DIMACS Technical Report 98-55
December 1998

Scientific Achievements at DIMACS:

Highlights of the Second Four Years

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Bellcore and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

Contents

Introduction

DIMACS, the Center for Discrete Mathematics and Theoretical Computer Science, is an NSF Science and Technology Center (STC) and is also supported by the New Jersey Commission on Science and Technology. The center is a consortium of Rutgers and Princeton Universities, AT&T Labs - Research, Bell Labs, Bellcore and NEC Research Institute. There are about 175 scientific "members" from the participating institutions at DIMACS active in research, education and knowledge transfer.

DIMACS was founded in February 1989 and in the ensuing years has become a national center for discrete mathematics and theoretical computer science researchers. The center has an active postdoctoral training program, hosts visiting researchers and graduate students, sponsors numerous workshops, seminars and lectures as well as conducting other programs. In addition to a general program of research, DIMACS's "Special Year" and "Special Focus" programs target research areas that are poised for rapid development and offer opportunities for important applications of the methods of discrete mathematics (dm) and theoretical computer science (tcs). Recent focus topics have been

- **Mathematical Support for Molecular Biology**, September 1994 - August 1998.
- **Logic and Algorithms**, August 1995 - July 1996.
- **Networks**, July 1996 - Present.
- **Discrete Probability**, September 1996 - August 1999.
- **DNA Computing**, January 1997 - August 1999.
- **Massive Data Sets**, June 1997 - August 1999.
- **Large Scale Discrete Optimization**, September 1998 - August 1999.

This report highlights some of the many notable research accomplishments of center members, postdoctoral fellows, graduate students, and visitors whose work was impacted by DIMACS' programs. A preceding report, *Scientific Achievements: Highlights of the First Four Years*[1], described the research accomplishments from 1989 to 1993. This report covers the years 1993-1997. This report, like its predecessor, has had to address the difficult question of what results of postdocs, graduate students, visitors, and members should be included. Most included results are directly the result of support of postdoctoral fellows or visitors or collaborations fostered by special year programs, though others are the result of many less specific but no less important influences of the center as a national center that encourages collaborations at workshops, through visitor programs, and in other ways.

The second comment that must be added is that the results in this report are, by no means, the only notable research accomplishments of the center. We have exercised editorial judgement about coverage of the special years and representing the full range of the center's research while keeping the report to a short and readable size. That has meant omitting some results that surely would be part of a thorough description of the center's most important research.

We will use the following designations for how researchers participate in the center.

- (PD) Postdoctoral Fellow
- (V) Visitor
- (PM) Permanent Member
- (GS) Center Supported Graduate Student
- (RU) Rutgers, The State University of New Jersey
- (PU) Princeton University
- (AT&T) AT&T Labs-Research
- (BC) Bellcore
- (BL) Bell Labs
- (NEC) NEC Research Institute.

[1] *Scientific Achievements: Highlights of the First Four Years*, available as DIMACS TR-92-50. Also available at `http://dimacs.rutgers.edu/About/Reports/achievements.html`.

Approximating Numerical Taxonomy

A common problem in phylogeny is to fit an $n \times n$ distance matrix D by a tree metric T. Richa Agarwala (PD), Vineet Bafna (PD), Babu Narayanan (PD), Martin Farach-Colton (PM, RU), Mikkel Thorup (V), and collaborator Mike Paterson presented an $O(n^2)$ algorithm for finding an additive tree T such that the difference between the matrix of tree distances and the distance matrix D is at most three times the optimal. This is the first algorithm for the problem with a performance guarantee. They also showed that it is NP-hard to find such a tree T with the difference at most 9/8 times the optimal. Later, Farach-Colton and Sampath Kannan (V) used this work as a building block in a clever algorithm for computing maximum likelihood trees, which is likely to have a great impact on the field. Experiments undertaken by Jaime Cohen (GS, RU) and Farach-Colton have shown that these distance methods construct better trees in practice than do the most popular heuristics used by biologists.

Ultra-selfish Genes in Mice

The ability of the t-haplotype, an ultra-selfish gene in mice, to persist in natural populations has been a subject of debate in the theoretical biology literature for forty years. Ultra-selfish genes, which have been observed in several species, are genes that are both aggressive and deleterious. They are transmitted to offspring more often than would be consistent with Mendel's laws, yet they are harmful to the organism that carries them in that they cause sterility and often carry recessive lethal mutations. Ultra-selfish genes have also been observed in fruitflies, mosquitoes, wheat, tomatoes and fungi. Because such genes challenge the basic notions of Mendelian genetics and Darwinian evolution, understanding their behavior in populations is an important research problem in evolutionary biology. Dannie Durand (V) has been working on problems in this area in collaboration with Lee Silver in the Molecular Biology department at Princeton since shortly after the Special Year in Mathematical Support for Molecular Biology started in 1994-95. Their first project, addressing the ability of the t-haplotype to persist at low frequencies in natural populations, was reported in *Genetics* in April 1997. Their results overturned the longstanding hypothesis that this low level persistence is due to a balance between loss of the gene due to stochastic effects in small populations and reintroduction of the gene through migration. Using

Monte Carlo simulation, they showed that although the proposed model does predict the gene frequencies seen in nature, the system is not stable and therefore not a biologically plausible explanation for the observed phenomena. Their simulations suggested that, in the context of the proposed model, stability can only be obtained if the aggressive transmission of ultra-selfish genes is reduced through mechanisms such as reduced fitness or female choice. These new hypotheses are now being tested through laboratory and field work.

