PROGRAM FOR

SYMPOSIUM ON NEW DIRECTIONS AND RECENT RESULTS IN ALGORITHMS AND COMPLEXITY

April 7-9, 1976

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pa. 15213

Program Committee: A. Borodin, Cornell University

Richard Karp, University of California, Berkeley

Donald Knuth, Stanford University

Joseph F. Traub, Carnegie-Mellon University

March 17, 1976

The Symposium is supported in part by the National Science Foundation and the Army Research Office.

This is the program for a Symposium on New Directions and Recent Results in Algorithms and Complexity held at the Computer Science Department, Carnegie-Mellon University on April 7-9, 1976. The Symposium consisted of 16 invited and 90 contributed papers; abstracts of most invited and all contributed papers are included. Texts of all the invited papers and abstracts of all contributed papers will appear in the Proceedings to be published by Academic Press in Fall 1976.

SYMPOSIUM ON NEW DIRECTIONS AND RECENT RESULTS
IN ALGORITHMS AND COMPLEXITY

Invited paper presentations will be either 25 or 40 minutes in length with 5 minutes for questions. Contributed paper presentations will be 15 minutes in length with 3 minutes for questions and 2 minutes for room changes. To permit participants to come to specific talks we will adhere strictly to the published schedule. All sessions will be held in Mellon Institute of Science, 4400 5th Avenue, Pittsburgh, Pa. For invited papers, the author giving the talk is starred.

PROGRAM

Tuesday, April 6

7:00 PM - 10:00 PM Registration and Cash Bar - Reunion Lounge at the University Club

Wednesday, April 7

Opening Remarks - Auditorium

9:00 AM - 9:10 AM J. F. Traub, Carnegie-Mellon University

Invited Paper Session I - Auditorium

9:10 AM - 9:55 AM David S. Johnson, Bell Laboratories
"Approximation Algorithms for Combinatorial Problems: Prospects and Limitations"

9:55 AM - 10:25 AM Coffee

10:25 AM - 11:10 AM Michael Ian Shamos, Carnegie-Mellon University
"Geometry and Statistics: Problems at the Interface"

11:10 AM - 11:45 AM Michael O. Rabin, Hebrew University, Israel, and MIT
"Short Expected Time Algorithm for the Nearest Pair"

11:40 AM - 12:25 PM Elwyn R. Berlekamp, University of California, Berkeley
"Combinatorial Games"

12:25 PM - 1:40 PM Lunch

Contributed Paper Sessions 1, 2, 3

Session 1 - Auditorium

1:40 PM - 2:00 PM M. G. Gouda and E. G. Manning, University of Waterloo
"Probabilistic Cost Machines"

Session 1 (continued)

2:00 PM - 2:20 PM — Jean-Claude Arditti, Universite de Paris-Sud, France and McGill University, Canada
"Comparability Graphs of Oriented Trees"

2:20 PM - 2:40 PM — Christos H. Papadimitriou, Princeton University
"Testing Mixed Graphs for Unicursality"

2:40 PM - 3:00 PM — C. G. Goldbogen and G. H. Williams, Union College
"A Comparison of Graph Isomorphism Algorithms"

3:00 PM - 3:20 PM — Kellogg S. Booth, Lawrence Livermore Laboratory
"Problems Polynomially Equivalent to Graph Isomorphism"

3:20 PM - 3:40 PM — Martin Charles Golumbic, Courant Institute, New York University
"The Complexity of Comparability Graph Recognition and Coloring"

3:40 PM - 4:00 PM — Da-Lun Wang, Chung C. Wang, University of Kentucky and Ping Y. Wang, University of Cincinnati
"An Efficient Algorithm for Finding a Minimal Equivalent Subgraph of a Digraph"

Session 2 - Room 355

1:40 PM - 2:00 PM — Silvio Ursic, University of Sao Paulo, Brazil
"Binomial Polytopes and NP-Complete Problems"

2:00 PM - 2:20 PM — Claudia Simionescu, University of Braşov, Romania
"Complex Analysis Methods in Control Theory: A Boundary Control Problem for the Equations in Areolar Derivatives"

2:20 PM - 2:40 PM — Arnie Rosenthal, University of Michigan
"The Structural Complexity of Discrete Optimization Problems"

2:40 PM - 3:00 PM — Fred Taylor, University of Cincinnati
"Minimum Energy Transfer in the Presence of Large Controller Gain Variations"

3:00 PM - 3:20 PM — G. Yuval ~~Sam Savage~~, University of Chicago
"A Convergent Random Walk"

3:20 PM - 3:40 PM — Dorothea A. Klip, University of Alabama at Birmingham
"Infinite Precision Integer Arithmetic in Base 30"

Session 2 (continued)

3:40 PM - 4:00 PM James Hanson and Phillip Russo, Cleveland State University
"Symbolic and Very High Precision Optimization of Constrained Systems of Equations by Newton's Method"

Session 3 - Conference Room

1:40 PM - 2:00 PM P. J. Weinberger, University of Michigan
"Fast Prime Testing"

2:00 PM - 2:20 PM H. T. Kung and J. F. Traub, Carnegie-Mellon University
"Fast Algorithms for Algebraic Functions"

2:20 PM - 2:40 PM C. M. Fiduccia, SUNY, Stony Brook
"Lower Bounds on Active Operations"

2:40 PM - 3:00 PM H. T. Kung, Carnegie-Mellon University and D. M. Tong, The University of Akron
"Fast Algorithms for Partial Fraction Decomposition"

3:00 PM - 3:20 PM Arthur G. Werschulz, Carnegie-Mellon University
"The Computational Complexity of Taylor Series Methods for Initial Value Problems"

3:20 PM - 3:40 PM Marietta J. Tretter, Penn State University and G. W. Walster, University of Wisconsin
"A Fast Algorithm for Significant Digit Computation of the Incomplete Beta Function for Extreme Values"

3:40 PM - 4:00 PM K. Y. Liu, I. S. Reed, and T. K. Truong, University of Southern California
"Fast Algorithms for Complex Integer Transforms"

4:00 PM - 4:30 PM Coffee

Invited Paper Session II - Auditorium

4:30 PM - 5:15 PM Jeffrey D. Ullman, Princeton University
"The Complexity of Code Generation"

5:15 PM - 5:45 PM Robert Endre Tarjan, Stanford University
"Iterative Algorithms for Global Flow Analysis"

6:00 PM - 7:00 PM	Cash Bar at University Club
7:00 PM -	Banquet at University Club

Thursday, April 8

Contributed Paper Sessions 4, 5, 6

Session 4 - Room 355

9:00 AM - 9:20 AM Bernard D. Coleman and Victor J. Mizel, Carnegie-Mellon University
"A Broad Generalization of the Perceptron Convergence Theorem"

9:20 AM - 9:40 AM Peter J. Gingo, University of Akron
"Analysis of the Spectrum of Knee-Generated Noises"

9:40 AM - 10:00 AM P. K. Dash, The University of Calgary, Canada
"Algorithms for Power System Protection Using a Minicomputer"

10:00 AM - 10:20 AM Norbert Cot, Stanford University
"Analysis of Optimal Variable-length Encoding Procedures"

10:20 AM - 10:50 AM Coffee

10:50 AM - 11:10 AM James R. Townes, University of Houston
"A Facial Recognition Algorithm"

11:10 AM - 11:30 AM M. J. Waxman, TRW Inc.
"On Problem Complexity"

11:30 AM - 11:50 AM [redacted] and S. G. Williamson, University of Minnesota
"Recursive Matching Algorithms and Linear Orders on the Subset Lattice"

11:50 AM - 12:10 PM E. A. Galperin, McGill University, Canada
"Loop Properties and Controllability of Linear Congruential Sequences"

Session 5 - Conference Room

9:00 AM - 9:20 AM Robert E. Shostak, Stanford Research Institute
"On the Complexity of Resolution"

9:20 AM - 9:40 AM G. Tourlakis, York University, Canada
"On the Computation of Functions by a Parallel Computer"

9:40 AM - 10:00 AM A. Van der Mude and Adrian Walker, Rutgers University
"Some Results on the Inference of Stochastic Grammars"

10:00 AM - 10:20 AM John Gaschnig, Carnegie-Mellon University
"Analysis of Heuristics as Functions"

Session 5 (continued)

10:20 AM - 10:50 AM Coffee

10:50 AM - 11:10 AM ~~Martine Paget~~ Sam Savage, Université Paris, France
"Markov Algorithms and Their Relations with Associative Processing System"

11:10 AM - 11:30 AM Chee-Keng Yap, Yale University
"New Results for Medians and Related Problems"

11:30 AM - 11:50 Zbigniew Raś, University of Warsaw, Poland
"On a Relationship between Programs of Address Machines, Mazurkiewicz Algorithms and FS-expressions of a Formalized Algorithmic Language"

11:50 AM - 12:10 PM Martin K. Solomon, Rutgers University
"A Complexity Theoretic Characterization of Non-Recursive Differential Ideals"

Session 6 - Auditorium

9:00 AM - 9:20 AM Francis Y. Chin and David J. Houck, University of Maryland
"Algorithms for Updating Minimal Spanning Trees"

9:20 AM - 9:40 AM Winston Riley III, Riley Systems Corporation
"Micro-Analysis as a Technique for Problems on a Graph"

9:40 AM - 10:00 AM Jan van Leeuwen, State University of New York at Buffalo
"On Finding Lowest Common Ancestors in Less than Logarithmic Average Time"

10:00 AM - 10:20 AM Da-Lun Wang, University of Kentucky
"On the Average Path Length between Two Nodes in a Randomly Generated Tree"

10:20 AM - 10:50 AM Coffee

10:50 AM - 11:10 AM Rici Liknaitzky and W. J. Meyers, Mackay, Alberta, Canada
"Fewest-Bit Encoding of Plane Trees"

11:10 AM - 11:30 AM R. W. Deming, State University of New York at Oswego
"An Algorithm for the Optimal Orientation of a Graph"

11:30 AM - 11:50 AM M. M. Newborn, McGill University, Canada
"The Efficiency of the Alpha-Beta Search on Trees"

11:50 AM - 12:10 PM Vijay B. Aggarwal and James W. Burgmeier, University of Vermont
"Error Complexity of Parallel Evaluation of Arithmetic Expressions"

12:10 PM - 1:20 PM Lunch

Contributed Paper Sessions 7, 8, 9

Session 7 - Conference Room

1:20 PM - 1:40 PM — A. Toni Cohen, Lincoln University and Leon S. Levy, University of Delaware
"A Unifying Model for Parallel Processing"

1:40 PM - 2:00 PM — M. S. Schlansker, The University of Michigan
"Complexity Analysis of Some Problems in Pipeline Processor Design"

2:00 PM - 2:20 PM — Paul G. Tuttle, Babcock and Wilcox
"Vector Processing of the Fast Fourier Transform"

2:20 PM - 2:40 PM — Frances L. Van Scoy, Old Dominion University
"Processor Utilization in a Parallel System"

2:40 PM - 3:00 PM — Miloš D. Ercegovac, University of California, Los Angeles
"An Approach in Reducing Complexity of Numerical Computations"

3:00 PM - 3:20 PM — D. A. Calahan, University of Michigan
"Complexity Analysis of Matrix Benchmarks on the Cray-1"

3:20 PM - 3:40 PM — W. Morven Gentleman, University of Waterloo, Canada
"Some Complexity Results for Matrix Computations on Parallel Processors"

Session 8 - Room 355

1:20 PM - 1:40 PM — Shuo-Yen Robert Li, MIT
"Why Binary Insertion Sorting and Tree Selection Sorting Require the Same Number of Comparisons"

1:40 PM - 2:00 PM — Donald B. Johnson and Samuel D. Kashdan, The Pennsylvania State University
"Finding the K-th Largest Pair is $\Omega(\max(n,\sqrt{K}\log K))$"

2:00 PM - 2:20 PM — Tetsuo Mizoguchi, Mitsubishi Electric Corporation, Japan
"Finding the Kth Pair Efficiently"

2:20 PM - 2:40 PM — Paul Cull, Oregon State University
"Short Tours of Sequential Machines"

2:40 PM - 3:00 PM — S. Goto, Nippon Electric Co., Ltd., on leave from University of California, Berkeley.
"On Updating Shortest Paths and a Decomposition Algorithm"

Session 8 (continued)

3:00 PM - 3:20 PM — Laurence L. Lang and J. Denbigh Starkey, Washington State University
"An o(e log n) Shortest Path Algorithm for Sparse Graphs"

3:20 PM - 3:40 PM — W. Hsieh and A. Kershenbaum, Network Analysis Corp.
"A Diverse Shortest Path Procedure"

Session 9 - Auditorium

1:20 PM - 1:40 PM — Gerald Cooper and David Dobkin, Yale University
"Computer Generated Algorithms - A Constructive Approach to the Complexity of Bilinear Multiplication Operations"

1:40 PM - 2:00 PM — Roger W. Brockett, Harvard University
"The Multiplicative Complexity of a Family of Bilinear Forms can be High"

2:00 PM - 2:20 PM — L. H. Harper, University of California, Riverside and J. E. Savage, Brown University
"An $O(n^2 \log_2 n^2)$ Lower Bound to the Complexity of Matrix Multiplication"

2:20 PM - 2:40 PM — Julian David Laderman, Lehman College
"Algorithms for Matrix Products Using Fewer Multiplications"

2:40 PM - 3:00 PM — Paolo Zellini, Università di Pisa and Carnegie-Mellon
"On the Optimal Computation of a Set of Symmetric, Persymmetric or both Symmetric and Persymmetric Bilinear Forms"

3:00 PM - 3:20 PM — Thomas D. Howell, Cornell University
"Optimal Algorithms for Quarternion Multiplication"

3:20 PM - 3:40 PM — R. C. Backhouse, Heriot-Watt University, Scotland
"Comparisons of Iterative and Elimination Techniques in Closure Algorithms"

3:40 PM - 4:10 PM — Coffee

Invited Paper Session III - Auditorium

4:10 PM - 4:40 PM — J. F. Traub, Carnegie-Mellon University and H. Woźniakowski*, University of Warsaw
"Optimal Linear Information for the Solution of Non-Linear Equations"

Invited Paper Session III (continued)

4:40 PM - 5:10 PM	R. William Gosper, Jr., Stanford University "A Calculus of Series Rearrangements"
5:10 PM - 5:40 PM	H. T. Kung, Carnegie-Mellon University "Efficient Parallel Algorithms for Asynchronous Multiprocessors"
5:40 PM - 6:10 PM	Randolph E. Bank, University of Chicago and Donald J. Rose*, Harvard University "Extrapolated Fast Direct Algorithms for Elliptic Boundary Value Problems"
6:30 PM - 7:30 PM	Cash Bar at Carnegie-Mellon Faculty Lounge
7:30 PM -	Banquet at Carnegie-Mellon Faculty Club

Friday, April 9

Invited Paper Session IV - Auditorium

8:30 AM - 9:15 PM R. M. Karp, University of California
"Probabilistic Analysis of Heuristic Search Methods"

9:15 AM - 9:45 AM Joseph B. Kadane* and Herbert A. Simon, Carnegie-Mellon University
"Problems of Computational Complexity in Artificial Intelligence"

9:45 AM - 10:30 AM Joel Moses, Massachusetts Institute of Technology
"Algebraic Structures and Their Algorithms"

10:30 AM - 11:00 AM Coffee

Contributed Paper Sessions 10, 11, 12

Session 10 - Auditorium

11:00 AM - 11:20 AM Armin Cremers and Thomas N. Hibbard, University of Southern California
"An Algebraic Approach to Concurrent Programming Control and Related Complexity Problems"

11:20 AM - 11:40 AM Reinhold Weicker, Pennsylvania State University
"Context Free Language Recognition by a RAM with Uniform Lost Criterion in Time $n^2 \log n$"

11:40 AM - 12:00 PM Karl Winklmann, Purdue University
"On the Complexity of Parameter Transmission Questions in Higher Programming Languages"

12:00 PM - 12:20 PM Hans-Georg Stork, Fachbereich Informatik der TH Darmstadt, West Germany
"On the Paging-Complexity of Loop-Like Structures"

Session 11 - Room 355

11:00 AM - 11:20 AM R. I. Andrushkiw, New Jersey Institute of Technology
"On a Class of Non-linear Eigenvalue Problems with Unbounded and Nonhermitian Operators"

11:20 AM - 11:40 AM R. Beauwens and L. Quenon, Universite Libre de Bruxelles, Belgium
"Existence Criteria for Partial Matrix Factorizations"

Session 11 (continued)

11:40 AM - 12:00 PM J. Herzberger, Universität Karlsruhe, West Germany
"Properties of Some Modified Interpolatory Iteration Functions"

12:00 PM - 12:20 PM Boris S. Verkhovsky, Princeton University
"Algorithm with Nonlinear Acceleration for a System of Linear Equations"

Session 12 - Conference Room

11:00 AM - 11:20 AM J. R. Charnetski, State University of New York, Binghamton
"RAM: A Generalized Algorithm for the Generation of Vector Points in Arbitrary Bounded N-Dimensional Polyhedral Sets"

11:20 AM - 11:40 AM Henry K. DeWitt, System Development Corp. and Michael M. Krieger, University of California, Los Angeles
"Expected Structure of Euclidean Graphs"

11:40 AM - 12:00 PM A. Keewatin Dewdney, University of Western Ontario, Canada
"Complexity of Nearest Neighbour Searching in Three Dimensions"

12:00 PM - 12:20 PM Franco P. Preparata, University of Illinois at Urbana-Champaign
"On the Determination of a Densest Hemisphere"

12:20 PM - 1:20 PM Lunch

Contributed Paper Sessions 13, 14, 15

Session 13 - Room 355

1:20 PM - 1:40 PM Robert S. Boyer, Stanford Research Institute and J. Strother Moore, Xerox Palo Alto Research Center
"A Fast String Searching Algorithm"

1:40 PM - 2:00 PM F. P. Preparata, D. E. Muller, University of Illinois and A. B. Barak, Hebrew University, Israel
"Reduction of Depth of Boolean Networks with a Fan-In Constraint"

2:00 PM - 2:20 PM Edmund A. Lamagna, The Mitre Corporation
"The Complexity of Monotone Networks for Certain Bilinear Forms, Routing Problems, Sorting, and Merging"

Session 13 (continued)

2:20 PM - 2:40 PM Paul C. Kainen, Case Western Reserve University
"Long and Short Walks in Trees"

Session 14 - Auditorium

1:20 PM - 1:40 PM Nicos Christofides, Carnegie-Mellon University and"Imperial College, London
"Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem"

1:40 PM - 2:00 PM Gerald L. Thompson, Carnegie-Mellon University
"Solutions of Very Large Travelling Salesman Problems"

2:00 PM - 2:20 PM R. J. Lipton, Yale University and Y. Zalcstein, State University of New York, Stony Brook
"Computational Complexity of Group-Theoretic Problems"

2:20 PM - 2:40 PM David A. Plaisted, University of Illinois
"Some NP-Hard Problems Involving Sparse Polynomials"

Session 15 - Conference Room

1:20 PM - 1:40 PM Lee J. White, Ohio State University
"An Analysis of a Geographic Data Base Problem"

1:40 PM - 2:00 PM Gaston H. Gonnet and Lawrence D. Rogers, University of Waterloo, Canada
"An Algorithmic and Complexity Analysis of the Heap as a Data Structure"

2:00 PM - 2:20 PM W. A. Burkhard, University of California, San Diego
"Optimal PMF-Trie Designs"

2:20 PM - 2:40 PM Lawrence Yelowitz, University of Pittsburgh
"Proving 'Producer/Consumer' Correctness by Path Expressions and Schemas"

Invited Paper Session V - Auditorium

2:45 PM - 3:15 PM Michael S. Paterson, University of Warwick, England, and Stanford University
"Analysis of Hashing Algorithms"

Invited Paper Session V (continued)

3:15 PM - 3:45 PM Richard P. Brent, Australian National University, Australia
"Analysis of the Binary Euclidean Algorithm"

3:45 PM - 4:15 PM D. E. Knuth, Stanford University and A. C. Yao*, MIT
"A Computational Model for Generating Numerical Distributions"

ABSTRACTS OF INVITED PAPERS

The abstracts are listed in alphabetical order according to the name of the first author.

