

Report of the Dagstuhl Seminar on

Data Structures

February, 28th – March, 4th 1994

Abstract

The second Dagstuhl Seminar on Data Structures was organized by Prof. H. Noltemeier (Universität Würzburg, FRG), Prof. Th. Ottmann (Universität Freiburg, FRG) and Prof. D. Wood (University of Western Ontario, Canada). It was held during the week of February, 28th until March, 4th 1994.

The seminar brought together 23 participants which came from nine countries with a particular strong group from the Scandinavian countries. Almost all participants were prepared and willing to give a presentation. So 21 scientific talks and one non-scientific talk about the state of the field of data structures in general were held during that week. The abstracts of the scientific talks can be found in this report.

Naturally most of the talks focussed on the classic field of *Data Structures*, though related topics such as *String Matching Algorithms*, *Graph Algorithms* and *Computational Geometry* were also included. A number of impressing and exiting new results were presented. The seminar at Schloß Dagstuhl offered ample opportunity for personal discussions and the exchange of new ideas between the participating researchers.

All participants appreciated the stimulating atmosphere and outstanding organisation of Schloß Dagstuhl which greatly contributed to the success of this seminar.

Contents

<i>Arne Andersson (joint work with Stefan Nilsson)</i> Forward Radix Sort with Applications	6
<i>Norbert Blum</i> Speeding up Dynamic Programming without Omitting any Optimal Solution	7
<i>Svante Carlsson (joint work with J. Chen)</i> On Partial Order Production	13
<i>Martin Dietzfelbinger (joint work with Y. Gil, Y. Matias, N. Pippenger)</i> Polynomial Hash Functions are Reliable.....	13
<i>Vladimir Estivill-Castro (joint work with D. Wood)</i> External Sorting and Nearly Sortedness	10
<i>Shimon Even (joint work with A. Litman, G. Even and E. Dinitz)</i> Layered Cross Product of Graphs and its Applications.....	7
<i>Martin Fürer</i> Approximating the Maximum Independent Set Problem in Bounded Degree Graphs	15
<i>Thorsten Graf (joint work with K. Hinrichs)</i> Direct Computations of Nearest Foreign Neighbors in Arbitrary L^t -Metrics.....	11
<i>Michael Kaufmann (joint work with U. Foessmeier, A. Zelikovsky, P. Berman, M. Karpinski)</i> Approximations for the Rectilinear Steiner Tree Problem	9
<i>Christos Levcopoulos (joint work with A. Lingas)</i> The Greedy Triangulation Approximates the Minimum Weight Triangulation and can be Computed in Linear Time in the Average Case	15
<i>Heikki Mannila (joint work with Th. Eiter)</i> Theory Distances and Their Computation.....	12
<i>Ian Munro (joint work with A. Brodnik)</i> Membership Queries in Constant Time and (Order of) Minimum Storage	13

<i>Stefan Näher (joint work with K. Mehlhorn)</i>	
A Robust Implementation of the Bentley/Ottmann Plane Sweep Algorithm	13
<i>Thomas Ottmann (joint work with S. Schuierer, S. Soundaralakshmi)</i>	
Computation of Extreme Points in d Dimensions	6
<i>Mireille Règnier (joint work with L. Rostami)</i>	
Minimal 2-Dimensional Periodicities and Maximal Space Coverings ...	14
<i>Sven Schuierer</i>	
Computing the L_1 -Diameter and Center of a Simple Rectilinear Polygon in Parallel	11
<i>Michiel Smid</i>	
An Efficient Solution for the L_∞ -Post-Office Problem	10
<i>Kokichi Sugihara</i>	
Combinatorial Abstraction for Numerically Robust Geometric Algorithms	8
<i>Esko Ukkonen (joint work with J. Kärkkäinen)</i>	
Two Dimensional Pattern Matching in Optimal Expected Time	9
<i>Ingo Wegener (joint work with B. Bullig, M. Sauerhoff, and D. Sieling)</i>	
An Efficient Data Structure for Boolean Functions: k OBDDs	6
<i>Derick Wood (joint work with D. Raymond)</i>	
Grail 2.0	16

Abstracts

An Efficient Data Structure for Boolean Functions: k OBDDs

by Ingo Wegener, Universität Dortmund, (joint work with Beate Bollig, Martin Sauerhoff, and Detlef Sieking)

Data Structures for Boolean functions are used in many applications like symbolic verification, logical synthesis, fast pattern generation, or analogous or sequential circuits. Efficient algorithms should exist for important applications like the satisfiability test, the synthesis problem, or the equality test. Moreover, many (important) Boolean functions should be representable in (small) polynomial size and the representation should be efficiently computable from circuits. The state of the art data structures is the OBDD (ordered binary decision diagram). We consider k layers of OBDDs. More functions can be represented in polynomial space, also important functions. In spite of the existence of null chains, all operations except the minimization problem can be performed in polynomial time. Also experimental results are reported.