Nearly Tight Bounds on the Learnability of Evolution

Evolution is often modeled as a stochastic process which modifies DNA. One of the most popular and successful of such processes are the Cavender-Farris (CF) trees which are represented as edge weighted trees. The *Phylogeny Construction Problem* is that of, given k samples drawn from a CF tree, output a CF tree which is close to the original. Each CF tree naturally defines a random variable, and the gold-standard for reconstructing such trees is the Maximum Likelihood Estimator (MLE) of this variable. This approach is notoriously computationally expensive. Andris Ambainis (V), Rick Desper (GS, RU), Martin Farach-Colton (PM, RU), and Sampath Kannan (V) showed that a very simple algorithm, which is a variant on one of the most popular algorithms used by practitioners, converges on the true tree at a rate which differs from the optimum by a constant. They did this by analyzing upper and lower bounds for the convergence rate of learning very simple CF trees, and then showing that the learnability of each CF tree is sandwiched between two such simpler trees. The results relied on the fact that, if the right metric is used, the likelihood space of CF trees is smooth. This is a very exciting result because it makes available a computationally efficient approach which gives almost as good a reconstruction as the computationally inefficient standard MLE method.

An Until Hierarchy for Temporal Logic

As a dramatic result of DIMACS' efforts to bring together two distinct research communities, ideas from finite model theory were used to solve a problem of computer-aided verification (CAV) concerning the expressibility of temporal logic statements. Temporal logic is an extension of propositional logic that allows one to talk about events happening in “time”. It is a logic much used in CAV. During the special year on Logic and Algorithms, Kousha Etessami (PD, RU), Denis Therien (V), and Thomas Wilke (PD, RU) designed a new Ehrenfeucht-Fraisse (E-F) game to capture the power of temporal logic. Ehrenfeucht-Fraisse games are a standard tool in finite model theory, but have not found much use previously in the study of temporal logic. An E-F game, associated with a particular logic, is the main mathematical tool for showing that certain things cannot be expressed in that logic. Briefly, a mathematical structure can be thought of as a game played on a pair of game boards. For Q some property, one player is trying to show that Q distinguishes the two boards, and the other player is trying to show that the two boards are indistinguishable with respect to Q. Depending on which player has a winning strategy in the game, either Q can or cannot be expressed in the logic. As a result of their ideas about E-F games, Etessami, Therien, and Wilke were also able to provide simple and elegant proofs of theorems concerning temporal logic, whose previous proofs involved long and intricate algebraic constructions. The work of Etessami, Therien and Wilke drew the attention of Moshe Vardi (V) to the connection between temporal logic and bounded-variable logics, and it led to his collaboration with F. Gradel (V) and P. Kolaitis (V) in which it was proved that the decision problem for first-order logic with two variables is NEXPTIME-complete, thus solving a classical open problem in logic.

Model Checking of Open Systems

Orna Kupferman (PD, BL) and Moshe Vardi (V) obtained beautiful results on the problem of model checking of open systems. In computer system design, checking if the system meets its specification involves checking if it is a “model” of a logic statement. Such model checking work is one of the most successful and exciting aspects of CAV. A *closed* system is one whose behavior is completely determined by the state of the system; an *open* system is one that interacts with its environment and whose behavior depends on this interaction. Current model-checking algorithms and tools, which are based on temporal logic, and which are designed for closed systems, are not able to take into account uncertainty regarding the environment. Kupferman and Vardi proved both a positive and negative result. They showed that although model checking of closed and open systems coincide for the *linear-time paradigm* in which one possible execution path is under consideration, model checking of open systems is much harder than model checking of closed systems for the *branching-time paradigm* in which all possible execution paths are under consideration.

Verification of Fair Transition Systems

Orna Kupferman (PD, BL) and Moshe Vardi (V) also obtained important results on verification of fair transition systems. A protocol or a chip involves "transitions" between states. "Fairness" is an important property when trying to verify aspects of a protocol. For example, if there are several processes wanting access to a resource (such as the use of a bus, main memory, the network, etc.), the protocol should assure that this is done in a "fair" way, according to a suitable formalization. An important problem in program verification is to check that an implementation meets its specification. Both the specification and the implementation describe the possible behaviors of the program, though at different levels of abstraction. Kupferman and Vardi distinguished between two approaches to implementation of specifications. The first approach is *trace-based implementation*, which requires that every computation of the implementation must correlate to some computation of the specification. The second approach is *tree-based implementation*, which requires that every computation tree embodied in the implementation must correlate to some computation tree embodied in the specification. The two approaches to implementation are strongly related to the linear-time versus branching-time dichotomy in temporal logic. Kupferman and Vardi examined the trace-based and tree-based approaches from a complexity-theoretic point of view, comparing the complexity of verification of fair transition systems. They showed that while both problems are PSPACE-complete, their complexities in terms of the size of the implementation do not coincide and the trace-based approach is easier. The results are surprising because they are at variance with the known results for the case of transition systems with no fairness, where neither approach is evidently advantageous.