ABSTRACTS OF INVITED PAPERS

The abstracts are listed in alphabetical order according to the name of the first author.

Extrapolated Fast Direct Algorithms for Elliptic Boundary Value Problems

Randolph E. Bank
Department of Mathematics
The University of Chicago

Donald J. Rose
Applied Mathematics, Aiken Computation Laboratory
Harvard University

Abstract. Recently the authors have derived several new direct methods for solving systems of equations arising from finite difference discretizations to elliptic boundary value problems on regular domains in the plane. These numerically satisfactory methods are "fast" in that they require $O(n^2)$ or $O(n^2\log(n/k))$ time and $O(n^2)$ space (k a parameter) to compute the solution at n^2 mesh points. Although this allows an efficient computation for even large values of n (say $n = 127$), the computed solution differs from the solution to the partial differential equation by an amount $e = O(n^{-2})$. In this paper we examine the use of fast direct algorithms in conjunction with Richardson extrapolation in order to efficiently compute much more accurate solutions to the partial differential equation at n_0^2 initially chosen mesh points.

COMBINATORIAL GAMES

by Elwyn R. Berlekamp

U. California, Berkeley

This talk will be an introductory survey of two-person zero-sum games with perfect information. We are particularly interested in sums of such games, in which the mover may select any single summand and make any legal move therein. When the player to move has no legal move, the game ends and the player unable to move loses. About 1970, J. H. Conway discovered a simple set of axioms for games, and these axioms have led to the complete solution of many previously unsolved games as well as to other results of theoretical interest. In the special case of impartial games, the theory reduces to the Sprague-Grundy generalization of nim. Another interesting special case is the class of zuchswang games, which underlie the "Hackenbush number system" described in Berlekamp's Information and Control paper, and Conway's Surreal Numbers, which were popularized by Knuth's 1975 paperback and extended by Martin Kruskal, who is now writing a book on Surreal Analysis. Other interesting special cases of the general theory are the "hot" games, which contain the model of economic games previously studied by Milnor and Hanner, and the short "infinitesmal" games, which are quite unlike anything else. These and many other topics are discussed in Conway's book ONAG, which is now in press, and in another book called Winning Ways, by Berlekamp, Conway and Guy, which is now in preparation.

This talk will attempt to illuminate some parts of the general theory which are illustrated by the game of Hackenbush. This game is played on a grounded graph in which each branch has one of three colors: blue, green, or red. At each turn, Right must remove a red branch or a green branch, and then Left must remove a blue branch or a green branch. Any subgraph no longer connected to the ground is immediately removed, and the game continues until there are no branches of any color takable by the mover. Graphs containing only green branches are impartial, and they may be solved via Sprague-Grundy numbers which can be computed via a polynomial-time algorithm described in Martin Gardner's January 1974 Scientific American column. Graphs containing only red and blue branches are zuchswang, and can be solved only by determining their numerical values. The values of trees are quickly obtained via the Hackenbush number system, but the problem of determining the values of another class of red-blue graphs, called redwood furniture, is shown to be NP complete. Nevertheless, several significant classes of graphs containing all three colors of branches have infinitesmal values, the highest order of which can be quickly determined via the min-cut max-flow approach.

ANALYSIS OF THE BINARY EUCLIDEAN ALGORITHM

Richard P. Brent
Australian National University
and Carnegie-Mellon University

The binary Euclidean algorithm finds the GCD of two integers u and v using subtraction, shifting and parity testing. Unlike the classical Euclidean algorithm, no divisions are required. We analyse a continuous model of the binary algorithm, and find the expected number of iterations.

Suppose that u and v are initially odd integers chosen randomly subject to $0 \leq v \leq u \leq N$, and are subsequently as at step B6 of the algorithm as described in Knuth (Vol. 2, Sec. 4.5.2, Algorithm B). Let $F_n(x)$ be the distribution function of $x = \min(u,v)/\max(u,v)$ at the n-th iteration (assuming $N \to \infty$ so x is regarded as continuous rather than discrete). We show that

$$F_{n+1}(x) = 1 + \sum_{k=1}^{\infty} 2^{-k} \left[F_n\left(\frac{x}{2^k+x}\right) - F_n\left(\frac{1}{1+2^k x}\right) \right] \qquad \text{for } n \geq 0,$$

and

$$F_0(x) = x \qquad (0 \leq x \leq 1).$$

We also show that $F_n(x)$ has the form

$$F_n(x) = \alpha_n(x)\log_2 x + \beta_n(x),$$

where $\alpha_n(x)$ and $\beta_n(x)$ are analytic and regular at the origin, and satisfy certain recurrence relations. The recurrence for $\alpha_n(x)$ may be solved explicitly. Assuming that a limiting distribution $F_\infty(x) = \alpha_\infty(x)\log_2 x + \beta_\infty(x)$ exists, we have

$$\alpha_\infty(x) = -\left(\frac{3F'_\infty(1)x}{4}\right)^{\infty} \sum_{k=0}^{\infty} 2^{-k} \sum_{j=0}^{2^k-1} \left(\frac{1}{2^k+jx}\right),$$

where (numerically) $F'_\infty(1) = 0.397922681...$ However, the recurrence for $\beta_n(x)$ has only been solved numerically, except for small n.

From numerical evidence we conjecture that $F_\infty(x)$ exists, and that

$$|F_n(x) - F_\infty(x)| = O(\lambda^n x|\ln(x)|),$$

uniformly for all $x \in (0, 1)$, all $n \geq 0$, and some $\lambda \in (0, 1)$. We can calculate $F_\infty(x)$ numerically by several different methods, and deduce that the expected number of iterations is $K\log_2 N + O(1)$ as $N \to \infty$, where $K = 0.705971246...$ This agrees fairly well with Knuth's result $K \simeq 0.703$, obtained by simulation with a discrete model (Sec. 4.5.2, eqn. (43)).

A Calculus of Series Rearrangements

by R. Wm. Gosper, Jr.
Stanford University

Abstract

Many summation identities, both old and new, can be derived with a transform based on a very general kind of series rearrangement determined by "splitting functions".

Let k count the number of times the series has been rearranged, and let n be the summation index. Let the nth term ratio of the rearranged series be $r_{k,n}$. Then the splitting function $s_{k,n}$ defines the rearrangement into the series with ratio $r_{k+1,n}$. Any error introduced by this rearrangement is easily expressed as an infinite product. As the number of "splits", k, goes to infinity, a series will be transformed into an expression involving two new series and two infinite products, but usually the result will simplify to a single series or even a closed form.

Examples:

$$\sum_{k\ge1} \frac{1}{\sinh kx} = \sum_{k\ge1} \frac{e^{-k^2x/2}}{\sinh kx/2}$$

$$\sum_{k\ge0} \frac{1}{(k+1)\binom{k+1/2}{k}^2} = 2\pi\lambda - 7\,\zeta(3)/2 \qquad \text{(Where } \lambda = \text{Catalan's constant} = \frac{1}{1^2} - \frac{1}{3^2} + \frac{1}{5^2} - \ldots\text{)}$$

$${}_4F_3\left[\begin{matrix}1, & a, & b, & c\\ d+1, & e+1, & a+b+c-d-e+1\end{matrix}\right] = \frac{a+c-d-e}{(a-d)(c-d)(a-e)(c-e)}\left(de(a+b+c-d-e) - \frac{(a+b+c-d-e)!\;d!\;e!}{(a-1)!\;(b-1)!\;(c-1)!}\right)$$

$$\text{(Provided } (a+b+c-d-e)^2 = a^2+b^2+c^2-d^2-e^2\text{)}$$

$$\sum_{k\ge1} \frac{30k-11}{16k^3(2k-1)\binom{2k-1}{k}^2} = \zeta(3)$$

$$\frac{5}{16} + \sum_{k\ge1} \frac{(42k+5)\binom{2k-1}{k}^3}{2^{12k+1}} = \frac{1}{\pi}$$

Each splitting function identity is itself subject to an infinite semigroup of transformations, resulting in many identities relating pairs of series apparently quite different from the original pair.

Applications will include:

Unification and extension of the theory of generalized hypergeometric series.
(Binomial coefficient identities are a special case.)
Acceleration of convergence, with telescopy to closed form as a limiting case.
Fourier series identities.

This work takes up almost exactly where James Stirling left off, and had he been granted access to a computerized symbolic mathematics system such as MIT MACSYMA, Stirling probably would have done most of this work before 1750. I am indebted to MACSYMA project director Joel Moses, his staff, and to MACSYMA itself for months of unflagging helpfulness. I hope to repay them soon by incorporating into MACSYMA some of the series transformation algorithms presented here.

APPROXIMATION ALGORITHMS FOR COMBINATORIAL PROBLEMS: PROSPECTS AND LIMITATIONS

by

David S. Johnson
Bell Laboratories
Murray Hill, New Jersey

ABSTRACT

An approximation algorithm for a combinatorial optimization problem is an algorithm which, while not guaranteed to find an optimal solution, does try to find a solution which is near-optimal. Since NP-completeness results for many of the commonly encountered optimization problems have made us abandon hope of finding practical optimization algorithms for these problems, attention has been turning to the possibility of designing fast approximation algorithms, especially those which guarantee that the solution they find will be close to optimal. Recent results have even shown that for some problems we can design what might be called approximation schemes: algorithms which for any $\epsilon > 0$ can produce in polynomial time a solution whose value has a ratio to the optimal solution value that differs from 1 by at most ϵ. We survey recent results about approximation algorithms and schemes, and discuss how new applications of the theory of NP-completeness are helping us to differentiate the cases when either one or both of these approaches fail to give useful results.

PROBLEMS OF COMPUTATIONAL COMPLEXITY IN ARTIFICIAL INTELLIGENCE

Joseph B. Kadane and Herbert A. Simon

Carnegie-Mellon University

ABSTRACT

Typical problems in computational complexity may be characterized as follows: We have a one-parameter domain of computations. We wish to determine the minimum number, C(N), of operations required to carry out computations from the subdomain of size N, for all N. The function, C(N), measures the complexity of the domain.

This paradigm appears to be unsuitable for measuring the complexity of problem-solving algorithms. Here we are interested in the number of computations required to solve problems, averaged over a sample of problems drawn from some domain. An algorithm may be regarded as efficient for a domain if it minimizes this average; and the computational complexity of a problem domain may be identified with the average calculated for an algorithm that is efficient for the domain.

This paper surveys what is known about the efficiency of algorithms for problem-solving search, and suggests how this knowledge can be applied to define the complexity of problem-solving domains.

Probabilistic Analysis of Heuristic Search Methods

R. M. Karp
University of California

Title: A computational Model for Generating Numerical Distributions

Authors: D.E. Knuth, Computer Science Department, Stanford University, and A.C. Yao, Department of Mathematics, MIT

Abstract: In the simulation of probabilistic events, it is often necessary to generate data values that specify a certain numerical distribution. This is generally done with procedures involving manipulation of random numbers. We propose a model for formalizing this process, and measure the cost of a procedure by the number of random bits it calls for. We can characterize optimal procedures for arbitrary distributions. Applications and trade-off phenomena are discussed.

Efficient Parallel Algorithms for Asynchronous Multiprocessors

H. T. Kung
Department of Computer Science
Carnegie-Mellon University

In the algorithm design for asynchronous multiprocessors, the following questions are encountered:

1. How can a problem be decomposed?
2. What is the proper size of a subproblem?
3. How many processes should be created?
4. What is the best interprocesses communication mechanism?
5. What are the worst and the expected running times?

To answer these questions, analytic models are needed so that algorithms can be analyzed. This paper surveys recent results on synthesis and analysis of multiprocessor algorithms. The problems to be considered include searching, sorting, iterative methods for linear systems and for nonlinear equations.

Algebraic Structures and Their Algorithms

Joel Moses
Laboratory for Computer Science
Massachusetts Institute of Technology

Abstract

Several algorithms in the area of algebraic manipulation are examined. It is shown how the proper representation for the expressions such as ideals and fields have led to relatively natural algorithms which are essentially driven by the representation. The solution of many nonnumerical problems may be held back by our current lack of understanding of good ways of structuring the data. Areas in which it would be worthwhile to pursue new ways of structuring information are data bases and pattern recognition.

Analysis of Hashing Algorithms

Michael S. Paterson
University of Warwick, England
and Stanford University

Short Expected Time Algorithm for the Nearest Pair

Michael O. Rabin

Hebrew University and MIT

Given n points $x_1, \ldots, x_n \in R^k$ in k-space, the nearest pair is the pair (x_i, x_j) such that $d(x_i, x_j) = \min d(x_p, x_q)$, $1 \leq p < q \leq n$. We give a stochastic algorithm for discovering the nearest pair which runs in expected time $O(n)$. The bound on the expected time is derived from the randomization method used in the algorithm, and applies uniformly to every n-tuple. It is not based on any assumption concerning the distribution of the n-tuples $(x_1, \ldots, x_n)$.

The notion of using randomization in combinatorial and algebraic algorithms seems to have rich potentialities. Three types of such algorithms can be defined. (1) Algorithms for which the expected time, for every instance of the problem, is short, and which always terminate in a finite number of steps. (2) Algorithms with a short expected time, which (with probability 0) may not terminate. (3) Algorithms employing randomization which may produce wrong answers but can be made arbitrarily reliable.

Some examples of these ideas, including an application to the discovery of prime numbers, will be given.

GEOMETRY AND STATISTICS: PROBLEMS AT THE INTERFACE

Michael Ian Shamos
Carnegie-Mellon University

Statistics provides a rich source of unanalyzed algorithms, many with a distinctly geometric character. The connection with geometry arises from viewing a statistical sample as a set of points in Euclidean space; such problems as correlation, regression, and estimating location can then be formulated in a purely geometric way. We exploit this analogy and borrow results from computational geometry to obtain upper and lower bounds on the complexity of computing various statistics. We present a collection of new, fast algorithms for familiar statistical problems.

Nonparametric statistics makes frequent use of sorting, selection, and other combinatorial procedures. We present several related algorithms that are directly motivated by statistical problems: the ranks of any k elements of a set of n objects can be found in $O(n \log k)$ time and the kth largest element of the Cartesian sum $X + Y$ of two n-sets can be found in $O(n \log n)$ time. This improves the best known (quadratic) algorithm for computing the Hodges-Lehmann estimator. On the disconcerting side, we show several statistical problems to be NP-complete.

Statistics comes to the aid of geometry in the average-case analysis of geometric algorithms, a frequently perilous study. We discuss the expected-time behavior of convex hull algorithms and present a paradox in geometrical probability which suggests that care is required in order to obtain meaningful average-case results.

A statistician chooses his procedures based on measures of statistical efficacy, such as power, robustness, and relative efficiency (the ability to use small samples). The computer scientist is interested in measures of computational efficacy, such as complexity and numerical stability. The challenge of computational statistics is to discover relationships among these measures and use algorithmic ideas to motivate new statistical methods.

Iterative Algorithms for Global Flow Analysis

Robert Endre Tarjan
Computer Science Department
Stanford University
Stanford, California 94305

Abstract. This paper studies iterative methods for the global flow analysis of computer programs. We define a hierarchy of global flow problem classes, each solvable by an appropriate generalization of the "node listing" method of Kennedy. We show that each of these generalized methods is optimum, among all iterative algorithms, for solving problems within its class. We give lower bounds on the time required by iterative algorithms for each of the problem classes.

OPTIMAL LINEAR INFORMATION FOR THE SOLUTION OF NON-LINEAR EQUATIONS

J. F. Traub
Department of Computer Science
Carnegie-Mellon University

H. Woźniakowski
Department of Mathematics
University of Warsaw

Classically, given a problem, we make a decision as to what information about this problem is relevant to its solution. In this paper we show how we may <u>derive</u> what information is relevant from first principles for an important general class of problems. Let F be an operator with $F: D \subset B_1 \to B_2$ where B_1, B_2 are real or complex Banach spaces of dimension N, $1 \leq N \leq \infty$, $N = \dim(B_1) = \dim(B_2)$. Let α be a simple zero of F.