Forward Radix Sort with Applications

by Arne Andersson, Lund University, (joint work with Stefan Nilsson)

We present new efficient algorithms for forward radix sort. As an application, we show how to improve the tradeoff between entropy and sorting cost, reported by Chen and Reif at FOCS '93.

In many situations, our algorithms will run in time proportional to the time required to verify that the set is sorted.

Computation of Extreme Points in d Dimensions

by Thomas Ottmann, (joint work with S. Schuierer, S. Soundaralakshmi)

Given a set P of n points in d dimensions. The set of extreme points of P is the minimal subset E of P with $\text{convex-hull}(E) = \text{convex-hull}(P)$. We consider the problem of enumerating all the extreme points. An algorithm

is presented which enumerates the extreme points with $O(n)$ space and $O(nm)$ time where m is the number of extreme points. This improves previously known algorithms, if the number of extreme points is small. The basic techniques used in the algorithm is (1) translating the test whether a point is extreme into dual space, and (2) to solve an LP-problem in fixed dimensions d . As a corollary we also show that the depth of all n points in d dimensions can be computed in time $O(n^2)$.

Layered Cross Product of Graphs and its Applications

by Shimon Even, University of Haifa, (joint work with Ami Litman, Guy Even and Efim Dinitz)

Layered Cross Product, LCP, of layered graphs is defined. Several well known interconnection networks are LCP's of simple graphs, such as trees. Some important properties of these networks are shown to be trivial consequences.

LCP, combined with elementary techniques of descriptive geometry, is used to derive aesthetic rectilinear layouts of some networks (such as the Butterfly network and the Mesh of Trees).

Speeding up Dynamic Programming without Omitting any Optimal Solution

by Norbert Blum, Universität Bonn

We consider a special case of dynamic programming where the subproblems can be considered as points in an $(m + 1) \times (n + 1)$ matrix and the minimal cost $v(i, j)$ of a solution for subproblem (i, j) is obtained by

$$v(i, j) = \min \begin{cases} \min_{0 \leq k \leq j} \{v(i, k) + c([i, k], [i, j])\} \\ \min_{0 \leq l \leq i} \{v(l, j) + c([l, j], [i, j])\} \\ v(i - 1, j - 1) + c([i - 1, j - 1], [i, j]) \end{cases}$$

If we store the edge $([i', j'], [i, j])$ if and only if $v(i, j) = v(i', j') + c([i', j'], [i, j])$ then we obtain the DP-graph which contains all optimal solutions in the following sense. There is a one-to-one correspondence between the paths from the node $[0, 0]$ to the node $[m, n]$ in the DP-graph

and the optimal solutions. If $c([i', j'], [i, j])$ can always be computed in constant time then it is obvious that the DP-graph can be computed in $O(m^2n + mn^2)$ time. In the case that the cost function c is concave, Galil and Gioncorto and independently Meyers and Miller have shown how to compute one optimal solution by speeding up dynamic programming. But they do not compute all optimal solutions. We show how to extend the method such that a compact representation of the DP-graph can be computed without increasing the time by more than a constant factor. Note that the DP-graph can contain $\Omega(m^2n + mn^2)$ edges.

Combinatorial Abstraction for Numerically Robust Geometric Algorithms

by Kokichi Sugihara, University of Tokyo

A new approach, called combinatorial abstraction, to the design of geometric algorithms robust against numerical errors is presented with an example. In this approach, we first choose combinatorial properties that should be satisfied by the solution, and next describe the basic part of the algorithm only in terms of combinatorial computation in such a way that the chosen properties are always guaranteed. This algorithm has ambiguity in choosing branches of the processing. So, we use numerical computation as secondary-priority information to choose the most promising branch of the processing. The algorithm designed in this way has many good properties. First, it is completely robust in the sense that it never comes across inconsistency no matter how large numerical errors take place. Secondly, the output is consistent at least in the combinatorial sense. Thirdly, the output converges to the true solution as the precision in computation becomes higher. Forthly, the structure of the algorithm is simple, because we need not consider degenerate cases. Fifthly, the design of the algorithm is also simple, because we need no numerical error analysis; the inconsistency issue is completely separated from error-evaluation issue.