Progress on the Berman-Hartmanis Conjecture

The Berman-Hartmanis conjecture, an old and well-known conjecture, first published in 1977, states that all sets complete for NP under polynomial time reductions are *isomorphic* via these reductions: that is they are reducible to each other by functions that are polynomial-time computable, bijective and polynomial time invertible. In an intuitive sense, this says that all the NP-complete sets are trivial re-encodings of each other. However, most complexity theorists currently believe that there exist one-way functions (functions that are easy to compute but hard to invert) and such functions could scramble a set into a non-isomorphic image; this has led to strong doubts about the truth of the conjecture. Manindra Agrawal (V), Eric Allender (PM, RU) and Steven Rudich found evidence for the conjecture, by proving a restricted version: They showed that the sets in NP which are complete under AC^0 reductions (essentially the same thing as reducibility expressible in first-order logic) – rather than those complete under polynomial-time reductions – are also isomorphic via AC^0 reductions. However this is surprising since one-way functions exist in AC^0 if they exist anywhere, and thus the presence of these one-way functions does not stop all complete sets from being AC^0 isomorphic, violating the intuition against the conjecture.

An Optimal Convex Hull Algorithm

The first DIMACS special year, in 1989-90, was on Discrete and Computational Geometry and work begun then continues to lead to exciting new research. The work of Jiri Matousek (V), who visited the center in 1989-1990, on geometric sampling proved to be tremendously influential. In addition to being breakthrough work in *derandomization*, that is, the construction of a deterministic algorithm to "simulate" a randomized algorithm, it was of fundamental importance in computing subdivisions of the plane or higher dimensional space based on random sampling, the basis for geometric divide-and-conquer algorithms. Combining these developments on derandomization with the research by Ken Clarkson (PM, BL) and Peter Shor (PM, ATT) on randomized geometric algorithms laid the groundwork for important new results.

In particular, Bernard Chazelle (PM, PU) developed the first optimal, deterministic algorithms for convex hulls and Voronoi diagrams in any fixed dimension. His method applies derandomization to a randomized geometric algorithm involving random insertion order of facets. This successfully ended an 18 year quest for optimal solutions to these problems. Other derandomization work led to major progress on optimization problems such as linear programming in fixed dimension and computing the diameter of a finite point set in three dimensions. The work of Chazelle, Matousek and others successfully developed and exploited the fundamental relationship between derandomization and discrepancy theory, an area of geometry which focuses on measures of distance from uniformity.

Linear Decision Trees: Volume Estimates and Topological Bounds

Many combinatorial and geometrical computational problems can be formulated as membership questions: given inputs $x_1, \ldots, x_n$, decide whether the point $x = (x_1, \ldots, x_n)$ is in S, a set of points in R^n. Could one link the computational complexity of a membership question to familiar properties of the set S? A series of results by David Dobkin (PM, PU) and Richard Lipton (PM, PU), Mike Steele and Andrew Yao (PM, PU), Michael Ben-Or, and others showed that, for various versions of decision tree models, the decision tree must have depth $\Omega(\log(C(S))$, where $C(S)$ is the number of path-connected components of S. The natural question was whether other standard properties of S may also be linked to decision tree complexity. A new result of Andres Bjorner, Laszlo Lovasz (PM, PU) and Andrew Yao (PM, PU) gave a positive answer, showing that for the membership question, any linear decision tree must have depth $\Omega(\log |E(S|)$, where $E(S)$ is the Euler characteristics of S. They applied this result to the k-equal problem, which decides for given n input numbers whether there exist k of them equal to each other. For this problem, they also developed another approach based on the volume of S and how much it has to be reduced during the decision process before a definite answer to the membership question can be determined. This result is important in its showing new connections between computational complexity and algebraic and topological properties of geometric sets, and thereby obtaining new lower bounds on complexity.