Let ϕ be any one-point algorithm for the solution of $F(x) = 0$. At each iterative step we can calculate $L_i(F;x)$, $i = 1,\ldots,n$ where L_i is any <u>linear operator</u>. We derive a purely algebraic condition for ϕ to be of order k. We show $k \leq n$ and that $k = n$ iff $L_i(F;x)$, $i = 1,\ldots,n$ is "standard" information.

Future extensions of these results are discussed.

THE COMPLEXITY OF CODE GENERATION

Jeffrey D. Ullman
Department of Electrical Engineering
&
Computer Science
Princeton University
Princeton, N. J. 08540

The problem of generating optimal code for a program is NP-complete, even for very trivial target machines. There is a long history of developments which isolated tractable subcases from intractable ones. Moreover, there has been recent study of heuristics for generating code and their worst case behavior. The principal results in these areas will be surveyed.

ABSTRACTS OF CONTRIBUTED PAPERS

The abstracts are listed in alphabetical order according to the name of the first author.

Error Complexity of Parallel Evaluation of Arithmetic Expressions

Vijay B. Aggarwal
James W. Burgmeier

Department of Mathematics
University of Vermont
Burlington, Vermont

Examples may be cited for which different techniques of evaluating a given arithmetic expression can result in significantly different error bounds. In this paper the error accumulation for different evaluations of division-free arithmetic expressions is analyzed. Strategies are sought which minimize computation time on an MIMD machine as well as satisfy the smallest bound for round-off error accumulation. Finally the problem of minimum error inherent in a given computational problem in this context is discussed.

Considering the errors associated with the terms of an arithmetic expression as an error vector, rules are developed to facilitate the computations of accumulated error in a particular evaluation. The sum norm and max norm are used to compare the error vectors resulting from different evaluations of an arithmetic expression. For the case of the addition of n numbers the sequential-sum and parallel-sum techniques yield significantly different error bounds with the latter satisfying the best possible bound. Further the error bound obtained for the parallel-sum technique represents the minimal error complexity inherent in the problem. On the other hand for the product of n numbers the error is invariant under all non-trivial evaluations.

The error terms due to additions are shown to be invariant using only the commutative and distributive laws. However, the use of associative law can result in equivalent forms of an arithmetic expression having radically different error bounds. In many cases suitable use of the associative law can lower the error bound from n to $\log_2 n$. Strategies using the associative law to reduce the error bounds may require the use of the distributive law. Strategies designed to minimize error bound and computation time are discussed.

On a Class of Non-linear Eigenvalue Problems with Unbounded and Nonhermitian Operators

R. I. Andrushkiw
Department of Mathematics
New Jersey Institute of Technology, Newark, New Jersey

It is known that for certain selfadjoint and positive definite differential eigenvalue problems (i) $Mu - \lambda Nu = 0$ at least one eigenvalue of the problem (i) lies in the interval (A,B) where $A = \mu_2 - \{(\mu_1 - \mu_2)\ \mu_2\}^{\frac{1}{2}}, B = \mu_2 + \{(\mu_1 - \mu_2)\ \mu_2\}^{\frac{1}{2}}$ and μ_1, μ_2 are the first two Schwarz constants. (See Collatz, "Eigenwertaufgaben mit Technischen Anwendungen," Academic Press, 1963, page 196.) In this paper we extend the above result to the eigenvalue problem (ii) $Au - \lambda Bu - \lambda^2 Cu = 0$ where A, B and C are some linear (in general unbounded and nonhermitian) operators in a complex and separable Hilbert space H. In addition, we construct a sequence of intervals $I_k^{(r)} = [A_k^{(r)}, B_k^{(r)}]$, $k = 1,2,\ldots,$ each containing at least one eigenvalue of (ii), such that the length of the intervals $I_k^{(r)}$ converges to zero, and the constants $A_k^{(r)}$, $B_k^{(r)}$ converge to the r^{th} eigenvalue (in order of increasing magnitude) of the problem (ii) as $k \to \infty$. The theory is illustrated by determining the smallest eigenvalue of a non-linear differential eigenvalue problem of the type (ii).

Comparability Graphs of Oriented Trees

Jean-Claude Arditti
Universite de Paris-Sud

on leave at
McGill University
Montreal, Canada

The Comparability graph $T(G)$ of an oriented (resp. rooted) tree G is the undirected graph corresponding to the transitive closure $\hat{T}(G)$ of the tree G.

An undirected graph H is a comparability graph of tree if for some oriented tree G, one has $H = T(G)$.

Comparability graphs (undirected graph such that one can direct their edges and produce a transitive graph) were characterised by Ghonilà-Houri, Gilmore and Hoffman.

Comparability graphs of rooted trees were characterised by E.S. Wolk as the connected graphs such that for any path $a\,b\,c\,d$, either a is a neighbour of c or b is a neigbour of d.

We prove that comparability graphs of oriented trees are characterised by a set of four properties :

P_1 : H is a comparability graph

P_2 : No subgraph of H is isomorphic to H_1

P_3 : No subgraph of H is isomorphic to H_2

P_4 : For any pair of vertices a and b, all minimal paths between a and b have the same length. Moreover the set $X_i(a,b)$ of all vertices of H that can occur in the i-th position in such a minimal path, defines in H a subgraph which is the comparability graph of a <u>rooted</u> tree

We obtain an algorithm that recognizes if a given undirected graph H is a comparability graph of tree, and (if it is the case) exhibits a tree G such that $T(G) = H$. This algorithm works in polynomial time.

Moreover if two trees G and G' are such $T(G) = T(G')$ then either G is isomorphic to G' or G is isomorphic to the symmetric of G'. Therefore the class of comparability graphs of trees is "large enough".

We also prove that, in the class of comparability graphs of trees, the existence and construction of <u>hamilton circuits</u> are solvable in polynomial time.

COMPARISONS OF ITERATIVE AND ELIMINATION TECHNIQUES IN CLOSURE ALGORITHMS

R.C. BACKHOUSE

Dept. of Computer Science, Heriot-Watt University

Edinburgh, Scotland

We consider algorithms for finding the minimal solution Y=A*B of the equation Y=AY+B in a regular algebra [1]. A general condition for the applicability of the iterative techniques is stated and it is then shown that a generalisation of Yen's algorithm [5] invariably requires the same number or fewer operations than the Gauss-Seidel method. (Note: Gauss-Seidel's method is known as Hecht's algorithm [3] when applied to program optimisation problems, and Ford's algorithm [2] when applied to shortest path problems.)

It is then shown that the well-known comparison between Gaussian elimination and Jordan elimination in a linear algebra [4] is equally applicable in a regular algebra.

It is important to note that the comparisons we give are not worst-case or probabilistic analyses, but are quite independent of the input.

1. Backhouse, R.C. and Carre, B.A. "Regular Algebra Applied to Path-Finding Problems" J. Inst. Maths. Applics. 15, 161-186.

2. Ford, L.R. and Fulkerson, D.R. "Flows in Networks" Princeton Univ. Press: Princeton, N.J. (1962).

3. Hecht, M.S. "Global data flow analysis of computer programs" Ph.D. thesis, Dept. of Elect. Eng., Princeton Univ. (1973).

4. Tewarson, R.P. "Sparse Matrices" Academic Press: New York and London (1973).

5. Yen, J.Y. "An algorithm for finding shortest routes from all source nodes to a given destination in general networks" Quart. Appl. Maths. 27 (1970) 526-530.

EXISTENCE CRITERIA FOR PARTIAL MATRIX FACTORIZATIONS

(R. BEAUWENS and L. QUENON)

Since a partial matrix factorization algorithm fails when a division by zero occurs, the iterative methods based on such algorithms raise, in addition to the usual convergence problem, a new problem of "feasibility". Conditions which prevent the occurrence of a division by zero during execution of a factorization algorithm or factorizability conditions have recently been investigated (2) for a class of algorithms called the Oliphant-Buleev or O.B. methods; we summarize here the main results obtained so far.

The class of O.B. algorithms, formerly introduced by one of the authors (1) was shown to include generalized forms of the Stone (4) and Buleev (3) schemes; the main interest of these generalized algorithms is of being applicable to sparse matrices of arbitrary structure.

We proved that in the range of practical interest for the iteration parameters, the generalized Stone and the generalized Buleev factorizations do not fail when they are applied to M-matrices or to H-matrices. Factorizability conditions then follow for wide classes of O.B. methods.

Extensions of these results to block O.B. factorization iterative methods have also been developed, for block-M-matrices and block-H-matrices.

Although convergence considerations lied in principle outside our scope, it was realized that the factorizability conditions relative to hermitian factorizations could provide essential indications on the convergence problem; in particular convergence criteria were obtained for point symmetric O.B. methods.

(1) R. BEAUWENS, "Convergence analysis of some factorization iterative methods", Ser. Appl. Math., Rep. 73-7, Northwestern University.

(2) R. BEAUWENS and L. QUENON, "Existence criteria for partial matrix factorizations in iterative methods", to appear in SIAM J. Numer. Anal.

(3) N.I. BULEEV, "A numerical method for the solution of two-dimensional and three-dimensional equations of diffusion", Mat. Sb., 51, 227-238 (1960).

(4) H.L. STONE, "Iterative solution of implicit approximations of multidimensional partial differential equations", SIAM J. Numer. Anal., 5, 530-558 (1968)

Problems Polynomially Equivalent to Graph Isomorphism

Kellogg S. Booth

Computer Systems Division
Lawrence Livermore Laboratory
Livermore, California 94550

Abstract

A language L_1 is polynomially transformable to a language L_2 iff there exists a deterministic polynomial time bounded algorithm, mapping strings w_1 over the alphabet of L_1 to strings w_2 over the alphabet of L_2, such that w_1 is in L_1 exactly when w_2 is in L_2. Two languages are polynomially equivalent iff L_1 is polynomially transformable to L_2 and *vice-versa*.

Two graphs are isomorphic iff there is a 1-1 mapping between the vertex sets which preserves the adjacency relation. A graph is regular if every vertex has degree r. A graph is an undirected-path graph if it has an intersection model all of whose sets are paths in some undirected tree. The problem of graph isomorphism is polynomially equivalent to the isomorphism problem for either regular graphs or undirected-path graphs.

It is already known that there is no loss of generality in restricting attention to digraphs, bipartite graphs, line graphs, transitively-orientable graphs, chordal graphs, rooted digraphs, or series-parallel graphs. The equivalence of undirected-path graph isomorphism to the general case is particularly interesting, however, because of the close relation between undirected-path graphs and interval graphs, for which there exists a linear time isomorphism algorithm. The isomorphism question for directed-path graphs remains open. (Directed-path graphs have intersection models whose sets are directed paths in some directed tree. They are a subset of undirected-path graphs and a superset of interval graphs.)

Previous results have shown that computing the automorphism partition (two vertices are in the same block of the partition iff one is mapped onto the other by some automorphism of the graph) is polynomially equivalent to graph isomorphism. Isomorphism for finite automata is also in the equivalence class with graph isomorphism, but this is a cheat, since graphs and automata are closely related anyway. It would be of interest to find other problems, not trivially related, which are equivalent.

There is an easy encoding which shows that group isomorphism (in fact, isomorphism for any algebra) is polynomially transformable to graph isomorphism, but a reverse transformation has not been established for groups. A new construction is given which transforms graph isomorphism to semigroup isomorphism, thus proving that graph isomorphism and semigroup isomorphism are equivalent.

It is conjectured that group isomorphism is not polynomially equivalent to graph isomorphism, but this may be hard to demonstrate, since it would imply that $P \neq NP$.

A Fast String Searching Algorithm

by

Robert S. Boyer[1]
J Strother Moore[2]

Abstract

An algorithm is presented that searches for the location, i, of the first occurrence of a character string, pat, in another string, string. During the search operation, the characters of pat are matched starting with the *last* character of pat. The information gained by starting the match at the end of the pattern often allows the algorithm to proceed in large jumps through the text being searched. Thus, the algorithm has the unusual property that, in most cases, not all of the first i characters of string are inspected. The number of characters actually inspected (on the average) decreases as a function of the length of pat. For a random English pattern of length 5, the algorithm will typically inspect i/4 characters of string before finding a match at i. Furthermore, the algorithm has been implemented so that (on the average) fewer than i machine instructions are executed. These conclusions are supported with empirical evidence and a theoretical analysis of the average behavior of the algorithm. The worst case behavior of the algorithm is linear in i.

---

[1]Computer Science Group, Stanford Research Institute, Menlo Park, Ca. 94025. This work was partially supported by ONR Contract N00014-75-C-0816.

[2]Computer Science Laboratory, Xerox Palo Alto Research Center, Palo Alto, Ca. 94304.

THE MULTIPLICATIVE COMPLEXITY OF A FAMILY OF BILINEAR FORMS CAN BE HIGH

Roger W. Brockett
Aiken Comp. Lab.
Harvard University

The purpose of this talk is to show that there exist families of bilinear forms which, in order to evaluate by the method:

(i) form linear functionals (no cost),
(ii) multiply pairs of linear functionals (unit cost per pair),
(iii) form linear combinations of the products (no cost),

have costs which grow nonlinearly with the size of the input data. In particular there exist families of p-bilinear forms on $R^m \times R^n$ whose cost is approximately $nmp/(n+m+p)$. It is also shown that this complexity is "typical" in a certain sense even though specific examples of such a family have apparently never been displayed and the most extensively studied cases, such as matrix multiplication, are known to have a much more modest rate of growth. The proof is based on elementary ideas in differential geometry.

OPTIMAL PMF-TRIE DESIGNS

W.A. Burkhard

Computer Science Division C014
University of California, San Diego
La Jolla, California 92093

Data base designs for retrieval from a file of records consisting of k secondary keys over a finite alphabet when queries may only be partially specified are examined. A large family of data base designs referred to as PMF designs which can yield data structures with good worst case and average case retrieval performances and yet require an amount of space essentially identical to that required of the records themselves is introduced and analyzed. The average case performance is completely discussed concluding with an explicit expression for the average. Moreover, it is observed that all k-key, w-bucket designs have identical average case retrieval performances.

PMF designs in which the worst case retrieval performance is as small as possible are evidently difficult to determine. A class of PMF designs which is based upon trie ideas is introduced. These PMF-trie designs allow extremely straightforward retrieval algorithms based essentially upon traversal of an implicit tree. A notion of optimal PMF-trie designs is promulgated which involves the distribution of attribute labels within the trie and a family of these tries is presented and analyzed. Indeed, the worst case retrieval performance of these tries is among the best known for any PMF design.

. . . . . .

W.A. Burkhard, Optimal PMF-trie designs. UCSD Computer Science Division, TR #8, (1975).

W.A. Burkhard, Non-uniform partial-match file designs. Theoretical Computer Science, submitted for publication.

W.A. Burkhard, Associative retrieval trie hash-coding, Proc. 8th Annual ACM Symposium on Theory of Computing, to appear.

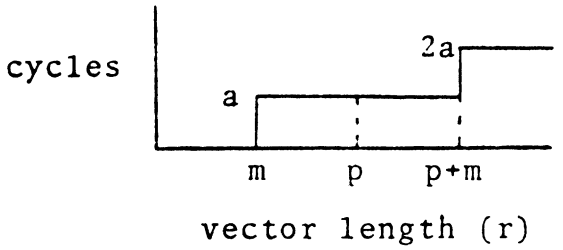
Abstract

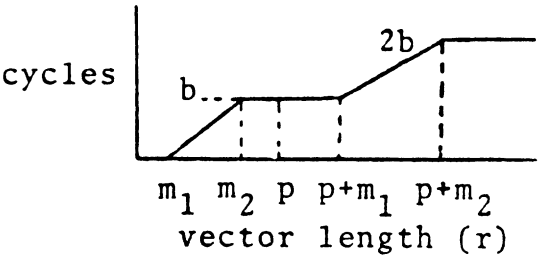
Complexity Analysis of Matrix Benchmarks on the Cray-1*

D. A. Calahan
Dept. of Elec. and Comp. Engineering
University of Michigan
Ann Arbor, Michigan 48109

Recent matrix benchmarks run on the CRAY 1 processor indicate that precise complexity analysis of such algorithms is possible due to the relative simplicity of rules for functional unit reservation and chaining and for control of the memory to processor pipeline.

With maximum vector lengths of p, the start-up, reservation, arithmetic processing, and storage times can be described by the two models of the figure below. Mathematically, it is





possible to describe these functions either in terms of ceiling functions (e.g., $\lceil \frac{r}{p} \rceil$) or modular arithmetic (e.g., $_p(r)$). Summation of these alternative representations, as required for a complexity study of vectorized matrix arithmetic, produces both "local" and "mod p" expressions. These complexity expressions are studied to deduce the effect of start-up and reservation times on the benchmark results.

---

*Sponsored by Directorate of Mathematical and Information Services, AFOSR, under Grant AFOSR 75-2812.

ABSTRACT

RAM: A GENERALIZED ALGORITHM FOR THE GENERATION OF VECTOR POINTS IN ARBITRARY BOUNDED N-DIMENSIONAL POLYHEDRAL SETS

by

J. R. Charnetski
State University of New York at Binghamton

A generalized algorithm, RAM, is developed to generate n-dimensional vector points which are elements of a prespecified but arbitrary, non-null polytope (bounded convex polyhedral set). From the construction of the algorithm, every vector point generated is an element of the specified polytope, regardless of the number and/or type of linear inequalities needed to define the polytope. Equality as well as inequality linear constraints may be used in defining the polytope. The primary use of such an algorithm is the evaluation of functions defined on the specified polytope. Using the algorithm, computing the expected values of functions on polytopes is greatly simplified, vis a vis computer time requirements. An extension of the algorithm to evaluate the expected values of a set of linear functions and a sample problem are provided.