Two Dimensional Pattern Matching in Optimal Expected Time

by *Esko Ukkonen, University of Helsinki, (joint work with Juna Kärkkäinen)*

Algorithms with optimal expected running time were presented for searching the occurrences of a two dimensional $m \times m$ pattern P in a two-dimensional $n \times n$ text T over an alphabet of size c . The algorithms are based on placing in the text a static grid of test points, determined only by n, m and c (not dynamically by earlier test results). Using test strings read from the test points the algorithms eliminate as many potential occurrences of P as possible. The remaining potential occurrences are separately checked for actual occurrences. By suitably choosing the test point set, we obtained algorithms with expected running time

$$O\left(\frac{n^2}{m^2} \log_c m^2\right)$$

which is optimal by a lower bound result by Yao (we assume the Bernoulli's model of randomness). The method was generalized for the k mismatches problem. The resulting algorithm has expected running time

$$O\left(\frac{n^2}{m^2}(k+1) \log_c m^2\right),$$

provided that $k \leq m^2/(4\lceil \log_c m^2 \rceil)$. All algorithms need preprocessing of P which takes time and space $O(m^2 \log_c m^2)$ or $O(cm^2)$.

Approximations for the Rectilinear Steiner Tree Problem

by *Michael Kaufmann, Universität Tübingen, (joint work with Uli Fößmeier, Alex Zelikovsky, Piotr Berman, Marek Karpinski)*

The rectilinear Steiner tree problem requires a shortest tree spanning a given vertex subset in the plane with rectilinear distance. Zelikovsky's and Berman/Ramayer's approaches are the only with provably better approximation factors than the minimum spanning tree method by Hwang. We give efficient implementations of their algorithms improving their time bounds from $O(n^3)$ to $O(n \log^2 n)$ for an ϵ -approximation to their $\frac{11}{8}$ -approximation factor.

Furthermore we improve their quality bounds from 1.375 to 1.271 and give even for the improved approaches efficient algorithms. A lower bound example has been found where Zelikovsky's approach fails to be better than factor 1.3.

External Sorting an Nearly Sortedness

by Vladimir Estivill-Castro, LANIA, Xalapa, Vera Cruz, Mexico, (joint work with D. Wood, University of Ontario)

The availability of large main memories and the new technologies for disk drives have modified the models for external sorting and have renewed interest in their study. Little is known about the performance of traditional and more recent sorting methods on nearly sorted files although such files are common in practice.

- We demonstrate that, during the merging phase, the floating-buffers technique not only reduces the sorting time by fully overlapping I/O and saving seeks, but also it profits significantly from existing order in the input.
- We propose a new algorithm for computing the *consumption sequence* for floating buffers that improves upon previous algorithms.

An Efficient Solution for the L_∞ -Post-Office Problem

by Michiel Smid, MPI Saarbrücken

Let S be a set of n points in D -dimensional space. It is shown that range trees can be used for solving the L_∞ -post-office problem: given any query point $p \in \mathbb{R}^D$, the L_∞ -neighbor of p in S can be found in $O((\log n)^{D-1} \log \log n)$ time. Moreover, the data structure can be maintained under insertions and deletions of points in $O((\log n)^{D-1} \log \log n)$ amortized time.

The approximate L_2 -post-office problem can be solved within the same bounds. In this problem, we have to find a point q in S such that the Euclidean distance between p and q is at most $1 + \varepsilon$ times the Euclidean distance between p and its true neighbor.

Computing the L_1 -Diameter and Center of a Simple Rectilinear Polygon in Parallel

by Sven Schuierer, Universität Freiburg

The diameter of a set S of points is the maximal distance between a pair of points in S . The center of S is the set of points that minimize the distance to their furthest neighbours. The problem of finding the diameter and center of a simple polygon for different distance measures has been studied extensively in recent years. There are algorithms that run in linear time if the geodesic Euclidean metric is used and $O(n \log n)$ time if the link metric is used.