Experimental Results on the Traveling Salesman Problem

Experimental computing is a key continuing theme at DIMACS. DIMACS' emphasis on this subject led to the visit in 1990-91 by Lyle McGeoch (V) of Amherst College, and to a continuing collaboration between McGeoch and David Johnson (PM, BL, now ATT) together with Ed Rothberg, which recently led to major results on the Traveling Salesman Problem (TSP). The TSP is easy to state: given a finite number of "cities" along with the "cost" of traveling between each pair of them, find the cheapest tour visiting all the cities and returning to the starting place. It gained a reputation as a prototype *hard* combinatorial optimization problem. The new results derive from extensive experimental runs analyzing the gap between the Held-Karp traveling salesman bound and the optimal value for a variety of instance classes, real and probabilistic. The Held-Karp bound is a lower bound on the optimal tour length that is significantly easier to compute and can thus stand as a surrogate for the optimal when attempting to evaluate the quality of heuristically-derived tours. One key question is just how good a surrogate the Held-Karp bound is. Using recently developed algorithms for computing and approximating this bound, together with massive computing power, Johnson, McGeoch, and Rothberg performed an extensive investigation of this question. For the real-world instances in the standard testbed, the gap between the Held-Karp bound and the optimal tour length was seen to be typically around 1%, and, for random Euclidean instances, it averaged under 0.7%, meaning that this is a very good bound indeed. For the random classes, they also investigated the asymptotics of the expected values of the Held-Karp bound and, by extrapolation based on detailed statistical analysis of the gaps, the expected optimal tour length. These results have already proved their value in evaluating the quality of earlier reported work on the TSP.

Application of TSP Approximations to Code Generation

David Johnson (PM, BL, now ATT), David Karger (PD, BL) and Cliff Young and Michael Smith studied a novel application of the Traveling Salesman Problem (TSP) to optimizing compilers. The machine code for a given subroutine can be divided into basic blocks, each consisting of a sequence of instructions which must be permuted into a single instruction stream with the expected delay penalty $d(a, b)$ if basic block b is placed immediately after basic block a in the instruction stream. To order the instructions so as to minimize the expected total delay, one has to solve a directed TSP, with delay penalties corresponding to distances.

Johnson and colleagues used the TSP methodology to obtain improvements for some of the standard SPECmark codes that are used to benchmark processor speed. The resulting expected pipelining delays averaged within 0.3% of optimal (bounded using Held-Karp code developed by David Applegate (PM, BL) and Bill Cook (PM, BL)). These delays were 36% less than what the standard compilers yielded.

A Polynomial Time Approximation Scheme for Euclidean TSP

The traveling salesman problem, though NP-complete, has variants that arise in applications that give it additional structure. The general TSP requires only that $d(a, b) = d(b, a)$. The *metric* TSP problem adds the requirement that the distances satisfy the triangle inequality. The *Euclidean* TSP requires that the cities lie in R^d and that the distances be those given by the ℓ_2 norm. These variants are also NP-complete.

Even if NP-complete problems are not polynomial time solvable, one could hope that a *polynomial time approximation scheme* (PTAS) exists; this is a family of algorithms such that for any $\epsilon > 0$, an algorithm in the family can solve problem instances to within a factor of $1 + \epsilon$ in polynomial time (the specific polynomial can depend on ϵ). In work that was a result of DIMACS' earlier special years, the hopes for PTAS were dashed: it was proved by Sanjeev Arora (PM, PU), Carsten Lund (PD, now PM, BL), Rajeev Motwani, Madhu Sudan and Mario Szegedy (PD, PU, now PM, ATT) that the metric TSP problem has no PTAS.

Surprisingly, the restriction to Euclidean TSP does allow a polynomial time approximation scheme, as proven by Arora in 1996. This also improves on the best prior approximation of 3/2 (due to N. Christofides in 1976). The methods extend to new PTAS for several related problems including the Steiner Tree Problem, k-TSP, Minimum-Degree-k Spanning Tree, k-MST and others. The algorithms also work, with minimal modifications, for ℓ_p norms for $p \geq 1$ and other Minkowski norms.

Factoring Numbers with Quantum Computing

Factoring integers has long been considered a hard computational problem. It is widely believed that integers cannot be factored in time polynomial in the number of bits or digits that represent the number. So widely is this believed that cryptography methods such as RSA rely on the difficulty of factoring. One of the cornerstones of basing a cryptographic method on the hardness (that is non-polynomial time) of a problem is that the definition of what is possible to compute in polynomial time is independent of the model of computing used.

Quantum computing (using devices based on quantum mechanics) was proposed as far back as 1982 by Richard Feynman, but only recently has attracted a lot of interest. Ethan Bernstein and Umesh Vazirani in 1993 gave an oracle problem that required more than polynomial time for ordinary computing models, but could be done in polynomial time in the quantum computing model. Dan Simon improved this result to give a problem requiring exponential time on an ordinary computer but polynomial time on a quantum computer.

In 1994, Peter Shor (PM, ATT) proved a remarkable result for factoring: that a number can be factored in polynomial time in the quantum computing model. A similar result was also obtained by Shor for the discrete log problem which is the basis of another cryptography method (Diffie-Hellman). Thus, for problems of very practical interest, the quantum model has significantly faster algorithms than are known for the traditional computing models. These results not only demonstrate that quantum computing can give new power to computing, they also demonstrate this on problems of practical importance. In the long run, they also cast doubt on the safety of cryptography based on the hardness of factoring or discrete-log.