ALGORITHMS FOR UPDATING MINIMAL SPANNING TREES

by

Francis Y. Chin and David J. Houck
University of Maryland

ABSTRACT

The problem of finding the minimum spanning tree on an undirected weighted graph has been investigated by many people and $O(n^2)$ algorithms are well known [1] ($O(e \log \log n)$ [2] for sparse graphs) where n and e are the number of vertices and edges in the graph. Spira and Pan [3] present an $O(n)$ algorithm for updating the minimum spanning tree if a new vertex is inserted into the graph. In this paper we present another $O(n)$ algorithm for the insertion of a vertex and show that a trivial extension of this leads to the result that if some of the edges adjacent to a vertex have decreased in weight, the minimum spanning tree can be updated in $O(n)$ steps. Spira and Pan [3] also show that $O(n^2)$ may be needed for updating the minimum spanning tree if a tree edge is deleted or increased in weight. Here, we present an $O(n^2)$ algorithm which labels all the tree edges in such a way that should any tree edge be deleted or increased in weight, the minimum spanning tree can be updated in constant time. Similarly the problem of updating the minimum spanning tree upon the deletion of a vertex from the graph may take $O(n^2)$ steps. Again, we show that this problem of deleting any vertex can also be done in $O(n^2)$ steps.

References

[1] Dijkstra, E.W., "A Note on Two Problems in Connection with Graphs", Numer. Math. 1, pp. 269-271 (1959)

[2] Yao, Andrew Chi-chih, "An $O(|E| \log \log |V|)$ Algorithm for Finding Minimum Spanning Trees," R-74-691 Dept. of Computer Science, University of Illinois, (1974).

[3] Spira, P.M., Pan, A., "On Finding and Updating Spanning Trees and Shortest Paths", SIAM Journal on Computing, Vol. 4, No. 3, pp. 375-380 (1975)

WORST-CASE ANALYSIS OF A NEW HEURISTIC FOR THE TRAVELLING SALESMAN PROBLEM

by

Nicos Christofides

Carnegie-Mellon University; on leave from Imperial College, London

ABSTRACT

An $O(n^3)$ heuristic algorithm is described for solving n-city travelling salesman problems (TSP) whose cost matrix satisfies the triangularity condition. The algorithm involves as substeps the computation of a shortest spanning tree of the graph G defining the TSP, and the finding of a minimum cost perfect matching of a certain induced subgraph of G. A worst-case analysis of this heuristic shows that the ratio of the answer obtained to the optimum TSP solution is strictly less than 3/2. This represents a 50% reduction over the value 2 which was the previously best known such ratio for the performance of other polynomial-growth algorithms for the TSP.

A UNIFYING MODEL FOR PARALLEL PROCESSING

by

A. Toni Cohen* and Leon S. Levy

University of Delaware
Newark, Delaware

Abstract

We introduce the sequential marked graph (smg), an extremely flexible model for depicting the flow of control through parallel processes. The representation is intended to serve as a unifying model, through which other, not-directly-comparable, graph formulations may be connected.

Two-way mappings between smg's and, respectively, computation graphs, UCLA graphs and parallel flowchart control graphs are defined and shown to preserve sequencing properties. Composition of these mappings enables graphs to be transformed from one model into another. Since smg's which cannot be simulated in less flexible models are mapped to the empty graph, we obtain (free) a decision procedure for determining whether a graph of one model may be simulated in another.

Several areas in which this research might be extended are discussed. In particular, we consider some potential benefits of relaxing the "normal" restraints on transfer of control.

*Presently at Lincoln University, Lincoln University, Pa.

A Broad Generalization of the Perceptron Convergence Theorem,

Bernard D. Coleman & Victor J. Mizel, Department of Mathematics, Carnegie-Mellon University, Pittsburgh, Pa. 15213.

Two disjoint sets G and P, lying on the surface S of the unit sphere in a Hilbert space, are said to be "W-separate" if the weak topology of S has two open sets O_1, O_2, with disjoint closures, such that $G \subset O_1$ and $P \subset O_2$. We here present a training program for constructing non-linear functionals which separate disjoint sets in S, and we show that when G and P are W-separate the program is finitely convergent. The functionals h produced by the program are, in an appropriate sense, of unit magnitude, and the program can be adjusted so that, after a finite number, N, of alterations of h, the program produces an h which separates G and P by a preassigned positive amount ε, in the sense that $h(\zeta)$ exceeds $\varepsilon/2$ for every ζ in G, and $h(\zeta)$ is less than $-\varepsilon/2$ for every ζ in P, where ε can be chosen arbitrarily between certain natural limits. An upper bound is given for the number N of alterations of h; this bound depends, of course, on properties of G and P and on the choice of ε.

Computer Generated Algorithms - A Constructive Approach to the Complexity of Bilinear Multiplication Operations*

by

Gerald Cooper and David Dobkin
Department of Computer Science
Yale University
New Haven, Connecticut 06520

Abstract

Issues involved in the use of a computer in the generation of efficient algorithms for bilinear multiplication operations are studied. Methods of reducing the complexity of algorithms are considered as well as methods of transforming algorithms into more transparent forms. Among the new results obtained in initial computer runs are algorithms which match known results for 2 x 2 and 3 x 3 matrix multiplication. In the latter case, the algorithms derived by computer are provably different from previously generated algorithms. We feel that further implementation of the techniques introduced here will yield significant improvements in algorithm design.

*This research was supported in part by the National Science Foundation Grant GJ-43157

Analysis of Optimal Variable-length Encoding Procedures

Norbert Cot

Dept. of Electrical Engineering
Stanford University

ABSTRACT

In recent work, we have exhibited efficient algorithms for generating optimal variable-length codes. These procedures have been proven to be linear in the case of binary as well as p-ary alphabets where symbols have different costs.

In this paper, we make a precise study of the complexity of these algorithms as a function of the cost/symbol distribution. We also present new results regarding the shape of the trees yield by the procedures and their weighted path lengths.

An Algebraic Approach to Concurrent Programming Control and Related Complexity Problems

Armin Cremers and Thomas N. Hibbard
University of Southern California

Short Tours of Sequential Machines

Paul Cull
Department of Computer Science
Oregon State University
Corvallis, Oregon 97331

Finding the shortest input sequence that takes a sequential machine through each state at least once is an NP-complete problem. However, we can obtain bounds on the lengths of these tours. Using these bounds, we can show that some simple algorithms find reasonably short tours, and in the worst case even find the shortest tour.

ALGORITHMS FOR POWER SYSTEM PROTECTION USING A MINICOMPUTER

by

Dr. P. K. Dash, Member IEEE
Department of Electrical Engineering
The University of Calgary
Calgary, Alberta, Canada

ABSTRACT

Various algorithms suitable for power system protection using a minicomputer are discussed in this paper. Among these the correlation techniques seem to be the best suited for the protection of power system elements, namely, the power generating units, transformers and long distance transmission lines with an on-line minicomputer. For differential protection schemes correlation between input and output quantities yields a very fast scheme of fault detection and tripping. However, for the protection of transformers and transmission lines correlation of voltage and current signals in the system with standard sine and cosine or square wave signals yields the desired protective functions. The time required for the detection of a fault in a power system and its subsequent tripping is of the order of half a cycle, based on a 60 Hz fundamental component. Assembler language is used to achieve real time operation for a laboratory power system using an on-line minicomputer. It is hoped that the algorithm will be subsequently implemented on a microcomputer, providing a cheaper and economic means for power system protection.

Abstract

An Algorithm for the Optimal Orientation of a Graph

R. W. Deming, State University of New York at Oswego

An acyclic orientation ω of a finite graph G is optimal if $\max\{|M|: M \in \mathcal{M}(\omega)\} \leq \max\{|M|: M \in \mathcal{M}(\omega')\}$ for all acyclic orientations ω', where $\mathcal{M}(\omega)$ is the family of assignments in $\overline{G}$, (G oriented by ω). It has been shown earlier by the author (and independently by D. J. Houck) that if ω is optimal then $\beta_o(G) = n(G) - \max\{|M|: M \in \mathcal{M}(\omega)\}$ where $\beta_o(G)$ and $n(G)$ are respectively the vertex independence number and number of vertices of G. In this paper an algorithm for computing an optimal orientation of a graph is given. The complexity depends on connectivity properties of edges in a perfect edge matching F. (The assumption that G admits a perfect edge matching can be justified.) If α is a subset of F, edges $\{x_1, x_2\}$, $\{y_1, y_2\} \in \alpha$ are α-biconnected if there are α-alternating paths $P(x_i, y_j)$ $i,j = 1,2$ beginning and ending with edges in $E \backslash F$. A polynomially-bounded algorithm is given which decomposes G into components $B_o, B_1, \ldots, B_k$ which for $i \neq j$, $e_i \in B_i \cap F$ is not F-biconnected to $e_j \in B_j \cap F$, any two edges in $B_o \cap F$ are not F-biconnected, and $\beta_o(G) = \Sigma \beta_o(B_i)$. If B_i is empty for $i \neq 0$, G can be oriented by a polynomially-bounded algorithm. In general, let $\mathcal{U}$ be the family of those subsets α of F which contain pairs of edges which are α-biconnected. If a minimum set of representatives of $\mathcal{U}$ is given, then an optimum orientation of G can be computed efficiently.

Complexity of Nearest Neighbour Searching in Three Dimensions

A. Keewatin Dewdney
Department of Computer Science
University of Western Ontario
London, Canada

The nearest neighbour problem in m dimensions may be stated geometrically as follows: given R^m, a metric d in R^n, and n points in R^n, construct a minimum-complexity algorithm for finding the closest (d-distance) point among these to an arbitrarily selected $n+1^{st}$ point. Here, "complexity" refers primarily to time and secondarily to space requirements for the algorithm. This problem is motivated by the need for efficient associative searching of an n-record file whose records are characterized by m attributes.

An algorithm which finds the nearest neighbour in R^2 in (worst case) time $O(\log n)$ and storage $O(n^2)$ was found by M.I. Shamos. In this paper we examine an extension of this algorithm to nearest neighbour searched in R^3 and point to a hierarchy of algorithms whose complexity increases as a joint function of N and p.

Expected Structure of Euclidean Graphs

Henry K. DeWitt
System Development Corporation

and

Michael M. Krieger[(a)]
University of California at Los Angeles

Analysis of graph theoretic algorithms has generally focused on worst case rather than expected behavior. Yet in a wide variety of applications, average running time is the most desirable parameter. To facilitate such analysis, it has been suggested that the work of Erdos and Renyi on random graphs be considered.

Our concern here is not with arbitrary abstract graphs, but rather geometric graphs: namely, those having vertices in some metric space with edges connecting vertices closer than a prescribed distance. Because of the structure imposed by the metric, algorithms exploiting the structure to achieve greater efficiency than more general algorithms are expected and sometimes found. To analyze such algorithms, results analogous to those Erdos and Renyi are desirable.

We deduce results for random 2-dimensional Euclidean graphs (graphs with vertices choosen in the plane by a Poisson process and edges subject to the Euclidean metric). Let G(N) be a random graph with N vertices, V(N) its average valence, and $C_k(N)$ the probability that such random graph has a connected component of k vertices. Then we prove the following:

Theorem 1: If $V(N) = O(N^{-1/(k-1)})$ then $C_k(N) = O(1)$ and $C_{k+1}(N) = o(1)$ as $N \to \infty$.

Theorem 2: If $V(N) = O(\log N)$, then $C_N(N) = O(1)$ as $N \to \infty$.

That these results agree with those for general graphs is surprising in view of the restriction on edges present in the graph. Potential applications of these results include radio packet switching networks and improved spanning tree algorithms.

(a) This research was supported by the Advanced Research Project Agency of the Department of Defense under Contract No. DAHC 73-15-C-0368 and by the National Science Foundation under Grant No. GK 42074.

AN APPROACH IN REDUCING COMPLEXITY OF NUMERICAL COMPUTATIONS

Miloš D. Ercegovac
Computer Science Department
School of Engineering and Applied Science
University of California, Los Angeles

Abstract

Complexity of a given method depends on the set of properties of the associated computational model. In the context of a particular implementation environment, the model properties should be defined to be compatible to the implementation properties in order to avoid a priori bounding of the complexity limits of any method, based on such model. In this paper a specific set of model properties, which forms the basis of a new efficient parallel computational method, is discussed. This method is applicable in problems such as the fast evaluation of polynomials, rational functions, integral powers or in solving a class of linear systems. The method requires only a single algorithm, i.e. solving of all problems in the domain of application is computationally isomorphic up to the initial conditions. The time is bounded by O(m) additive steps in computing the answers to an m-digit precision. The step-time is independent of the precision of the operands, while the number of processing units, operating in parallel, is only by one greater than the largest power involved or equal to the order of the linear system. Furthermore, the method has a non-decreasing efficiency factor with minimal interprocessor communication requirements and, unlike known parallel methods, no redundant operations.

LOWER BOUNDS ON ACTIVE OPERATIONS

C.M. Fiduccia
Department of Computer Science
SUNY at Stony Brook
Stony Brook, New York 11794

ABSTRACT

Several existing lower bounds on multiplications/divisions are shown (1) to apply to any set Ω of rules not subsumed by linear combination (including arbitrary partial non-deterministic n-ary operations), and (2) to be valid over any vector space V over any field F (not simply the often used space $F(x_1,\ldots,x_n)$ of rational functions).

Model: Given a subset G of V, generate the elements $h_1,\ldots,h_t$ of V, by constructing a sequence $v_1,\ldots,v_n$ such that, each v_k is obtained from G and $v_1,\ldots,v_{k-1}$ by using a rule in Ω, and $\{h_1,\ldots,h_t\}$ is a subset of $G\cup\{v_1,\ldots,v_n\}$.

Definition: Step k of the sequence is active if v_k is not in the subspace $S_{k-1} = \langle G\cup\{v_1,\ldots,v_{k-1}\}\rangle$ spanned by the existing pool of given or generated elements. Note that any rule subsumed by linear combination never yields an active step.

Without loss of generality, we may assume that G is a subspace of V and that $h_1,\ldots,h_t$ are independent modulo G, i.e. $c_1h_1+\ldots+c_th_t$ in G with c_i in F imply all $c_i=0$.

Lower Bound: Choose integers $0\le r\le s\le t$ and let $u_i=a_{i1}h_1+\ldots+a_{is}h_s$ $(1\le i\le r)$. If p or more active steps are necessary to generate $u_1,\ldots,u_r$ from $G\cup\{h_{s+1},\ldots,h_t\}$ for any row-reduced echelon $r\times s$ matrix A of rank r, then generating $h_1,\ldots,h_t$ from G requires at least $q \ge t+p-r$ active steps.

Note that $q\ge t$ by taking r=0 and s=t. Bounds as high as 2t-1 have been obtained by this method.

LOOP PROPERTIES and CONTROLLABILITY of LINEAR CONGRUENTIAL SEQUENCES

E.A. Galperin

School of Computer Science
McGill University
Montreal, Quebec, Canada

Loop properties of linear congruential sequences generated by $X_{n+1} = AX_n + C \pmod m$ are considered. Existence theorems and procedures are given to single out loops and fixed points, if any, in terms of generator parameters X_o, A, C, m. On the basis of the fundamental sequence $0, 1, 1+A, 1+A+A^2, \ldots \pmod m$, introduced by G. Marsaglia /[1]/, criteria are found for absolutely and conditionally convergent generators and split generators.

Main result. The following statements are equivalent:

(i) $A^p = 0 \pmod m$ for some p, - a generator is absolutely convergent, i.e. converges to unique fixed point whatever X_o.

(ii) $\gcd(A,m) = 1$, - a generator is split, i.e. produces pure loops only.

(iii) $\gcd(A,m) > 1$ & $A^p \neq 0 \pmod m$ for any p, - a generator is conditionally convergent, i.e. rotates in or converges to a loop depending on X_o.

There exist only these three types of inner structure.

The role of parameters in respect of inner structure is as follows : the modulus m defines the set of structural types (prime m = split, a power of one prime = split or absolutely convergent, otherwise = all 3 types); the multiplier A establishes the inner structure among the types mentioned; the additive constant C delivers the set of subsequences existing within the structural type; the initial value X_o singles out a particular sub - sequence to appear.

To improve quasirandom features and overcome nonrandom appearance for amounts greater than the modulus applied, there are introduced non-autonomous random number generators. They are shown to be completely controllable in the sense that any prescribed sequence of numbers can be obtained by means of a non-autonomous linear congruential random number generator. As an application, a generator $(\text{mod } 2^{32})$ is constructed having a good lattice, a period greater than $2^{600} > 10^{180}$ and no translated-block structure.

Reference

1. G.Marsaglia, The Structure of Linear Congruential Sequences.
Applications of number theory to numerical analysis,
Academic Press, Inc., New York and London, 1972, p.p. 249-285.

Analysis of Heuristics as Functions

John Gaschnig
Carnegie-Mellon University

Many problems can be solved in terms of finding a path between an initial node and a goal node in a graph that represents a state space. An exhaustive search typically examines a number of nodes exponential in the length, k, of the solution path. A search guided by the use of heuristic functions is usually "much faster". How much faster? A formal, domain-independent model is defined, consisting of a tree-like state space, the A* search algorithm, and heuristics defined as functions F from natural numbers to reals. A typical goal in complexity research is, given a particular problem parameterized by the size of input k, to determine the amount of time required by a particular algorithm to solve the problem, as a function T(k). In the present model, each heuristic function f (i.e., each element of F) defines a distinct algorithm, with corresponding time complexity function $T_f(k)$. Thus the model defines a mapping M:F-->T(k) from heuristic functions to time complexity functions. In other words, a goal of this research is to perform a complexity analysis not for a single algorithm, but rather for a whole class of algorithms.

Upon what properties of f does the complexity function $T_f(k)$ depend? It is shown, for an interesting subset of F, that in a search using a heuristic function f, the number of nodes expanded in the worst case, i.e., the function $T_f(k)$, depends in a precise way upon the accuracy of f as a distance estimator, suitably defined.

Solving problems would be easy if, given a particular problem, there was a way to find a heuristic for which the number of nodes expanded is linear in k. This goal is approached in terms of the model by defining the set of "perfect" heuristics to be the subset of F for whose elements T(k)=k. Some of the members of this subset are identified, by the use of an equivalence relation defined on the heuristic functions. Empirical evidence is presented, by which to assess the validity of the model. Open questions and extensions to the model are suggested, which may yield useful practical applications of the theoretical results.

Some Complexity Results for Matrix Computations on Parallel Processors

W. Morven Gentleman*

University of Waterloo

Waterloo, Ontario, Canada.

Abstract

In this paper it is shown how data movement, rather than arithmetic operations, can be the limiting factor in the performance of parallel computers on matrix computations. In particular it is proved that for machines with two dimensional rectangular grid connectivity (such as ILLIAC IV), multiplication and inversion of N x N matrices inherently require O(N) steps, even if the processing elements are not constrained to execute identical instructions.