In this talk we consider the L_1 -metric inside a simple rectilinear polygon \mathbf{P} , ie the distance between two points in \mathbf{P} is defined as the length of a shortest rectilinear path connecting them. We give an $O(\log n)$ time algorithm to compute the L_1 -diameter and center on a EREW-PRAM with $n/\log n$ processors if the polygon has n vertices and a triangulation of the polygon is provided.

Direct Computations of Nearest Foreign Neighbors in Arbitrary L^t -Metrics

by Thorsten Graf, Universität Münster, (joint work with Klaus Hinrichs)

Solving proximity problems with simple algorithms is of great importance in fields like traffic-controlling, motion-planning and VLSI-design.

In the all-nearest-foreign-neighbors (ANFN) problem one has to find a nearest foreign neighbor of each point in a set of differently colored points, i. e. a nearest point with a different color. A competitive algorithm uses several L^t -Voronoi diagrams. We show that plane-sweep is a good method to find nearest foreign neighbors directly.

The algorithms we present first distribute the points on receivers which are attached to each of the color sets. The total number of points contained in the receivers is linear after the distribution process. The point is that a nearest foreign neighbor of a point can be found in the receiver of its color. This is a two color ANFN problem for the receivers and their attached points. This can be solved using a generalized plane-sweep

algorithm based on the all-nearest-neighbor algorithm by Hinrichs et al. Our algorithms for distributing the points on the receivers in optimal $O(n \log n)$ time use simple mechanisms and can be implemented using quadrant priority search trees and “colored” BBS-trees. One uses the “Voronoi approach” and the other uses the “Attractor approach”.

Theory Distances and Their Computation

by Heikki Mannila, University of Helsinki, (joint work with Thomas Eiter, TU Wien)

We consider the problem of measuring the distance between theories. We review some of the distance functions proposed in the literature, among them the sum of minimum distances measure, the surjection measure, and the fair surjection measure. Furthermore, we introduce the minimum link measure. We present polynomial time algorithms for computing these measures.

We further address the issue of defining a metric on theories. We present the metric infimum method that constructs a metric from any distance function on theories. In particular, the metric infimum of the minimum link measure is quite intuitive. The computation of this measure is shown to be in NP for a broad class of instances; it is NP-hard for a natural problem class.

Membership Queries in Constant Time and (Order of) Minimum Storage

by Ian Munro, University of Waterloo, (joint work with A. Brodnik)

We consider the problem of representing n elements from a universe of size m [$\log m$ bit words]. This representation must use only $O(\log(\binom{m}{n}))$ bits (i. e. within a constant of the minimum) and support membership queries in constant time, subject to the assumption that memory consists of $\log m$ bit words.

On Partial Order Production

by Svante Carlsson, Luleå Technical University, (joint work with Jingsen Chen)

A survey over the exact cost to produce partial orders (posets) in a comparison-based model is given. We introduce new techniques for constructing lower bounds for the production cost that we also applied to some small posets. We also discussed the notions of mass production and size sensitivity and presented several new posets that are easier to produce using extra elements if several copies of the poset are produced simultaneously.

Polynomial Hash Functions are Reliable

by Martin Dietzfelbinger, Universität Dortmund, (joint work with Y. Gil, Y. Matias, N. Pippenger)

We consider universal hashing with polynomials over finite fields as hash functions. Specifically, consider the class H_m^d of hash functions $h_a : GF(2^k) \rightarrow \{0, 1\}^\mu$, where for $a = (a_0, \dots, a_{d-1}) \in GF(2^k)^d$ we let

$$h_a(x) = \pi \left(\sum_{0 \leq i \leq d} a_i x^i \right),$$

for $x \in GF(2^k)$, where $\pi : GF(2^k) \rightarrow \{0, 1\}^\mu$ is a projection. Assume a set S , $|S| = n = 2^\mu$, is mapped to a hash table $\{0, \dots, m-1\}$, $m = 2^\mu$, by a randomly chosen such function. Then the number B_2 of pairs $x, y \in S$ with $h(x) = h(y)$ is bounded by $2E(B_2) = O(n)$ for $d \geq 4$ with probability $1 - O(n^{-\lfloor d/4 \rfloor})$. No such bound exists for $d = 2$. We indicate consequences for chained hashing, and static and dynamic perfect hashing.