An Improved Computer Proof of the Four Color Theorem

In a collaboration that goes back many years and was strengthened in the DIMACS special year on graph theory and algorithms, Paul Seymour (PM, PU), Dan Sanders (V), Neil Robertson (V) and Robin Thomas (V) gave an improved proof of the four-color theorem. The theorem is easy to state: if a graph can be embedded in the plane with no crossing edges, then its vertices can be colored with four colors so that no adjacent vertices have the same color. The four color problem has been one of the motivating problems driving graph theory research for over a hundred years; indeed it led to many branches of modern graph theory that were born from attempts to prove the conjecture - for instance graph coloring, edge-coloring, Hamilton circuits, matching theory and planarity.

The original proof of the four color theorem in 1977 by Haken and Appel was unsatisfactory in several respects:

(a) it was very long, complicated and tedious;

(b) it required a computer;

(c) some of the details were not very clear and led to a prolonged debate about whether the proof was correct or not (which was fueled by the fact that no one seemed to have checked the paper completely).

The new proof by Robertson, Sanders, Seymour and Thomas, while still relying on a computer, is simpler in its details and thus more accessible to complete checking (which has been done by several researchers). An additional benefit is that the new proof can now be modified to attack other, related problems. For instance, Sanders and Thomas modified the proof to show that cubic graphs that are planar except for one vertex can be three-edge-colored (the four-color conjecture says the same thing, but only for cubic graphs that are completely planar); and Tom Fowler and Thomas modified the proof to show that no planar graph is uniquely four-colorable except the ones known.

Solution to the Even Cycle Problem

A collaboration among Neil Robertson (V), Robin Thomas (V), and Paul Seymour (PM, PU), further developed during the DIMACS special year on graph theory and algorithms, led to the solution of the long-standing open problem of finding a polynomial time algorithm for the following: given a directed graph, decide if there is a directed cycle of even length in it. This problem originated from the famous mathematician George Polya in connection with evaluating the permanent of a matrix. It is also known to be equivalent to many other problems, including matrix sign-stability (a very important problem in economics that goes back to Nobel Prize winner Paul Samuelson), Pfaffian orientation of bipartite graphs, and variants of hypergraph two-coloring. Essentially, Robertson, Seymour, and Thomas explicitly characterized all digraphs which have no even directed cycle. They were all built from certain planar digraphs and one exceptional one by gluing all digraphs together on small cutsets. Then, to test if a digraph has no even cycle, one tests if it has this structure.

Dispersers and Randomized Algorithms

Michael Saks (PM, RU), Shiyu Zhou (GS, RU), and Aravind Srinivasin (PD, RU) collaborated on work significantly strengthening fundamental results on randomized algorithms. For many applications, a randomized algorithm is the simplest algorithm available, or the fastest, or both. An area of considerable research interest in recent years concerns how dependent randomized algorithms are on the quality of the random source: Given an algorithm that uses random bits, can it be made robust in the sense that it works correctly even if the bits deviate from randomness to some extent? What is desired is a general method that converts any algorithm that uses random bits to one that still works correctly if the random bits are faulty. Aviad Cohen and Avi Wigderson established that for such a conversion method to be possible, the random source needs a certain minimum amount of intrinsic randomness. Saks, Zhou, and Srinivasin gave the first construction to show that the Cohen-Wigderson conditions were sufficient. Their construction is obtained via a new construction of a class of disperser graphs, which are graphs whose edges are (in some precise sense) well distributed. This construction also provided an affirmative answer to a question from computational complexity theory published by Mike Sipser in 1988.

Span Programs

One of the most important and perplexing challenges in theoretical computer science (and indeed in all of mathematics) is the problem of proving that certain problems are hard to compute. The P vs. NP problem is the most famous problem of this sort, but it is equally important to be able to prove that certain Boolean functions cannot be represented by small formulae, or computed by certain sorts of circuits, etc. Often these problems take the form of showing that one complexity class is properly contained in another. As just one sign of the importance of this research topic, note that progress along this front is necessary before public-key cryptography can be placed on a rigorous footing. Yet, despite a great deal of effort in this direction, progress has been modest. One of the major obstacles blocking progress toward this goal is the "sequential" nature of computation as it evolves over time. Proving that no circuit (or algorithm) of a certain kind can exist means that, in some sense, the tree of all possible execution sequences must be examined and that it must be shown that none of these sequences can perform a certain task.

