximum Matchings

Ingo Wegener (Universität Dortmund, D)

There is a growing interest in the analysis of the behavior of randomized search heuristics like randomized local search, simulated annealing and evolutionary algorithms on combinatorial optimization problems. This research direction is motivated by the many applications of these heuristics in real-life optimization.

The common idea is to understand why these heuristics are often quite successful.

This is done by the analysis of the expected optimization time either for the worst-case instance, for random instances, or for semi-random instances. Here, the behavior of randomized local search and a simple and fundamental evolutionary algorithm on the well-known maximum matching problem is analyzed.

Both are polynomial-time randomized approximation schemes although they do not employ the idea of augmenting paths. They find a maximum matching in expected polynomial time for simple graphs but their exist bipartite graphs with a degree bounded by 3 where the expected optimization time grows exponentially.

Keywords: Evolutionary Algorithms, Randomized Local Search, Maximum Matchins, Polynomial-time Ranomized Approximation Scheme.

Data Structures for Polygonal Path Approximation

Alexander Wolff (Universität Karlsruhe, D)

We solve the problem of approximating polygonal paths by reducing it in three steps to simpler and simpler geometric farthest-point problems. The simplest of these is the problem Farthest-Point-Left-of-Line-in-Convex-Polygon (FPLLCP) that asks to preprocess a convex polygon C of n points for queries of the following type: Given a directed line ℓ and a point $q \in \ell$, find the point farthest from q to the left of ℓ. Our data structure for this problem takes $O(n \log n)$ preprocessing time, $O(n \log n)$ storage and $O(\log^2 n)$ query time.

This results in an output-sensitive algorithm that approximates a polygonal path $P = (p_1, \ldots, p_n)$ in $O(F(m) n \log^3 n)$ time using $O(n \log^2 n)$ storage, where m is the number of vertices on an optimal approximating path and $F(m) \in O(n)$ is the number of vertices that can be reached from p_1 with at most m approximating segments.

Keywords: Farthest-point queries, polygonal path approximation

Joint work of: Wolff, Alexander; Daescu, Ovidiu; Mi, Ningfang; Shin, Chan-Su;

Improved Bounds for Finger Search on a RAM

Christos Zaroliagis (University of Patras, GR)

We present a new finger search tree with $O(1)$ worst-case update time and $O(loglogd)$ expected search time with high probability in the Random Access Machine (RAM) model of computation for a large class of input distributions.

The parameter d represents the number of elements (distance) between the search element and an element pointed to by a finger, in a finger search tree that stores n elements.

Our data structure improves upon a previous result by Andersson and Mattson that exhibits expected $O(loglogn)$ time and $O(1)$ worst-case update time, by incorporating the distance d into the search time complexity, thus removing the dependence on n, and by proving that the attained search time complexity holds with high probability. For the need of the analysis we model the updates by a “balls and binscombinatorial game that is interesting in its own right as it involves insertions and deletions of balls according to an unknown distribution.

Joint work of: Kaporis, Alexis; Makris, Christos; Sioutas, Spyros; Tsakalidis, Athanasios; Tsichlas, Kostas; Zaroliagis, Christos;

New Hashing Primitives and Applications in Data Structures

Rasmus Östlin Pagh (The IT University of Copenhagen, DK)

We survey two recent constructions of hash functions:

1. A hash function that, on any set of n inputs, behaves like a truly random function with high probability, can be evaluated in constant time on a RAM, and can be stored in O(n) words, which is optimal.

2. A way of maintaining a perfect hash function for a data stream of n distinct w-bit integers. The space usage is O(n log w) bits, and the range of the perfect hash function has size n+o(n).

Keywords: Hashing

Joint work of: Pagh, Rasmus; Pagh, Anna; Brodal, Gerth; Worm Mortensen, Christian;