A Robust Implementation of the Bentley/Ottmann Plane Sweep Algorithm

by Stefan Näher, MPI Saarbrücken, (joint work with K. Mehlhorn)

We use the well-known plane sweep algorithm of Bentley & Ottmann for computing the intersection of a set of straight line segments in the plane

to demonstrate some typical problems that arise in the implementation of geometric algorithms.

The first problem we address is the problem of degenerate inputs. We show how this problem can be solved by considering the degenerate case as the normal case when designing the algorithm. This leads to an even shortened program because duplication of code for different special cases can be avoided. The second problem we deal with comes from numerical difficulties with floating point computation that lead to inconsistent decisions. One solution to this problem is the use of exact arithmetic. However, this makes computation more expensive, in our example by a factor of 20 to 30. Finally, we show how to use a so-called floating point filter to substantially reduce the number of exact computations. Floating point filters are based on the observation that (at least for random input) floating point arithmetic gives reliable results in most cases and that then are very few decisions in the program requiring exact arithmetic.

Minimal 2-Dimensional Periodicities and Maximal Space Coverings

by Mireille R egnier, INRIA, (joint work with Ladan Rostami, LITP)

We analyze the structure of 2-dimensional patterns. We propose an extension of 1D period definition and present a classification of periods in 3 classes. Notably, we point out a degeneracy phenomenon due to border effects. We rely on Gauss' results to prove a lattice distribution of non-degenerated periods and we characterize the distribution in the space of degenerated periods. We also prove that 2D periodic patterns have minimal generators, as 1D patterns.

We discussed two algorithmic issues in 2D pattern matching. The characterization of mutually consistent sets of overlapping potential occurrences (of a pattern); maximal coverings of a plane by a given pattern; efficient witness computation algorithms.

The Greedy Triangulation Approximates the Minimum Weight Triangulation and can be Computed in Linear Time in the Average Case

by Christos Levcopoulos, Lund University, (joint work with A. Lingas)

Let S be a set of n points uniformly distributed in the unit square. We show that the expected ratio between the length of the greedy triangulation of S and the minimum weight triangulation of S is constantly bounded. Our main result is an algorithm for constructing the greedy triangulation of S which runs in linear expected time.

Approximating the Maximum Independent Set Problem in Bounded Degree Graphs

by Martin Fürer, Pennsylvania State University, (joint work with Piotr Berman)

The Maximum Independent Set problem for graphs of degree Δ (MIS- Δ) is a well known MAX SNP-complete problem. The best previously known polynomial time approximation algorithm produces an independent set of size $2/\Delta$ times the size of a Maximum Independent Set. $2/\Delta$ is called a performance guarantee.

For every $\Delta \geq 3$ and $\varepsilon > 0$ we present a polynomial time approximation algorithm for the MIS- Δ problem that has a performance guarantee of $\frac{5}{\Delta+3} - \varepsilon$ for even Δ and within ratio $\frac{5}{\Delta+3.25} - \varepsilon$ for odd Δ . For $\Delta = 3$, this is an improvement from $2/3$ to $4/5$. For large Δ 's the improvement is even more significant.

The algorithm is quite simple, mainly a local optimization algorithm. The analysis involves studying combinatorial properties of the graph, where the vertices of a Maximum Independent Set as well as the vertices found by the algorithm are specially marked.

Grail 2.0

by Derick Wood, University of Western Ontario, (joint work with Darrell R. Raymond, University of Waterloo)

Grail is a symbolic manipulation system for formal-language-theory objects. Specifically, it currently handles finite languages, finite-state machines, and regular expressions. The system is implemented in C++ with a heavy use of templates. We discussed the novel use of finite-state object machines, finite-state machines with transition labels taken from some class of objects. For example, when we take letters from an alphabet Σ , we obtain the usual finite-state machines; with ordered pairs from $\Sigma^* \times \Delta^*$, we obtain finite-state transducers; with ordered pairs from $\Sigma^* \times \Delta^{-1}\Delta^*$, where Δ^{-1} is the alphabet of inverse Δ -symbols, we obtain pushdown machines; and so on. In a similar manner we define object languages and (regular) object expressions. The crucial point is that these generalizations enable us to handle any type of machine as it, syntactically, is a finite-state machine; thus, we need only one template (parametrized class) of machines, rather than many classes of machines. Moreover, the transfer functions for regular expressions to finite-state machines and vice versa, are applicable to the object versions. (The system is freely available for academic use.)