Thus it is exciting that researchers at DIMACS have made considerable progress on showing that certain functions require large monotone formula size (and thus are "hard to compute" in this sense), using a new model of computation called a span program. Span programs are of great interest because (a) proving that a function requires large span programs implies that it requires large formulae and large circuits from a rather powerful class of circuits, and (b) some functions have been shown to require large span programs, in a way that avoids dealing with the "sequential" nature of computation. (That is, the problem of proving lower bounds for certain classes of circuits reduces to proving lower bounds on span program size, which in turn is a purely combinatorial question, with none of the additional complications caused by the evolving nature of computation.) Work of Laszlo Babai, Ann Gal (PD), Janos Kollar, Lajos Ronyai, Tibor Szabo, and Avi Wigderson (V) gave lower bounds for monotone span programs, complementing existing lower bounds for monotone circuits, which are also of great interest in complexity theory. This work in turn implied explicit lower bounds for monotone Boolean formulae and certain kinds of circuits. Later work by Babai, Gal, and Wigderson gave the first superpolynomial lower bounds for monotone span programs by exhibiting a function computable in this model in linear time while requiring superpolynomial size monotone circuits and exponential size monotone formulae. Their work has implications for the complexity analysis of natural and important problems in graph theory such as the problem of determining if a graph contains a perfect matching.

The work of Babai, et al., referred to above, concerns span programs over finite fields. Until recently, not much was known about span programs over infinite fields, such as the rationals. A result by Eric Allender (PM, RU), together with Robert Beals (PD, RU) and Mitsunori Ogihara, gave an exact characterization of the computational power of uniform span programs over the rationals with implications for the complexity analysis of natural and important problems in linear algebra such as the problem of determining if a system of linear equations is feasible or the problem of computing the rank of a matrix. This work contained the first significant results in characterizing problems from linear algebra in terms of complexity classes and has been influential in leading to several further results at the interface between complexity theory and linear algebra.

Purely Functional Programming

Purely functional programming, in which only a constrained set of functions is employed, has received much attention, not only because functional languages have many desirable properties, but because data structures built purely functionally are automatically *fully persistent*: Any and all versions of a structure can coexist indefinitely so that changes to the data structure can be made without destroying the old version. A number of results illustrate the surprising power of pure functionality in the construction of persistent and efficient data structures. One such result was the development of a representation of double-ended queues with catenation that supports all operations in worst-case constant time. Haim Kaplan (GS, PU, now PD, ATT) and Robert Tarjan (PM, PU) developed methods for purely functional representations of sorted lists, implemented as finger search trees or, that is, balanced search trees in which access in the vicinity of certain preferred positions, indicated by "fingers," is especially efficient. The most efficient of three implementations they developed achieves logarithmic access, insertion, and deletion time, and double-logarithmic catenation time. The bounds to find, insert, and delete were the same as the best known bounds for an ephemeral implementation of these operations using finger search trees. The representations developed are the first purely functional representations of sorted lists and the first for which fast implementations of catenation and split are presented. They are also simple to implement.

The Virtual Circuit Holding Time Problem for IP-over-ATM

Steven Phillips (PD, now PM, ATT) developed new, cost-effective approaches for the virtual circuit holding time problem for IP-over-ATM networks. This problem arises at the interface between an IP (Internet Protocol) network and an ATM (Asynchronous Transfer Mode) network since there is a mismatch between the packet switching communications techniques employed by local internet networks used for functions such as email and web browsing and the circuit assignment-and-hold communications techniques employed by high capacity backbone networks into which these local networks feed. The central issue is to determine how long to hold a circuit connection once it has been set up and assigned to an incoming packet in view of the uncertainty about the arrival time of the next packet, and the essential theoretical issue is to predict packet interarrival times. Phillips's approach was to estimate the distribution of packet interarrival times for each conversation from observations, and use the modeled distributions to determine cost-reducing circuit hold policies. He discovered that, with a small amount of overhead to collect the necessary information, policies can be implemented which reduce circuit set up and hold costs 10-15% over policies typically used in practice.

In addition, Phillips observed an interesting and extremely useful analogy between the virtual circuit holding time problem and the analysis of online spin-locking algorithms which arise in multiprocessor computer systems, thus providing an approach to develop new spin-locking protocols. In this environment, a parallel program with multiple threads is running, and the threads may need to coordinate. One protocol is the spin-lock approach in which the first thread to access a piece of code can lock onto it while the next thread to arrive must spin (consuming processor resources) or block (with a start-up cost when it resumes) until the held code is released. Again, the issue is to predict the length of time until a lock will be released in order to establish whether to spin or block. (In the theoretical computer science literature this problem is sometimes referred to as the ski-rental problem in which the beginning skier wishes to assess his or her interest in skiing in an inexpensive fashion by renting equipment for some number of ski trips, or, that is, the small cost of keeping an established circuit open before investing in purchased equipment, or, that is, the high cost of closing down an existing circuit to which is attached the cost of opening the next one.)

The Russian Option for Stock Hedging

Larry Shepp (PM, BL, now RU) and Albert Shiryaev (V) proved a theorem in pure probability that gives the stopping time which maximizes the expected time-discounted supremum of an exponential Brownian motion and an explicit solution for this optimal value. The exponential Brownian motion is the widely accepted Black-Scholes model for asset fluctuations, and thus the theorem can be interpreted as an explicitly valued stock market hedge option. In particular, the results of Shepp and Shiryaev give a simple exact formula for the fair price of the option along with the optimal exercise strategy. Theirs is the first exact formula for the expected present value or fair price for any of several similar option types. Currently, this option, known as the Russian option, has become widely discussed on Wall Street and is beginning to be sold as a standard long buy hedge option.