---

* This research was carried out while the author was on sabbatical at the National Physical Laboratory, Teddington, England.

Analysis of the Spectrum of Knee Generated Noises

Peter J. Gingo
University of Akron

Noises generated by the human knee rotating in a manner which is consistent with walking has become an extremely powerful tool for evaluating the physical condition of the knee. The method is used to determine whether the human knee is healthy, rheumatoid or arthritic.

The noise amplitudes developed by a moving knee are recorded, digitized and stored on a computer tape in the Binary mode. The Binary data is reread and converted into the decimal mode to undergo a series of computerized calculations.

The noise amplitude vs. time is plotted for an initial survey. The noise amplitude is then transformed into the frequency domain using a Fast Fourier Transform which was again plotted. It was discovered that there were some very low random noises as well as 60 cycle, 120 cycle and 180 cycle resonant frequencies superimposed on the noise spectrum. A Chebyshev high pass filter algorithm was written to filter the low frequency noise. The resonant frequencies were smoothed by interpolating routines.

The low frequency noise was inherently induced because of the mechanical instrumentation while the 60 cycle harmonics are obviously due to the power supply.

The inverse transform was taken on the reconstructed frequency spectrum which was now corrected by the filtering and smoothing.

TITLE: A Comparison of Graph Isomorphism Algorithms

AUTHORS: C. G. Goldbogen and G. H. Williams, Union College

ABSTRACT:

This paper presents a comparison of the running time for a number of graph isomorphism algorithms. Included are the Ullmann [1], Corneil-Gotlieb [2], Berztiss [3], and Goldbogen [4] algorithms. The graphs considered were randomly generated and divided into several categories according to their properties: directed, undirected, regular, strongly regular, and connectivity. Within each of these categories the relationship between running time and the size of the graph (number of nodes) is established empirically.

REFERENCES

[1] Ullman, J. R., "An Algorithm for Subgraph Isomorphism" JACM, Vol 23, No. 1, 1976.

[2] Corneil, O. G. and C. C. Gotlieb, "An Efficient Algorithm for Graph Isomorphism," JACM, Vol. 17, No. 1, 1970.

[3] Berztiss, A. T., "A Backtrack Procedure for Isomorphism of Directed graphs," JACM, Vol. 20, No. 3, 1973.

[4] Goldbogen, C. G., To be submitted.

THE COMPLEXITY OF COMPARABILITY GRAPH RECOGNITION AND COLORING

Martin Charles Golumbic
Courant Institute of Mathematical Sciences
New York University

In Golumbic [1,2] the notion of G-decomposition was presented in order to describe a new matroid associated with any undirected graph. It was shown that the length of a G-decomposition equals the rank of the matroid and that a set of representatives, called a scheme, constitutes a basis of the matroid. A new characterization of comparability graphs was given including a method for producing a transitive orientation and counting the number of such orientations.

In this paper we move from the strict algebraic setting into an algorithmic presentation of these terms. The computational complexity of recognizing comparability graphs is discussed. An implementation is shown for an algorithm which assigns a transitive orientation in $O(d \cdot |E|)$ time and $O(|E|)$ space where d is the maximum degree of a vertex and $|E|$ is the number of edges. Another section deals with efficient solutions to the clique problem, minimum coloring problem, et al. for comparability graphs. These problems are NP-complete for arbitrary graphs, but polynomial solutions can be found for many special classes of graphs by exploiting their particular structure. (See Gavril [3] for a similar treatment of chordal graphs.)

1. Martin Charles Golumbic, Comparability graphs and a new matroid, Ph.D. thesis, Columbia University (1975), to appear in J. Comb. Th., Series B.

2. Martin Charles Golumbic, Comparability graphs and a new matroid, (extended abstract), Proc. Conf. on Algebraic Aspects of Combinatorics, Univ. of Toronto, Jan. 1975, Utilitas Math, Winnipeg (1975).

3. Fanica Gavril, Algorithms for minimum coloring, maximum cliques and maximum independent set of a chordal graph, SIAM J. Comp., 1 (1972), 180-187.

An Algorithmic and Complexity Analysis of the Heap as a Data Structure

by
Gaston H Gonnet
and
Lawrence D Rogers

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Abstract

A heap is defined as a class of data structures used to represent a partially ordered set of data from which we can repeatedly and efficiently extract the maximum (or minimum) element. The definition of the class covers heaps of integer as well as real branch factor. We define and describe the basic operations on a heap and evaluate the running times of various algorithms to perform these operations. The complexity of a heap is defined in a way that proves to be a useful tool for evaluation of running times. We give, in detail, an average case analysis of these algorithms as well as considering the worst case. A relation between probabilistic models of heaps is given and various comparisons of the running times are presented for the most common heap structures. In addition we consider some less common heap structures and describe one heap which resembles the tournament selection process.

Key words and phrases: Data structures, heap, trees, priority queues, analysis of algorithms, complexity analysis, maximum-minimum extraction.

On updating shortest paths and a decomposition algorithm

S. Goto *

Central Research Laboratories, Nippon Electric Co., Ltd.
Kawasaki, Japan

Abstract

New updating and decompotion algorithms for shortest path problem are presented.

First, the problem of updating shortest paths from a set of specified vertices to all vertices following a decreasing-lenght-modification of some edges, is discussed. We shall propose a new algorithm that is based on a formula which has the algebraic analogy with the well-known Householder's formula for inverting modified matrices. The number of operations(i.e., additions and comparisons) required for solving the modified shortest path problem is estimated as

$$\alpha n^2 + \left((q+\beta)m + q(\beta+1) \right)_n + qm^2 + \frac{1}{3}m^3$$

where n is the overall number of vertices, q the number of specified vertices, $m = \alpha + \beta$ and α and β are the number of columns and rows, respectively which together contain all modified elements (edge-lengths) in the vertex-vertex weighted adjacency matrix.

Secondly, a new decompotion method based on the preceding algorithm is proposed. The computational complexieties are evaluated for some commonly encountered structures such as ladder or star networks.

Finally, the results obtained by using the proposed method are compared with oter existing algorithms.

---

* On leave at Department of Electrical Engineerig and Computer Sciences University of California, Berkeley, California 94720

PROBABILISTIC COST MACHINES

M.G. Gouda and E.G. Manning
Computer Communications Networks Group
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

A PCM is a directed graph with single entry and exit nodes such that:

(i) Each node is on a directed path from the entry node to the exit node.

(ii) Each edge is labelled with a (probability, cost) pair, where the probabilities of the output edges at any node are added to 1.

Most algorithms can be abstracted as PCM's. The probability label for each edge is the probability that the control flows over the specific edge given that it reaches its tail node. The cost label is the cost paid whenever the control flows over the edge.

In the paper we present isamorphism - and equivalence - preserving transformations that can be used to transform any PCM to a normal form that has only one edge labelled with (1,C). In that case we conclude that C is the average cost of the original PCM. The variance of the cost can also be obtained.

The model is useful in developing complexity measures for algorithms, which are based on average behaviour rather than the traditional worst-case analysis.

Symbolic and Very High Precision Optimization of Constrained Systems of Equations by Newton's Method.

James Hanson, Professor and Phillip Russo, Asst. Professor, Computer and Information Science Department, Cleveland State University, Cleveland, Ohio.

Preprocessing, PL/I and Formac have been used to provide a flexible code for static optimization of diverse problems. Newton's Method has been employed because of its potentially rapid convergence and in order to comprehensively test its applicability to a wide range of problems (i.e. to test the feasibility of symbolically constructing the Hessian matrix and iteratively solving a linear system of the same). In addition, statistics on efficiency, storage requirements, computing time and precision have been assembled as well as convergence characteristics as compared with theoretical rates and convergence regions (e.g. those derived by Kantorovich). The main feature of this code is its ability to preserve precision, even very high precision of several hundred significant digits, by appropriate mode conversions during execution. Four computational modes, i.e. symbolic, rational or extended floating point, floating point (double precision) and float-to-rational, are available to this optimizer in order to achieve various solution forms and in order to exert control over the computational processes and thereby insure appropriate accuracy. This is accomplished through functions TRUNCATE (which suppresses the precision of a rational number thus averting stringsize overflow), FLOATRATIONAL (which converts a floating point number to rational form for subsequent rational arithmetic thus eliminating floating round-off. This is especially important in the linear system solution for a large number of variables) and EXTENDEDFLOAT (which converts a rational number to floating point form of arbitrary precision). The symbolic mode finds application in developing an analytical approximation of one or two iterations around some set point, and for analytically assembling the final symbolic iteration from symbolic input, e.g. analytically developing the Chebyshev norm. The rational mode permits the intermitant or continued computation by rational numbers whose total length are continually limited to a prescribed precision thus permitting percision of several hundred decimal digits and thereby, to a very large extent, avoiding the insurmountable obstacles of floating point fixed precision. When symbolic or rational (or extended float) modes are not invoked computation defaults to double precision floating point. Equality constraints have been included by sum-of-squares penalty and inequality constraints by the Fiacco-McCormick penalty. Gaussian elimination has been chosen as the linear equation solver, since this phase of the computations will usually be done in rational mode in which one finite method is as good as another. A large and wide range of applications have been tested: Himmelblau's problems (Himmelblau, Applied Nonlinear Programming, 1972), minimizing Chebyshev and other norms for obtaining rational functions and other approximation forms for special functions, large-difficult polynomials (Dejon & Henrici, Constructive Aspects of The Fundamental Theorem of Algebra, 1968), and objective functions containing derivatives, substitutions and jump discontinuities, et al.

An $O(n^2 \log_2 n^2)$ Lower Bound to the Complexity of Matrix Multiplication

by

L. H. Harper*
Department of Mathematics
University of California, Riverside
Riverside, CA

J. E. Savage†
Division of Engineering
Brown University
Providence, RI

Abstract

We develop an $O(n^2 \log_2 n^2)$ lower bound to the synchronous combinational complexity of matrix multiplication over GF(2). Our method of proof can be readily applied to a number of other interesting problems.

A synchronous combinational machine is a logic circuit in which unit delays have been introduced where necessary so that the lengths (number of elements) on all paths to each logic element are the same. Thus, inputs simultaneously applied arrive simultaneously at a logic element. The synchronous combinational complexity of a function $f:\{0,1\}^n \to \{0,1\}^m$ relative to a basis Ω of Boolean functions, $C_\Omega^S(f)$, is the minimum number of logic elements from Ω and delay elements in a synchronous combinational machine for f. Synchronous combinational complexity is at worst a polynomial of degree two in standard combinational complexity (only logic elements are counted).

We define classes $P_{p,k}^{(n,m)}(\varepsilon)$ of functions $f:\{0,1\}^n \to \{0,1\}^m$ with the property that all but a fraction ε, $0 < \varepsilon < 1$ of the p-subsets $\{i_1,\ldots,i_p\} \subset \{1,2,\ldots,n\}$, the functions have at least k subfunctions $f \Big|_{1 \le j \le p} x_{i_j} = c_j$ as $(c_1,\ldots,c_p)$ ranges over $\{0,1\}^n$. We show that if $f \in P_{p,k}^{(n,m)}(\varepsilon)$ then $C_\Omega^S(f) \ge (1-\varepsilon)\left[L - \frac{4(n-p)(2^L-1)}{p-2^L}\right] \log_2 k$ where L is the largest integer satisfying

$$(1-\varepsilon)\left[1 - \frac{2(n-p)2^L}{p-2^L}\right] \log_2 k \ge m$$

We show that matrix multiplication of rxr matrices over GF(2) is in $P_{(m,2^m)}^{(2r^2,2r^2)}(\varepsilon)$ where $m = 2r^2 - 2r(K+\log_2 r)$ and $\varepsilon = 22^{-K}$. From this we have the stated lower bound. Strassen's algorithm can be realized by a synchronous combinational machine of size $O(n^{\log_2 7})$.

This work is an extension of earlier work by L. H. Harper (A nlogn Lower Bound on Synchronous Combinational complexity, to appear in Trans. AMS).

*Supported in part by NSF Grant GJ-42907
†Supported in part by NSF Grant DCR72-03612-A01

PROPERTIES OF SOME MODIFIED INTERPOLATORY ITERATION FUNCTIONS

J. Herzberger
Universität Karlsruhe, West Germany

Abstract: The purpose of this note is to present a family of iteration functions for bracketing a zero α of a real function. These methods are based on the well-known class of interpolatory iteration functions. They can be considered as slight modifications of these classical methods requiring only a small amount of additional arithmetic costs. As payoff we get properties like (i) global convergence behavior (ii) monotone sequences of bounds for α and (iii) automatic round-off error estimates at each step. The order of convergence can be shown to be the same as for the unmodified functions.

Title: Optimal Algorithms for Quarternion Multiplication

Author: Thomas D. Howell
Cornell University

Abstract:

It has recently been shown that eight ring multiplications are necessary and sufficient to multiply two quarternions over an arbitrary (possibly noncommutative) ring. The original proofs of the lower bound by de Groote and independently, by Howell and Lafon were somewhat complicated and technical. Some more recent results by other authors allow these proofs to be shortened considerably.

A simple proof of the lower bound is given. Also presented is a modified algorithm which achieves the lower bound of eight multiplications and which has some computational advantages over the most familiar eight-multiplication algorithm.

A DIVERSE SHORTEST PATH PROCEDURE

by

W. Hsieh and A. Kershenbaum

Network Analysis Corporation
Beechwood, Old Tappan Road
Glen Cove, New York 11542

ABSTRACT

Given a (undirect) labeled graph G and a distinguished node s in G, suppose we have already found the shortest-path tree T from s to all other nodes. We consider here the problem of finding the shortest paths from s to other nodes which are respectively diverse from the primary shortest paths.

Based on a particular end order of the tree T, an efficient procedure is developed for finding the diverse shortest-paths from s to all other nodes. In this procedure a standard shortest-path algorithm is only applied as many times as the number of next-to-end nodes in T, and only on subgraphs of usually much reduced size.

Finding the K-th Largest Pair is $\Omega(\max(n, \sqrt{K} \log K))$

Donald B. Johnson and Samuel D. Kashdan
Computer Science Department
The Pennsylvania State University

Abstract

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of n distinct real numbers, let $A^2 = \{(a_i, a_j) \mid a_i, a_j \in A \text{ and } a_i > a_j\}$, and let $s = |A^2| = \frac{n^2-n}{2}$. Given any decision tree, incorporating tests on linear forms over A, which finds the K-th largest pair (a_{i_K}, a_{j_K}) in some ordering $a_{i_1} + a_{j_1} \geq a_{i_2} + a_{j_2} \geq \ldots \geq a_{i_K} + a_{j_K} \geq \ldots a_{i_s} + a_{j_s}$ of A^2, a decision tree of virtually the same worst case complexity can be found which sorts certain sets of size $O(\sqrt{K})$. It follows that finding the K-th largest pair is $\Omega(\sqrt{K} \log K)$, where by $\Omega(f(n))$ is designated any function which is greater than $cf(n)$ for some positive c and all but finitely many n. The $\Omega(n)$ bound is obtained through a similar construction and the application of results by Rabin regarding the width of a complete proof for the largest pair. The results extend to the analogous problem on integers and apply also to the problem of finding the K-th largest pair in $X + Y$ where X and Y are sets of real numbers.

Long and Short Walks in Trees

Paul C. Kainen
Department of Mathematics
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

Given a graph G and a labeling λ of its vertices, say $v_1,\ldots,v_n$, define $h(\lambda)$ to be the sum $\sum_{1\leq i\leq n} d(v_i, v_{i+1})$, where $v_{n+1} = v_1$ and $d(v_i, v_{i+1})$ is the distance from vertex v_i to v_{i+1}. Let $h(G)$ and $H(G)$ denote the minimum and maximum, respectively, of $h(\lambda)$ over all labelings λ of G. For G a tree, we give algorithms for computing $h(G)$ and $H(G)$ and for finding the corresponding labelings. An application is made to the Traveling Salesman Problem.

Infinite Precision Integer Arithmetic in Base 30.

Dorothea A. Klip
Department of Physiology and Biophysics
Department of Information Sciences

Abstract

It was found that extension of the principle of the binary GCD algorithm to an arbitrary base b led to a substantial reduction in number of iterations to obtain the GCD as compared with the traditional Euclid's procedure. If the integers N and M (N > M) do not contain factors which divide the base b, then a single digit k can be chosen such that N - kM is divisible by b. Removal of one (or more) factor(s) d, which divide b, from (N - kM)/b then transforms the pair (N,M) into the equivalent pair ((N -kM)/(bd),M). Arguments aimed at obtaining the greatest reduction in number of iterations, while at the same time minimizing the effort per division step, point to 30, which is the product of the three smallest primes, as the optimal value for b, when actually implementing the algorithm. For this value one needs only 1/5 of the number of iterations for sequential Fibonacci numbers, whereas 1/3 is conjectured to be the average for arbitrary number pairs.

When choosing the set of residues mod 30 symmetrically with respect to 0, representation of the digits from this set in factored form further enhances the efficiency of the GCD algorithm. Distinction of two classes of digits, namely those whose prime factors divide the base and those which contain additional prime factors, leads to the construction of an identification tree such that a digit can be identified--apart from its sign--in an average of 4.2 steps, involving 6.7 bit comparisons. Digit-to-digit multiplication and addition are suggested by means of an ordered-pair identification table constructed on this basis. In the case of multiplication, the number of bit comparisons is approximately 1/3 of that for standard binary multiplication on IBM 360/370 computers, if applied to 5-bit entities.

With respect to division in the proper sense, the first step in the GCD procedure finds the quotient, in case the divisor divides the dividend. Otherwise, this first step has to be followed by the reduction mod the divisor of a 'remainder', which mainly consists of a power of 30. This second step therefore actually amounts to generating the divisor's reciprocal. Since the first step requires only 68% of the number of division steps of traditional division and has moreover the computational complexity of addition, it appears worthwhile to attempt a time analysis for division on this basis.