Auctions for Combinations of Lots

The theory of auctions and bidding has enormous practical significance, and is used widely in industry, for instance in the recent bids by telecommunications companies for pieces of the FCC radio spectrum and in oil lease bids. Yet, many fundamental problems of auctions and bidding remain unresolved because of their combinatorial difficulty. Aleksandar Pekec (GS, RU), Michael Rothkopf (PM, RU), and Ronald Harstad (V) collaborated to study simultaneous auctions for situations in which the value of assets to a bidder depends on which other assets he or she wins. In such cases, bidders may well wish to submit bids for combinations of assets. When this is allowed, the problem of determining the revenue maximizing set of nonconflicting bids can be a difficult one. Pekec, Rothkopf, and Harstad combined methods of combinatorics with techniques from the theory of bidding to identify different structures of combinatorial bids for which computational tractability can be constructively demonstrated. This work shows that computationally manageable auctions are possible in a number of economically important situations including certain sales of spectrum rights and airline takeoff and landing slots. In fact, this work proved extremely influential in that the FCC recently let a contract for software to allow auctions in which bidders can bid on some combinations of licences. This kind of auction is an innovation for the FCC and is based on groundwork laid by the work of Rothkopf, Pekec and Harstad.

Dependence of Optimal Solutions on Scales of Measurement

Mathematical formulation of an optimization problem often depends upon data which can be measured in more than one acceptable way. Motivated by practical problems of scheduling, Aleksandar Pekec (GS, RU), Fred Roberts (PM, RU), and N.V.R. Mahadev (V) found examples where an admissible change of scale of measurement can change an optimal solution to a problem into one that is no longer optimal, making it questionable as to whether or not it makes sense to claim optimality. They also identified conditions under which this anomaly can be avoided. The results provide an important warning for those who solve optimization problems that involve variables measured in some way and in particular have been applied to problems of "just-in-time" scheduling in manufacturing and to problems of transportation when certain items to be transported have higher priority than others. Pekec also showed that under certain general conditions on a combinatorial optimization problem, if a solution remains optimal under all possible admissible changes of scale and the scale is admissible up to ordinal transformations, then the greedy algorithm will give an optimal solution.

The $\mathcal{Z}_4$ Linearity of Kerdock, Preparata and Other Codes

Most work on codes for error correction in communication over unreliable channels has emphasized linear codes consisting of all vectors x over some finite field (commonly the integers mod 2 for binary codes) such that $Ax = 0$, for some matrix A. These codes have a clean structure that makes them simple to discover, to understand, and to encode and decode. However, certain nonlinear binary codes contain more codewords than any known linear code and so would be preferred to linear codes if efficient encoders and decoders were known for them. Two of these nonlinear binary codes, the Kerdock and Preparata codes, have long been known to be dual to one another in a particular weak formal sense. A stronger notion of duality is known to hold for pairs of linear codes, suitably chosen. For roughly 20 years this curious observation left coding theorists with the open question of whether these nonlinear codes might not be duals in some stronger sense. In the course of examining this question, Rob Calderbank (PM, BL, now ATT), Roger Hammons, Vijay Kumar, Neil Sloane (PM, BL, now ATT) and Patrick Solé (V) obtained the stunning result that the Kerdock and Preparata codes are in fact linear, if one views them in the right way over the ring of integers mod 4 instead of mod 2, where the mapping between the mod 2 and mod 4 versions of the codes is extremely simple, based on a Gray code. Furthermore, over this larger alphabet, the two codes *are* duals in the stronger sense. As a result of recognizing the underlying linear structure of classes of nonlinear binary codes, new techniques for the design of efficient encoders and decoders are now available for powerful code classes.

A Randomized Linear Time Algorithm for Finding Minimum Spanning Trees

The Minimum Spanning Tree problem is to take as inputs a set of nodes and weights between edges in a graph and to find a tree connecting the nodes that minimizes the sum of the edge weights. This problem is motivated by applications such as finding minimal cost communication networks to connect a set of locations. The problem has a long history with many algorithmic approaches. The best algorithms previously had been a deterministic algorithm (of Gabow, Galil and Spencer) that used time $O(m \log \beta(m,n))$ on a graph with n nodes and m edges (where $\beta(m,n) = \min\{i | \log^i n \leq m/n\}$) and a randomized algorithm (by Karger) that runs in $O(m + n\log(n))$ time. Fredman and Willard also had a linear time algorithm that uses a model that allows bit-manipulation of edge weights.

Philip Klein (V) and Robert Tarjan (PM, PU) devised a new randomized algorithm in the unit-cost RAM model for the minimum spanning tree problem that runs in $O(m)$ time, which is optimal. The new algorithm uses only comparisons of edge weights.

A PCP Characterization of NP

A probabilistically checkable proof (PCP) system consists of a language L together with a verifier which can recognize that a string x is in L, but if x is not in L, then the verifier determines this fact only with high probability. Arora and Safra characterized NP in terms of PCP systems, and Sanjeev Arora (PM, PU), Carsten Lund (PD, now PM, ATT), Rajeev Motwani, Madhu Sudan and Mario Szegedy (PM, ATT) improved on their result by showing that $NP = PCP(\log n, 1)$. That is, the set of computationally intractable problems comprising NP is equivalent to a particular PCP system, one in which the probabilistic polynomial time verifier V uses $O(\log n)$ coin flips and queries only a constant number of random bits from the verifier's membership proof for x.

This result has the extremely significant consequence that many problems in NP cannot be approximated closely unless P=NP. The result says that proofs which require polynomial time for exact verification based on examination of the entire proof can be verified in only linear time and using only a constant number of randomly selected bits from the proof if it is sufficient to retain some uncertainty about the outcome of the verifier. But these verifiers can be thought of as robust in the sense that small changes in the bits of the proof do not affect the verifier's outcome in a probabilistic sense. Because there is a (highly technical) connection between robust proofs and approximate solutions to optimization problems, that is solutions which are close to the optimal, the characterization result of Arora et al. implies that some intractable problems in NP remain computationally intractable even if only an approximate solution is desired. This resolves what had been open questions about the complexity of approximation algorithms for general classes of optimization problems.

Probabilistically Checkable Debate Systems

Anne Condon, Joan Feigenbaum (PM, ATT), Carsten Lund (PM, ATT), and Peter Shor (PM, ATT) obtained important new results concerning randomized probabilistically checkable debate systems (RPCDS) which are a natural extension of the probabilistically checkable proof systems of Arora, Lund, Motwani, Sudan and Szegedy described in the previous example. In these debate systems, a probabilistic polynomial time verifier V observes part of a debate between two players, one of which is trying to convince V that a string x is in some set L, the other of which is trying to prove it is not in L. The results of Condon, et al. have characterized sets L in PSPACE as those that have an RPCDS in which the verifier uses $O(logn)$ coin flips and reads only a constant amount of the debate.

This new characterization allowed them to characterize approximation versions of certain problems in PSPACE in much the same way that PCP systems were used to characterize approximation versions of certain problems in NP. Condon, Feigenbaum, Lund and Shor showed that optimization versions of dynamic graph reliability, stochastic satisfiability, Mah-Jongg, stochastic coloring, stochastic generalized geography, and other "games against nature" (as introduced by Christos Papadimitriou) are as hard to approximate as they are to compute exactly.

An Approximation Scheme for Network Reliability

David Karger (PD, ATT) obtained major results on an important problem in communications related to network reliability under edge failures, the so-called "all-terminal network reliability problem". Given a network in which each edge has a specified failure probability, the goal of the all-terminal network reliability problem is to compute the probability that the edge failures disconnect the graph. This is one of the original #P-complete problems which are intractable (and possibly harder than NP-complete). Because of the obvious practical applications of this problem to communication networks and its believed difficulty of being solved exactly, a great deal of research has been devoted to approximate solutions. Karger showed how to give a *Fully Polynomial Approximation Scheme* (FPAS) that, for any ϵ, gives an approximation to the answer that is accurate to within relative error $1+\epsilon$ in time polynomial in the network size and ϵ. Deterministic FPASs for nontrivial problems are quite rare; however, several randomized FPASs have been given for #P-complete problems such as counting maximum matchings in dense graphs and measuring the volume of a convex polytope. Thus Karger's randomized FPAS for the all-terminal network reliability problem resolves what had been one of the remaining open problems needing an approximation scheme.

LINK: An Environment for Interactive Graph Theory and Combinatorics

LINK is a software system designed to be a general-purpose environment for experimentation with discrete mathematics. Researchers designing LINK were Jonathan Berry (PD), Nathaniel Dean (PM, BL), Mark Goldberg (V), Greg Shannon (V), and Steve Skiena (V). The LINK environment is friendly enough so that non-technical users are able to visualize problems easily, yet powerful enough for computer scientists and mathematicians to express complicated computations. The system is designed to support educational pursuits, to be used as a research tool, and to be useful in solving real-life problems. (LINK's release date was July 1996, and it is available at the DIMACS web site.)

One of the exciting developments with LINK has been its connection with applications at Los Alamos National Laboratory (LANL). At LANL, LINK is being used as the foundation for a system to aid the Internal Revenue Service (IRS) in fraud detection - using graphical clustering techniques to identify similar (possibly fraudulent) returns. In March 1995, IRS tax examiners started using this LINK-based application, Inspect, in their work to investigate organized fraud in electronically filed personal income tax returns. The Link Inspect software tool has now been installed at all IRS service centers for use by tax examiners.