FAST ALGORITHMS FOR PARTIAL FRACTION DECOMPOSITION

H. T. Kung
Carnegie-Mellon University
Pittsburgh, Pa. 15213

D. M. Tong
The University of Akron
Akron, Ohio 44304

ABSTRACT

The partial fraction decomposition of a proper rational function whose denominator has degree n and is given in general factored form can be done in $O(n \log^2 n)$ operations in the worst case. Previous algorithms require $O(n^3)$ operations, and $O(n \log^2 n)$ operations for the special case where the factors appearing in the denominator are all linear.

FAST ALGORITHMS FOR ALGEBRAIC FUNCTIONS

H. T. Kung
Department of Computer Science
Carnegie-Mellon University

J. F. Traub
Department of Computer Science
Carnegie-Mellon University

Let

$$P(w,z) = A_0(z)w^n + \ldots A_n(z), \tag{1}$$

where the $A_i(z)$ are polynomials in z. Functions $w = w(z)$ for which $P(w(z),z) \equiv 0$ are called algebraic functions. In general, algebraic functions can be represented by fractional power series. Algebraic functions play an important role in many parts of mathematics including classical algebraic geometry, elliptic integral theory, complex analysis, and theory of plane curves. Indeed algebraic function theory is a major subject in its own right.

We show that the coefficients of an algebraic function can be computed "fast". We consider both regular and singular points. The complexity analysis depends on whether we wish to compute the coefficients numerically or symbolically. Dropping out as special cases of our analysis are results of Kung on reciprocation of polynomials and results of Brent and Kung on reversion of polynomials. The reciprocation problem is (1) with $n = 1$, $A_1(z) = 1$ while reversion is (1) with $A_i(z) = A_i$ (independent of z), $i = 0,1,\ldots,n-1$, and $A_n(z) = -z$.

Applications and extensions to the theory will be indicated.

ALGORITHMS FOR MATRIX PRODUCTS USING FEWER MULTIPLICATIONS

by Julian David Laderman

Lehman College, Bronx, N.Y.

Abstract: A technique is developed to study the structure of algorithms used in matrix multiplication and to create new algorithms requiring fewer multiplications. By this method an integral solution of a system of 729 nonlinear algebraic equations in 621 unknowns is obtained which yields an algorithm for multiplying two 3x3 matrices using only 23 multiplications. The system solved by the technique is

$$\sum_{t=1}^{23} z_{ikt}x_{ght}y_{rst} = \delta_{ig}\delta_{hr}\delta_{sk} \qquad \text{for } i,k,g,h,r,s=1,2,3.$$

The algorithm, without presenting the technique used to obtain it, appeared in an article by the author in the January 1976 issue of the Bulletin of the American Mathematical Society. The title of the article is "A Noncommutative Algorithm for Multiplying Two 3x3 Matrices in 23 Multiplications". The algorithm does not require that the matrix elements be commutative under multiplication. Such an algorithm has the advantage that it can be used to multiply two matrices whose elements are matrices. By the use of partitioning the algorithm enables the creation of algorithms with fewer multiplications than any known algorithms for the product of two nxn matrices for an infinite number of values of n (including $n=5,6,7,$ and 9).

THE COMPLEXITY OF MONOTONE NETWORKS FOR CERTAIN BILINEAR FORMS, ROUTING PROBLEMS, SORTING, AND MERGING*

Edmund A. Lamagna
The Mitre Corporation
Bedford, Massachusetts 01730

Abstract

Recent research efforts in the field of computational complexity have been directed toward the development of nontrivial lower bounds on the amount of resource required to compute specific functions. For example, one might inquire into the fewest number of logical operations needed to synthesize a combinational switching network for a given function, where the operations are selected from some functionally complete set of primitives (i.e., a set of operations, such as {*disjunction*, *conjunction*, *negation*}, from which every Boolean function can be realized). Counting arguments can be used to show that almost all Boolean functions of n inputs and $O(n)$ or fewer outputs require networks of size growing approximately exponential in n. However, despite much research effort, no one has yet exhibited a particular n-input function whose realization is known to require a number of operations greater than linear in n.

In view of this difficulty, several researchers have recently begun investigating the restricted problem of determining the complexity of monotone increasing Boolean functions in networks built from the incomplete set of primitives {*disjunction*, *conjunction*}. This line of research is advanced here through the development of a general technique, which can be used to derive $O(n \log n)$ lower bounds on the logical complexity of monotone switching networks for the following functions:

1. certain bilinear forms, including Toeplitz and circulant matrix-vector products, and Boolean convolution
2. certain routing problems, including cyclic and logical shifting
3. sorting and merging

Using a homomorphic mapping technique, the lower bounds derived on the complexity of monotone switching networks for Boolean functions can be extended to a larger class of problems. Included are the corresponding functions computed over the following algebraic systems:

1. the operations {*maximum*, *minimum*} over the real numbers
2. the operations {*minimum*, *addition*} over the nonnegative reals
3. the operations {*addition*, *multiplication*} over the nonnegative reals
4. the operations {*union*, *concatenation*} over the family of sets of finite length strings of symbols from some finite alphabet

* This research was performed under the support of National Science Foundation Grant GJ-32270 while the author was a graduate student at Brown University, Providence, Rhode Island 02912, and constitutes a portion of his Ph.D. dissertation entitled "The Complexity of Monotone Functions."

An o(e log n) Shortest Path Algorithm for Sparse Graphs

Laurence L. Lang and J. Denbigh Starkey
Computer Science Department
Washington State University
Pullman, WA 99163

Abstract

An o(e log n) algorithm will be presented which calculates the shortest path between one selected start node and all the other nodes in a graph with non-negative weights. For sparse graphs this will be shown to be a considerable improvement over traditional techniques such as Dijkstra's algorithm, which are $o(n^2)$. Theoretical and experimental comparisons with Dijkstra's algorithm will be given.

Why binary insertion sorting and tree selection sorting require the same number of comparisons

Shuo-Yen Robert Li
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Mass. 02139

ABSTRACT

Binary insertion and tree selection are two independent sorting methods and surprisingly they require exactly the same number of comparisons in sorting any numbers of elements. We provide a simple reason to show that this coincidence is not accidental. On the other hand, analysis of both methods indicates their difference in the expected number of comparisons and in the probability for the worst case to occur.

FEWEST-BIT ENCODING OF PLANE TREES

Rici Liknaitzky and W. J. Meyers

The number of structurally different rooted plane trees with n points ($n > 0$) is the n-th Catalan number $T(n) = \binom{2n-2}{n-1}/n$. The number of bits necessary to distinguish such trees is the binary logarithm $\lg(T(n))$. We derive closed form and asymptotic expressions for this logarithm. The closed form expression provides an elementary proof that $\lg(T(n)) < 2n$. The asymptotic expression is $2n - 3/2 \lg(n) - (2 + 1/2 \lg(\pi)) + O(1/n)$. We then investigate methods of encoding plane tree structure compactly. A minimal encoding is presented, as a correspondence between n-point plane trees and the integers $0, 1, \ldots, T(n) - 1$. We also survey and extend the conventional methods of encoding n-point plane trees using sequences of 2n or fewer bits. These encodings use only $O(\lg(n))$ more bits than the minimum, and provide greater versatility. They exhibit two general styles: one well suited to standard word organized memories, the other to bit oriented memories such as large shift registers. Each style allows wide variation in the choice of encoding and decoding algorithms, and means of incorporating other data into the tree structure. Together they include all previously known and many new ways of encoding n-point plane trees using 2n or fewer bits.

Keywords: plane tree, Catalan number, binary logarithm, ballot sequence, representation, notation, encoding, packing.

CR Categories: 3.70, 4.0, 4.34, 5.0, 5.25, 5.32.

COMPUTATIONAL COMPLEXITY OF GROUP-THEORETIC PROBLEMS

R.J. Lipton
Computer Science Department
Yale University
New Haven, Connecticut 06520

Y. Zalcstein
Computer Science Department
State University of New York
Stony Brook, New York 11794

ABSTRACT

In this paper we begin the study of computational complexity of algorithmic problems in group theory. There are several specific reasons for studying these questions:

1. The word problem is closely related to language recognition problems and in fact one of our results solves an open problem on language recognition.

2. The word and isomorphism problem arise in many applications ranging from chemistry to the structure theory of finite simple groups. Furthermore, current research in finite simple groups involves computation with groups of very large cardinality where asymptotically optimal results have a distinct advantage.

A presentation of a finitely generated group G is an ordered pair (X,D), where $X = \{x_1,\ldots,x_m\}$ is a finite set (of generators) and D is a set of words over $X \cup \{x_1^{-1},\ldots,x_m^{-1}\}$ such that G is isomorphic to the quotient group formed by the free group on X modulo the normal subgroup generated by the words in D. A presentation of a semigroup is defined in a similar manner.

Let (X,D) be a presentation of a group G. The word problem for (X,D) is the problem of deciding whether an arbitrary word w over the alphabet $X \cup \{x_1^{-1},\ldots,x_m^{-1}\}$ reduces to the identity element of G. As is well known (see e.g., [4, Ch. 12]) the word problem for groups is unsolvable.

Theorem 1. If a finitely generated group G is isomorphic to a group of matrices over a field then the word problem for G is solvable by a deterministic Turing machine of logspace complexity.

Corollary 1. The word problem for finitely generated free groups is solvable in logspace.

Theorem 1 extends a result of Rabin [3] who showed that the word problem is solvable for groups of matrices over a field. Corollary 1 implies an affirmative answer to a question raised by Albert Meyer whether the two-sided Dyck language has logspace complexity. Furthermore, since DLOG is a subset of $\mathcal{L}_*^o$, Corollary 1 provides an affirmative answer to an open problem of Cannonito [1].

FAST ALGORITHMS FOR

COMPLEX INTEGER TRANSFORMS

K. Y. Liu, I. S. Reed, and T. K. Truong
University of Southern California
Department of Electrical Engineering
Los Angeles, California 90007

ABSTRACT

Recently it was shown that an algorithm similar to the fast Fourier transform (FFT) exists over the Galois field $GF(q^2)$, a finite field analogous to the complex number field, when $q = 2^p - 1$ is a Mersenne prime. Digital filtering can be achieved without round-off error, using such transforms. It is shown in this paper that the binary arithmetic in $GF(q^2)$ is simpler than complex number arithmetic and also that the d^{th} roots of unity in $GF(q^2)$, where d divides 2^{p+1}, possess similar symmetry properties. Therefore, algorithms analogous to the conventional twiddle factor FFT algorithms can be used to compute the FFT in $GF(q^2)$. Furthermore, because the components of $\gamma^{kd/8}$, where γ is a primitive d^{th} root of unity in $GF(q^2)$ and k is an odd integer, are fixed powers of 2, complex multiplications involving $\gamma^{kd/8}$ can be performed merely by additions and circular shifts of $\frac{p-1}{2}$ bits in a p-bit register. Hence, the FFT algorithms over $GF(q^2)$ can be accomplished faster, simpler, and more accurately than the conventional FFT algorithm.

Finding the Kth pair efficiently

Tetsuo Mizoguchi

Electronics Laboratory, Mitsubishi Electric Corporation
325 Kamimachiya, Kamakura JAPAN

Abstract

The pair selection problem is, given a set $\{a_i\}$ of n distinct numbers and an integer K, to find the pair (a_p, a_q), $p \neq q$ such that the value (a_p+a_q) is the Kth largest.

A nlogn pair selection algorithm is given here.
i) Sort the set $\{a_i\}$ (assume matrix representation of pairs),
ii) find the median of each row, iii) find the median m of the median set given by ii) and divide the median set into two by m, iv) if the numbers of pairs belong to these two sets are unbalanced then repeat iii) on the subset until balanced, v) divide the pair matrix by the weight-balanced center, vi) repeat ii) through v).

The successive applications of median selection on the subset whose size is one half require O(n) time. Since there are O(logn) repetitions of ii) through v), this algorithm selects the Kth pair in O(nlogn) time.

The Efficiency of the Alpha-Beta Search on Trees with Branch-dependent Terminal Node Scores

by

M.M. Newborn
School of Computer Science
McGill University
Montreal, Quebec

A B S T R A C T

An analysis of the efficiency of the alpha-beta algorithm is carried out based on a probabilistic model in which terminal node scores depend on random branch values. Explicit formulas are derived for the expected number of terminal nodes scored for the cases of uniform trees of depths 2 and 3. For trees of depth 2, the expected number is of order $O(NH_N)$; for trees of depth 3, the expected number is of order $O(N^2)$. An upper bound on the expected number of terminal nodes scored for trees of depth 4 is shown to be no greater than $O(N^2H_N^2)$ and no less than $O(N^2)$.

MARKOV ALGORITHMS AND THEIR RELATIONS WITH ASSOCIATIVE PROCESSING SYSTEM

by

Martine PAGET

Institut de Programmation
Université Paris 6, 4 Place Jussieu
75230 PARIS CEDEX 05

The concept of a Normal Markov Algorithm (NMA) is very important in the theory of computability. It has been used in different domain such as recursivity, semantics and models of programming languages ; we are here interested with time and space complexity of such algorithms. A NMA is of time complexity T(n), (n equal to the length of the input word) if the number of productions which have been executed is T(n) ; we prove the following relations between Türing Machine, Random Access Machine (RAM) and NMA : If a RAM [resp. Türing Machine with 1 tape] is of time complexity T and of space complexity L then there is a NMA which simulates the RAM [resp. Türing Machine] in time complexity $O(T^4)$ [resp. O (T)] and in space complexity $O(L^2)$ [resp. O(L)]; if a NMA is of time complexity T and of space complexity L then there is a RAM [resp. Türing Machine] which simulates the NMA in time complexity $O(T^2 \text{ Log } T)$ [resp. $O(T^3)$] and in space complexity O(L) [resp. O(L)]. Then we introduce an associative abstract machine (AAM). We prove that if an AAM is of time complexity T and of space complexity L then there is a NMA which simulates the AAM in time complexity $O(T^2)$ and in space complexity O(L). If a NMA is of time complexity T and of space complexity L then there is a AAM which simulates the NMA in time complexity O(T) and in space complexity O(L).

We conclude that the utilisation of associative processor will fairly improve the execution of Markov Algorithm and programs written in languages constructed on the scheme of NMA as SNOBOL.

TESTING MIXED GRAPHS FOR UNICURSALITY

Christos H. Papadimitriou
Princeton University

The minimum Eulerian graph (MEG) problem is the following: given a mixed graph (i.e. a set of nodes V, directed edges A and undirected edges E), find the minimal set of directed edges that need to be added in order to make the resulting graph Eulerian. We present an $O(|A| + |E|^{1.5} + |E| \ |V|^{.67})$ algorithm for this problem. The algorithm can also be viewed as a unicursality test for mixed graphs; as such it is an improvement over the best known (exponential) test. Since the max-flow problem with unit edge capacities can be reduced in linear time to the MEG problem, it follows that any improvement on the above bound would result in a similar improvement on the time requirements of this max-flow problem.

Some NP-Hard Problems Involving Sparse Polynomials

David A. Plaisted

University of Illinois

Abstract

In this paper we show that the following problems are NP-hard: (Assume all polynomials are sparse polynomials with integer coefficients.)

1. Determining if the degree of the least common multiple of a set of polynomials is a given integer.
2. The same as 1, but for greatest common divisor.
3. Determining if a polynomial is a factor of a product of polynomials.
4. Determining if the number of (distinct) complex zeroes of a product of polynomials, is a given integer.
5. Determining if $p(z)/q(z)$ is analytic where p is a product of finitely many polynomials and q is a polynomial.
6. Determining if an exponential expression of integers is a factor of another such expression. An exponential expression of integers is an expression formed from integers and the operations of addition, subtraction, multiplication, and exponentiation.

Other results are also presented.

ON THE DETERMINATION OF A DENSEST HEMISPHERE

Franco P. Preparata
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Abstract

Let E^d be the d-dimensional Euclidean space and, let S^d be the sphere of unit radius with center in the origin of E^d. Let K be a set of n points on S^d. Find a hemisphere of S^d which contains a largest subset of K. This problem originated from applications in statistical analysis. It is shown that a solution of this problem can be algorithmically obtained with a number of operations (comparisons of numbers) $O(n^{d-1}\log n)$, so that the complexity is polynomially bounded for fixed number of dimensions.

The approach consists in reducing the problem on n points in E^d to a collection of $\binom{n}{d-2}$ subproblems, each in $(n - d + 2)$ points in E^2, by projecting the hypersphere onto a hyperplane tangent to the sphere. The result follows from the fact that a very simple algorithm for s points in E^2, can be constructed, which runs in time $O(s \log s)$.

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

REDUCTION OF DEPTH OF BOOLEAN NETWORKS WITH A FAN-IN CONSTRAINT

F. P. Preparata,* D. E. Muller,* and A. B. Barak**

Abstract

We present a family of techniques for the design of combinational networks. The objective of these techniques is the reduction of the number of levels, subject to a constraint on the fan-in of the logic gates. We show that a boolean expression with n literals and involving the connectives AND and OR can be restructured so that the resulting network has depth at most $C_\ell \log_2 n + \delta$, where $\delta < 0.415$ and C_ℓ is 1.81, 1.38, 1.17, and 1 for maximum fan-in ℓ of 2, 3, 4, and 5, respectively. If we additionally require that the amount of equipment of the resulting network be bounded by a linear function of n, it is possible to bound the depth by $2 \log_2 n$ with a fan-in of at most 3.

The design techniques adopted are germane to those used for the restructuring of arithmetic expressions for parallel evaluation. The additional speed-ups (depth reductions) are achieved by taking advantage of the dual distributivity of boolean algebra and of fan-ins not restricted to the value of 2.

---

*Coordinated Science Laboratory, University of Illinois at Urbana, 61801.
**Department of Computer Science, the Hebrew University, Jerusalem, Israel.

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

On a relationship between programs of address machines , Mazurkiewicz algorithms and FS-expressions of a formalized algorithmic language.

by

Zbigniew Raś
University of Warsaw
Institute of Mathematics
Warsaw, PKiN, Poland

ABSTRACT. The properties of programs we can investigate from the point of view of a computer in which they are realized or we can introduce some abstract model of a program in which we can describe all interesting for us attributes of programs. It seems that without a computer we can not precisely investigate the nature of programs, properties of programs. On the other hand an introducing of a computer restricts a liberty in an acceptance of a model of a program because in this case we speak about a program in a computer. So it is natural an investigation some relationships between this approaches. As a model of a computer we accepte an address machine due to Pawlak and on the other hand the notion of Mazurkiewicz algorithms in abstract quasinets and the notion of FS-expressions of a formalized algorithmic language due to Salwicki as models of programs. There is introduced a category of programs of an address machine, a category of Mazurkiewicz algorithms in an abstract quasinet and a category of FS-expressions of a formalized algorithmic language. Morphisms are some homomorphisms. It appears that the category of programs of an address machine is equivalent with some category of Mazurkiewicz algorithms and some category of FS-expressions of a formalized algorithmic language. There are transfered some properties of programs of address machines on the case of Mazurkiewicz algorithms on the case of FS-expressions and inversely. On the other hand it is worth to observe that the control in address machines is by means of the instruction counter, in Mazurkiewicz algorithms by means of labels and in FS-expressions by means of formulas. Thus the theorem about the equivalence between this three categories show that the control by means of the instruction counter, by means of labels and by means of formulas are equivalent .

Micro-Analysis as a Technique for Problems on a Graph

Winston Riley III

Riley Systems Corporation

In addressing graph problems for which efficient algorithms are not known, the fundamental manner by which the researcher views the problem appears to restrict if only implicitely his research design. A proposed approach, seldom considered, involves the examination in lowest-level detail of few-element cases for patterns of inter-relationships among the elements of the problem which may be extrapolated to the general case.

The micro-analytical approach can provide data rich enough for observing the full complexity of the basic relationships from which algorithmic discoveries may be made and proof analyses developed. Examples will be presented, along with interesting problems posed in using computers in the micro-analysis process.

The Structural Complexity of Discrete Optimization Problems

Arnie Rosenthal
The University of Michigan
Department of Computer and Communication Sciences

Abstract

We investigate the complexity of decomposition procedures for unconstrained discrete optimization problems, and prove the "optimality" of dynamic programming. If most pairs of variables interact only linearly, and the pattern of nonlinear interactions is particularly simple, then dynamic programming is known to be particularly efficient. Since the general optimization problem is NP-hard, we can only hope our exact algorithms will be efficient in special situations.

We know of only one admissable operation: Compare two partial solutions that assign values to the same set of variables, and which both assign the same value to any variable which interacts with an unassigned variable. Bellman's principle of optimality may then be applied to eliminate all extensions of the partial solution which produced the inferior value of the objective function. Since many algorithms perform the elimination implicitly, no cost is assigned to this step.

Bellman's principle seems to capture one notion of decomposition. A more strict notion also forbids comparisons on sets of variables which "overlap". Nonserial dynamic programming algorithms are in this subclass. Our major theorem proves that there is a particular (hard to find) dynamic programming algorithm which is optimal in a very strong sense: Any set of nonoverlapping comparisons that proves a particular solution to be optimal (i.e. nondeterministically finds an optimum) has at least as many comparisons as that dynamic program. Since the dynamic program is nonadaptive, the number of comparisons it uses depends only on the interaction structure of the objective function, and not on the values. Thus the number of comparisons can be used to obtain a reasonable measure of the complexity of the interactions. This measure can also be expressed in terms of the graph of the relation "nonlinearly interacts".

We conjecture that if the restriction that comparisons do not overlap is removed, then dynamic programming will be worst-case optimal among deterministic comparison-based decomposition algorithms.

Keywords: Complexity, dynamic programming, decomposition, lower bound, interaction graph, nonserial dynamic programming.

A CONVERGENT RANDOM WALK

by

Sam Savage
University of Chicago

Abstract

Many optimization algorithms generate a trajectory in a multidimensional space. Often this trajectory may be viewed as the result of a random walk in which steps which reduce the value of the objective function are forbidden. This paper discusses such a random walk optimization as applied to a linear objective function and shows that the convergence rate deteriorates roughly as $\frac{1}{\sqrt{d}}$, where d is the dimension of the space in which the walk takes place.

Complexity Analysis of Some Problems in Pipeline Processor Design

by

M. S. Schlansker

The University of Michigan
Ann Arbor, Michigan

This paper discusses some ongoing work at the University of Michigan in the area of pipeline processor design and scheduling. A model of pipeline computation is presented, and compared to a pre-existing model. A design optimization algorithm implementated on this pre-existing model is reviewed and, extensions to this algorithm necessary to allow it to operate within the new modelling framework are presented. It is shown that the design space represented by the new model is larger than the original restricted space.

The complexity of both algorithms is examined and, both are found to require exponential time. Further, the design optimization problem on the extended design space is shown to be NP complete.

ON THE COMPLEXITY OF RESOLUTION

by

Robert E. Shostak
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

A propositional decision procedure quite different from resolution is introduced and is seen to give rise to a nested hierarchy of infinite classes of clause sets. The class at the kth degree of the hierarchy is decided by the procedure in polynomial (in the number of clauses) time of degree k. The procedure thus operates with greater efficiency on sets appearing at lower levels than at higher levels. Resolution, on the other hand, tends to perform in the opposite manner. A theorem is proved giving a simple syntactic characterization of a large subclass of sets at the bottom level for which there exist no simple resolution refutation trees. It is then shown that there are sets at the bottom level having unrestricted resolution refutations simpler than the simplest resolution trees for those sets. This result has the consequence that unrestricted resolution sometimes admits proofs simpler than the simplest admitted by the best-known linear strategies (e.g. SL-resolution, t-linear, msl.).

COMPLEX ANALYSIS METHODS IN CONTROL THEORY: A BOUNDARY CONTROL PROBLEM FOR THE EQUATIONS IN AREOLAR DERIVATIVES

Claudia Simionescu
Department of Mathematics
University of Braşov, Braşov, Romania

ABSTRACT

For an equation of the form

$$P(D)y = f + Bv \qquad v \in U_{ad} \subset U$$

the optimal control problem with a quadratic cost functional $J(v) = \|Cy - z_d\|_H^2 + (Nv,v)_U$ has, under some regularity conditions, a unique solution $u \in U_{ad} \subset U$, which is characterized by the optimality system ([2])

$$\begin{cases} P(D)y = f + Bu \\ u \in U_{ad} \\ (I_U^{-1} B^x E(C^x I_H(C(Ex(f+Bu)) - z_d)) + Nu, v-u)_H \geq 0 \end{cases} \tag{1}$$

where E is the elementary solution for P(D).

This result is applied to the equation in areolar derivative

$$\frac{Dy}{D\omega} = f + Bu \qquad u \in U_{ad} \subset U \tag{2}$$

where $\frac{D}{D\omega} = \frac{i}{2}(D_1 + iD_2)$ is defined in the distributional sense.

Since the elementary solution for (2) is ([1])

$$E = \frac{1}{\pi z} \in B_{\infty}^{loc}, \frac{\tilde{D}}{D\omega}(R^2)$$

the integral representation for the solution of (2) is given by

$$y(u) = y_1(u) + (\frac{1}{\pi z} x(f+Bu)) \tag{3}$$

Using this fact and the corresponding optimality system (1), we obtain an integral representation for the solution of a generalized boundary control problem of Riemann-Hilbert type for the equation (2), its iterates, and for a more general equation

$$\frac{Dy}{D\omega} + Ay + By = f + Bu \tag{4}$$

also. We consider and the case of a distributed control.

We give a stable and convergent approximation for the solution and discuss the result.

The problem has a large field of applications: many aspects in applied physics leading to equations of the form (2) or (4).

A COMPLEXITY THEORETIC CHARACTERIZATION OF NON-RECURSIVE DIFFERENTIAL IDEALS

Martin K. Solomon
Rutgers University

The existence of finitely generated non-recursive differential ideals is currently an open question.

We show that such a differential ideal I exists iff the corresponding perfect ideal P is a measure independent Gödel speed-up of I. In particular, I is not recursive iff every computer program with domain P is a Gödel speed-up of every program with domain I.

We also show that I is not recursive iff for every recursive number theoretic function, r, there exists an x in I and a natural number n such that $r(L(x^n)+n) < L(x)$, where $L(y)$ = length of the "shortest representation" of y in terms of the generators of I.

TITLE: On the paging-complexity of loop-like structures

AUTHOR: Hans-Georg Stork

ADDRESS: Fachbereich Informatik der TH Darmstadt, Forschungsgruppe Automatentheorie und formale Sprachen, Magdalenenstr. 11, 6100 Darmstadt, West Germany

ABSTRACT: The following "paging-pagination problem" is treated: "Given a paging-algorithm and a particular address-reference-structure (generated by some program); how should code and data be written on pages in order to obtain page-reference strings that bring about the least page-traffic between main- and secondary memory of a two-level computer system?"

Here, paging-algorithms are modelled by finite automata with an additional cost structure; address-reference structures are total recursive functions from the set of natural numbers into the free monoid over a set of addresses. A "paging-complexity" (loosely speaking, the minimal page-fault rate) is introduced in order to measure optimal performance obtainable for a particular reference structure.

We investigate this problem in a very simple setting: paging-algorithms are restricted to demand-paging-algorithms and the reference-structures under consideration are intended to model various ways of looping over a linear address space (a situation though that occurs quite frequently in connection with processing large arrays):

We prove that the obvious way of paginating - writing adjacent addresses on the same page if possible - is optimal for any paging-automaton that realizes Belady's optimal algorithm for the given reference structure (a "principle of locality"). The proof rests on a lower bound argument which is due to the fact that Belady's algorithm is a "stack algorithm".

There are two natural subproblems of the paging-pagination problem which may be labeled "synthesis" and "analysis". The latter concerns the search for optimal paginations with respect to a given paging-automaton (this problem is closely related to the analysis of finite acceptors). Treating the case of an LRU-automaton more explicitly, we show how the "principle of locality" exhibited before is invalid for LRU-paging of loops.

Finally, we prove that the paging-complexity of a concatenation of simple loops is just the arithmetic mean of the paging-complexities of the simple loops involved.

Minimum Energy Transfer in the Presence of Large Controller Gain Variations

Fred Taylor
University of Cincinnati

ABSTRACT

The problem of minimal energy state transfer to the origin in finite time is considered. The problem is approached through linear programming. This method of mathematical programming requires that the discrete problem of finite dimension be formulated. An efficient method, by which a continuous problem can be formatted into an acceptable discrete form, is presented. The resulting mathematical programming problem, once solved, is supported with a new post optimal analysis. This theory, now complete, can be used to test the sensitivity of an optimal solution to possible variation in controller gains. The theory by which the mathematical structure of an optimal control policy is treated in general as well as a special scalar case. The general problem requires an auxillary linear programming problem be solved while the scaler problem degenerates into a simple algebraic test.

Solutions of Very Large Travelling Salesman Problems

by

Gerald L. Thompson
Carnegie-Mellon University

This paper gives a survey of computational results obtained by T. H. C. Smith, V. Srinivasan and the author for the solution of large travelling salesman problems. The three classes of such problems solved are as follows:

(a) Symmetric Sum Problems. Solutions using an implicit enumeration LIFO search, subgradient method, for problems having up to 100 cities.

(b) Asymmetric Sum Problems. Solutions using LIFO search, operator parametric programming variation of the assignment problem solution, for problems having up to 200 cities.

(c) Symmetric and Asymmetric Bottleneck Problems. Solutions using LIFO search, operator parametric programming variation of the bottleneck assignment problem solution, for problems having up to 2,000 cities.

On the Computation of Functions by a Parallel Computer

G. Tourlakis
Dept. of Computer Science
Atkinson College
York University
Downsview, Ontario, Canada

Abstract. In this paper we introduce a model of parallel computer, the GLPM, which is an infinite binary tree each node of which is an LPM. Briefly, an LPM is an infinite array of processors arranged in a "tree-like" manner. The configuration is such that the Floyd-Meyer algorithm for detecting existence of a dipath in a digraph can be implemented on the LPM.

Given now any total function f(x) which can be computed by a deterministic serial computer T in space $L(|x|)$ (where $|x|$ = length of x) and an input x, the start-processor of the GLPM will:

1) Activate all the LPM's on its first $0(2^{L(|x|)})$ nodes (note that $f(x) \leq 0\ (2^{L(|x|)})$)
2) The i^{th} activated LPM ($i=0,1,\ldots,0(2^{L(|x|)})$)
   2.1) constructs a digraph, whose nodes are all the $L(|x|)$ - bounded T-ID's, and edges (a,b) exist iff ID a yields ID b.
   2.2) solves the problem of whether the initial ID reaches the final ID, with result i, through a dipath (the Floyd-Meyer algorithm is used).
3) One and only one LPM, say the y^{th}, will detect connectivity. The number y is made known at the start-processor of the GLPM and is the value of f(x).

The above can be done in $0(L(|x|)^2)$ steps, a similar time bound as that needed by the vector-machines of Pratt et al. for non-deterministic L(n)-space bounded recognition problems.

It is also shown that an LPM (and, a fortiori, the GLPM) is as fast as the vector-machines for non-deterministic recognition problems.

Reference:

V. R. Pratt, M. Rabin, L. J. Stckmeyer, "A characterization of the power of vector machines", Proc. 6th Annual ACM Symposium on theory of computing, April 30 - May 2, 1974, pp. 122-134

A Facial Recognition Algorithm

James R. Townes
Assistant Professor
Department of Electrical Engineering
University of Houston

After observing the commitment of a crime, a witness is frequently asked by a law enforcement agency to look through a collection or mugfile of photographs of previous offenders in hopes of finding or identifying a criminal suspect. Due to the alarming high rate of repeat offenses, the probability of the appearance of a suspect's photograph in such a collecting is relatively high.

However, practical considerations concerning the manual sorting through mugfile photographs by a witness tend to severely reduce the chances of identifying the desired photograph. First, even for a moderate size city, the number of images in a mugfile is enormous (in the thousands); a fact which causes fatigue and mental confusion in any witness. Secondly, due to the increase in the number of drug-related crimes, it has become increasingly difficult (if not impossible) to specifically categorize criminals into distinct groups (burglary, grand theft, etc.). This results in a general overlapping of many types of criminal activity.

The study reported in this paper addresses the problem of selecting, via a mini-computer, the mugfile photograph most closely resembling the description from a witness. In synoptic form, the algorithm has the following structure: (1) a witness/sketch-artist team generates a drawing or other facial facsimile of the suspect, (2) a set of nine geometric measurements are recorded from the sketch, (3) along with optional qualitative assessments, the algorithm computes a "distance" between the sketch and each photograph in the mugfile, and (4) the photographs are ranked in order increasing distances.

Of course, no panacea claims are made. However the algorithm is shown to be efficient both in reducing a large mugfile down to a small number of "look-alike" images and in operating in a real-time environment on a moderate size mini-computer. The form of the distance algorithm, its training, its overall achievements, as well as specific examples of its usage are presented in the paper.

A FAST ALGORITHM FOR SIGNIFICANT DIGIT COMPUTATION OF THE INCOMPLETE BETA FUNCTION FOR EXTREME VALUES

MARIETTA J. TRETTER
PENN. STATE UNIVERSITY

G.W. WALSTER
UNIVERSITY OF WISCONSIN

Several algorithms are available for computing the incomplete beta function: $\frac{x^a(1-x)^b}{aB(a,b)}\,{}_2F_1(1,a+b,a+1;x)$, (1)
(e.g. Amos, Sandia Labs rept. no. SC-DR-69-591). Most algorithms include a combination of series representations and recursion relations. However, none of these methods are very useful when parameter a is much larger than b and the argument is close to one(which occasionally occurs in statistical computations). For example, if $a = 1 \times 10^{20}$, $b = 5000$, and $x = .9999999999999999$, these algorithms would require thousands of terms for a single significant digit, when the algorithm is numerically stable.

An obvious solution to the convergence problem would be to accelerate the series representations. W. Gosper graciously used his powerful splitting functions in an attempt to accelerate (1) above with no result. The corresponding continued fractions for the above series and, $\frac{x^a(1-x)^{b-1}}{aB(a,b)}\,{}_2F_1\left(1,1-b,a+1,\frac{-x}{1-x}\right)$, (2)
Aroian (Ann. Math Statis., v12, 218-223), were then investigated as a means of accelerationg convergence. However, subtraction destroyed all digits before any semblance of convergence was obtained and there was no way to eliminate the subtraction analytically. The continued fractions were then transformed into associated continued fractions (Perron, DIE LEHRE VON KETTENBRUCHEN, 322-330). Aroian(erroneously) reported this faster converging continued fraction for (1) but never tried it as he considered it too complicated for hand calculation. The continued fraction (associated) for (2) has not been reported before.

An immediate advantage to the associated representations is the facility of using analytic subtraction yielding a continued fraction that is easily bounded by its even and odd convergents. The most astounding advantage, though, is the convergence rate; both associated continued fractions produced 25 significant digits with 21 convergents for the above example.

Paul G. Tuttle
BABCOCK & WILCOX, Nuclear Power Generation Division
P. O. Box 1260
Lynchburg, Virginia 24505

VECTOR PROCESSING OF THE FAST FOURIER TRANSFORM

A study of the implementation of the Fast Fourier Transform on a vector computer is made. A set of characteristics for a vector computer is assumed. Among these characteristics is that the operation count can be expressed as

$$P + n \cdot R$$

where P and R are, respectively, the vector prime time and the vector rate time (P >> R) and n is the vector length. It is shown that the direct implementation of either the Cooley-Tukey or Cooley-Sande algorithm results in an operation count on the order of

$$n \cdot P + n \cdot m \cdot R$$

where $n = 2^m$. A variant of the Cooley-Sande algorithm is developed. This variant yields an operation count on the order of

$$m \cdot P + n \cdot m \cdot R.$$

The variant of the Cooley-Sande algorithm was implemented on a CDC-7600 using the Lawrence Livermore vector subroutines. The efficiency of the algorithm is judged with respect to the achievable vector execution rate.

BINOMIAL POLYTOPES AND NP-COMPLETE PROBLEMS

Silvio Ursic

Department of Applied Mathematics

University of Sao Paulo, Brasil

The use of the relationships $x_i^2 = x_i$, where x_i are the zero-one variables of a linear integer programming problem, permit the derivation of a family of equivalent problems, whose constraints are systems of integer polynomial equalities and inequalities. A family of polytopes, Binomial Polytopes, is defined and shown to contain some of the structure present in the equivalent integer programs. It is also shown that Binomial Polytopes are strongly related to Boolean Logic Conjunctive Normal Forms. Some of the properties of Binomial Polytopes lead to new approaches for the solution of discrete optimization problems.

SOME RESULTS ON THE INFERENCE OF STOCHASTIC GRAMMARS.

A. Van der Mude and Adrian Walker

Department of Computer Science

Rutgers University

A stochastic regular grammar (SRG) can be thought of as a notation for several of the structures used in computer science, (e.g. a directed weighted labelled graph, a set of situation-action rules), hence the automatic construction of a best SRG from empirical data is of interest. It is known that this can be done in principle [1], but the computation time of the best known algorithms is too large for practical applications [2].

With the aim of finding an economical algorithm, we give some results [3] about the best SRG's for a data set. We treat SRG's with rational weights, and we use the sum of the denominators of the weights as a complexity measure. We show, for instance, that ambiguity is necessary to model certain data sets. We also show how, during a search, the complexity of the remaining SRG's of interest may be progressively bounded in terms of the SRG's already found.

1. J. J. Horning. A Study of Grammatical Inference, Report No. CS 139, Computer Science Department, Stanford University, 1969.

2. K. S. Fu and T. L. Booth. Grammatical Inference: Introduction and Survey - Part II, IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-5, No.4, July 1975, 409-423.

3. A. Van der Mude and Adrian Walker. A Program Which Constructs a Stochastic Regular Grammar to Fit a Data Set, Report CBM-TR-59, Department of Computer Science, Rutgers University February, 1976.

ON FINDING LOWEST COMMON ANCESTORS IN LESS THAN LOGARITHMIC AVERAGE TIME

Jan van Leeuwen
Department of Computer Science
State University of New York at Buffalo
4226 Ridge Lea Road
Amherst, New York 14226

Constructions of Aho, Hopcroft, and Ullman [1], show that n LINK/LCA instructions on an n-node forest can be processed on-line in only $O(n\log n)$ steps. In this paper we shall consider similar set manipulation programs in which both arguments of each LINK instruction happen to be roots in the forest at any time such an instruction applies. Such programs occur in relation to ordinary UNION/FIND programs in which one must keep track of the smallest set to which two elements simultaneously belong. We show that n UNION/LCA instructions can be processed on-line in only $O(n\log\log n)$ steps. A related reference is van Emde Boas [2].

[1] Aho, A.V., J.E. Hopcroft, and J.D. Ullman, On finding lowest common ancestors in trees, Proc. 5th Annual ACM Symp. on Theory of Computing (1973) 253-265.

[2] van Emde Boas, P., Preserving order in a forest in less than logarithmic time, Proc. 16th Annual Symp. on Found. of Computer Science (1975) 75-84.

Processor Utilization in a Parallel System

Frances L. Van Scoy

Old Dominion University

Norfolk, Virginia

A model of parallel computation based on a modification of von Neumann's cellular space (Smith, 1971) is presented. The system is synchronous and consists of a rectangular array of processors. Each interior processor can access the memories of its four adjacent neighbors.

Various algorithms designed for this system are mentioned. Among these are algorithms requiring n^2 processors for performing matrix multiplication, transposition, and inversion and for computing the transitive closure of a Boolean matrix. These algorithms require time $O(n)$, $O(n)$, $O(n^2)$, and $O(n)$, respectively.

While these algorithms offer an improvement in time over the known sequential algorithms, they do so at a cost of requiring n^2 processors instead of 1. Using the terms computation, bookkeeping, and bottleneck from a paper by Miranker (1971), a description of mean processor utilization for each of these algorithms is given. It is shown that a typical processor generally spends less than 1/3 of its time performing computations and over 1/2 of its time in a bottleneck status awaiting data or a signal to resume processing.

References:

Miranker, W., 1971. "A Survey of Parallelism in Numerical Analysis". SLAM Review. 13: 524-547.

Smith, A., 1971. "Two-Dimensional Formal Languages and Pattern Recognition by Cellular Automata". 12th IEEE Symposium on Switching and Automata Theory. 12: 144-150.

ALGORITHM WITH NONLINEAR ACCELERATION FOR A SYSTEM OF LINEAR EQUATIONS

Boris S. Verkhovsky
Department of Civil Engineering, Princeton University

ABSTRACT

Many mathematical and physical problems can be reduced to the task of solving a system of linear equations with a large number of unknowns: the solution of partial differential equations using numerical methods is an example.

The purpose of this paper is to introduce a nonlinear algorithm containing an accelerator term which, subject to certain conditions, is a powerful approach to solving such systems of linear equations.

On the average path length between two nodes in a randomly generated tree.

Da-Lun Wang
Department of Mathematics
University of Kentucky
Lexington, KY 40506

Abstract:

In a computer experiment, it is found that the average path length between two nodes of a randomly generated tree of 1000 nodes is around 11. In this paper, we set up a recursive equation among the expected path length L_n between two nodes in a randomly generated tree of n nodes. It is found that $L_{n+1} = \frac{2}{n+1} + \frac{n^2 + n - 2}{n^2 + n} L_n$ if paths of length 1 included, and $L_{n+1} = \frac{4n+4}{n^2} + \frac{n^2-4}{n^2} L_n$ if paths of length 1 are not included. We therefore obtain $L_n \approx 2 \log_e n - c$ where c is a constant near 3. This result agrees with the experiments very well.

An Efficient Algorithm for Finding a Minimal Equivalent Subgraph of a Digraph

by

Da-Lun Wang
Department of Mathematics
University of Kentucky
Lexington, Kentucky 40506

Chung C. Wang
Department of Computer Science
University of Kentucky
Lexington, Kentucky 40506

Ping Y. Wang
Department of Mathematical Sciences
University of Cincinnati
Cincinnati, Ohio 45221

ABSTRACT

Let G be a digraph. A subgraph of G is equivalent to G if it has the same transitive closure as G. We define that an equivalent subgraph of G is minimum if it has the least number of edges and minimal if it has no redundant edges. Sahni has shown that the problem of finding a minimum equivalent subgraph of G is NP-complete. The problem of finding a minimal equivalent subgraph of G is clearly not NP-complete. In this paper, we present a best known efficient algorithm for finding a minimal equivalent subgraph of G.

ABSTRACT

ON PROBLEM COMPLEXITY

M. J. Waxman, Ph.D.
TRW Inc.

Development of better methods of problem solving requires determining what makes a specific problem complex and how to simplify it. This paper proposes a theoretical basis for studying the question of problem complexity in general, not algorithmic complexity. It starts with the identification of the basic concept of problem complexity as being related qualitatively to the unknown elements of the problem, defines measures for problem complexity and then discusses resource application for reduction and elimination of complexity in order to solve problems. A definition is proposed for complexity of problems which is then related to a number of specific types of problems Then possible measures for problem complexity based on information theoretic approaches are proposed. Examples of sources of complexity are presented including complexity due to problem statement, inherent size of a problem, interrelationships among elements, etc., and how these appear in specific problems in management, computing, etc., are discussed together with areas needing further exploration.

Context Free Language Recognition by a RAM with Uniform Lost Criterion in Time $n^2 \log n$ *

Reinhold Weicker**

The Pennsylvania State University
Computer Science Department
University Park, PA. 16802

It is shown that if a Random Access Machine (RAM) with uniform cost criterion is used as the underlying model, context free language recognition can be performed in time less than n^3 or $n^{2.81}$ even in the case of an ambiguous language. This is done by a modification of Earley's algorithm. This algorithm can be regarded as a set-manipulating algorithm which generates elements of a recognition matrix. The new algorithm avoids generating a matrix element more than once, for the sets which can contribute to the generation of a new element are united before they are used. This requires a specific sequence of "Union" (of non-disjoint sets) and "Find" steps to be performed. The use of a suitable representation for sets makes it possible to execute this sequence, by a RAM with additional boolean operations $(\cup,\cap)$, in time $O(n^2)$, and in time $O(n^2 \log n)$ by a RAM with arithmetic operations $(+,-)$ only, provided that a uniform cost criterion is used.

* This work was supported by the German Academic Exchange Service (DAAD), grant no. 430/402/562/6.

** On leave from Universität Hamburg (Germany), Institut für Informatik.

FAST PRIME TESTING

P.J. Weinberger
University of Michigan

Last year Gary Miller showed that if certain Riemann hypotheses are true, then any natural number N can be tested for primality in polynomial time ($(\log N)^A$) . The constants involved in his algorithm can be determined. My determination of them gives a prime testing algorithm which seems to be competitive with the best one known, for large N (around 10^{60}) . Further, when the other algorithms are failing to terminate quickly, the information they gather can be used to reduce significantly the running time of Miller's algorithm.

THE COMPUTATIONAL COMPLEXITY OF TAYLOR SERIES METHODS FOR INITIAL VALUE PROBLEMS

Arthur G. Werschulz
Department of Mathematics
Carnegie-Mellon University
Pittsburgh, PA 15213

ABSTRACT

We consider the problem of solving an autonomous ordinary differential equation initial-value problem in N dimensions, using Taylor series methods. Using fast power series techniques due to R. P. Brent and H. T. Kung, we show that the Taylor series method of order p may be done with combinatory cost per step of $O(p^N \log p)$ as $p \to \infty$; the linear lower bound for this problem is $O(p^N)$. These results are then used to find complexity bounds for the problem of solving an initial-value problem to within a given error criterion, using Taylor series methods.

Recursive Matching Algorithms and Linear Orders on the Subset Lattice

D. E. White and S. G. Williamson

School of Mathematics
University of Minnesota
Minneapolis, Minnesota 55455

Abstract. Let $C(m,p)$ denote all subsets of size p contained in $C(m)$, the set of all subsets of $\{1,2,\ldots,m\}$. We study the natural recursive matching algorithms based on the standard recursions $|C(m)| = 2|C(m-1)|$ and $|C(m,p)| = \sum_{k=p}^{m} |C(k-1,p-1)|$. The resulting matching functions are the same in both cases and are characterized explicitly. We show that for an extensive class of linear orders on $C(m)$, the matching function defined recursively by mapping in order the elements of $C(m,p)$ to the first available element of $C(m,p+1)$ is the same as the above matching. Included in this class of orders are colexicographic and lexicographic orders.

An Analysis of a Geographic Data Base Problem

by

Lee J. White
Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

George Nagy has suggested the following problem. Given N points in the plane, and assume there are four pointers $P_{nw}(x)$, $P_{ne}(x)$, $P_{sw}(x)$, and $P_{se}(x)$ associated with each such point x. Consider the four quadrants formed by choosing each point x as the origin. The nearest point to x in each quadrant is indicated by each of these pointers, where a pointer is set to null if no such point exists. The problem is to devise an algorithm so that points may be inserted or deleted without examining all N points and updating all the pointers. An algorithm has been devised for this purpose, and a complexity analysis is provided.

The analysis of this algorithm involves the following question. For N points, set the $P_{sw}(x)$ pointer to indicate the nearest point in the southwest quadrant. This structure may be considered as a number of rooted trees, the root of each tree being a point with no southwest neighbor. Given N points randomly distributed in the plane, the problem is to obtain the expected number of trees and terminal nodes of the trees, as these parameters determine the complexity of the algorithmic solution of the geographic data base problem. The expected values of these parameters are given for several models of randomly distributed points.

ON THE COMPLEXITY OF PARAMETER TRANSMISSION QUESTIONS IN HIGHER PROGRAMMING LANGUAGES (ABSTRACT)*

Karl Winklmann
Department of Computer Sciences, Purdue University
West Lafayette, Indiana 47907

A simple recursion-theoretic argument shows that it is in general undecidable whether or not execution of a given procedure call in a higher level language program involves execution of another call of the same procedure (recursivity of procedures). This decision problem can be simplified in a natural way by assuming that whenever a procedure is called, all the call statements within the procedure body get executed in parallel. Under this assumption, recursivity of procedures (formal recursivity) is trivial to decide if the programming language in question does not allow procedure identifiers to be used as parameters for procedures. For languages that do allow procedure parameters, Langmaack [1] has shown that formal recursivity of procedures is decidable if and only if the language does not allow constructions like

proc p(x); begin proc q; begin .. x .. end q; ... end p

where the parameter x of the outer procedure p is used in the declaration of the inner procedure q (global formal parameter).

We show that even for languages that do not allow global formal parameters, deciding formal recursivity of procedures must be expected to be infeasible in general: the decision problem is roughly equally difficult as the halting problem for non-deterministic space-bounded machines, a problem which, even though it involves two major open questions of complexity theory, is widely conjectured to be prohibitively difficult to solve.

We prove analogous results for reachability of procedures and for correctness of parameter transmissions in programs.

Our proofs use standard complexity-theoretic techniques.

Reference:

1. Langmaack, H.: On Correct Procedure Parameter Transmission In Higher Programming Languages. Acta Informatica 2, 110-142 (1973).

* Partially supported by NSF Grant GJ27127A1.

Submitted by: Chee-Keng Yap

Affiliation: Yale University

Abstract

New Results for Medians and Related Problems

The previous best lower bound for the Median Problem is 7N/4 - log N due to Pratt and Yao. Since then, no substantial progress had been made, although both Kirkpatrick and Schonhage independently made the slightly improved bound of 7N/4 in simplified proofs. In this paper, we obtain a new lower bound of 11N/6. The proof employs a rather sophisticated Adversary. A number of new techniques are introduced that may find application elsewhere. The most significant gain came from a kind of "Church-Rosser Property" of our Adversaries.

Combining our result with one due to Paterson, Pippenger and Schonhage this is the first general example showing that the average cost of "mass producing" a particular structure ("spiders") is cheaper than individual production. Such a phenomenon (what Schnorr calls "hidden dependencies") is by no means an isolated one, and which we think is important in our understanding of intrinsic complexity. We also give the first non-trivial lower bound of 11N/8 of the average cost of "mass producing" spider-structures. This leaves a gap of 3N/8 between known bounds.

Based on results for the Median Problem, the lower bound for the w-Fractile Problem due to Pratt and Yao are improved. Finally, we give a lower bound for a very general class of problems, called the General Partition Problem, $P(i_1,\ldots,i_k)$. This class includes most of the important comparison-based problems in the literature as special cases (E.g. Sorting, i-th Selection, i-th Ordering and the Bipartition Problem).

PROVING "PRODUCER/CONSUMER" CORRECTNESS BY PATH EXPRESSIONS AND SCHEMAS

Lawrence Yelowitz
Computer Science Department

University of Pittsburgh
Pittsburgh, PA 15260

The following must be shown for a system [1] consisting of a single producer and a single consumer, communicating via an infinite buffer, and synchronized via binary semaphores, to establish correctness:

1. No buffer underflow occurs;
2. No simultaneous "insert" and "retrieve" occur at any buffer position;
3. The producer goes through its cycle of "produce" and "insert" in strictly sequential order (that is, there are never two occurrences of "produce" without an intervening "insert", and vise versa);
4. The consumer goes through its cycle of "retrieve" and "consume" in strictly sequential order;
5. All semaphore operations are "well-behaved"; that is:
   5.1 No deadlock occurs;
   5.2 There is no semaphore sem for which "V(sem)" is executed twice without an intervening "P(sem)".

In terms of path expressions [2], conditions (3), (4), and (5.2) may be expressed as:

path produce; insert end, path retrieve; consume end,

(∀sem) (path P(sem); V(sem) end OR path V(sem); P(sem) end)

where the last condition depends on sem being initialized to one or to zero.

Conditions (1), (2), and (3) are trivial to show. Condition (5.1) can be shown by standard arguments [e.g. 3]. The really interesting part is proving (4) and (5.2). The major result of this note is the introduction of several multi-processing schemas whose structure is so simple that various path expressions may be asserted as characterizing their behavior. The producer/consumer problem is then shown to be an instance of these schemas, from which (4) and (5.2) follow immediately.

A key part of the proof is based on the following observation. Suppose (7) and (8) are known to hold, and it is desired to show that (9) also holds.

7. path x; y; y* end
8. path y; z end
9. path x; z; z* end

To establish (9) it is sufficient to show that whenever control is in x, then in terms of (8), control is somewhere between y and z. A schema is given in which this sufficient condition holds, and the producer/consumer problem is shown to be an instance of this schema.

[1] Dijkstra, E. W., A constructive approach to the problem of program correctness, BIT 9, 1968, pp. 174-186.

[2] Habermann, A. N., Path expressions, CMU Technical Report, June 1975.

[3] Holt, R. C., Some deadlock properties of computer systems, Computing Surveys, v. 4, n. 3, Sept. 1972, pp. 179-196.

ON THE OPTIMAL COMPUTATION OF A SET OF SYMMETRIC, PERSYMMETRIC OR BOTH SYMMETRIC AND PERSYMMETRIC BILINEAR FORMS

Paolo Zellini
Istituto di Scienze dell'Informazione
Università di Pisa
and Carnegie-Mellon University

We prove that the tensor rank of symmetric, persymmetric and both symmetric and persymmetric matrices, is respectively $\frac{n^2}{2} + \frac{n}{2}$, $\frac{n^2}{2} + \frac{n}{2}$ and $\left[\frac{(n+1)^2}{4}\right]$. For all these matrices a tensor base is found which is expressed in terms of the tensor bases of two special classes of matrices of tensor rank n. This proves that, in the general case, respectively $\frac{n^2}{2} + \frac{n}{2}$, $\frac{n^2}{2} + \frac{n}{2}$ and $\left[\frac{(n+1)^2}{4}\right]$ general multiplications are sufficient and necessary to compute p bilinear forms $f_i = \underline{x}^t A_i \underline{y}$ $(i = 1,\ldots,p)$, where the A_i are symmetric, persymmetric or both symmetric and persymmetric. The same number of general multiplications is also necessary and sufficient to compute the matrix-vector product $A\,\underline{b}$, where A is symmetric, persymmetric or both symmetric and persymmetric. The algorithm which computes $A\,\underline{b}$ is extended to the case A and $\underline{b}$ have their elements in the ring of square matrices of order m.

This work was supported by the National Council of Research of Italy while the author was visiting the Carnegie-Mellon University Department of Computer Science for the academic year 1975-76.