Theoretical Computer Science, TCS

Article · November 1999

CITATION
1

READS
3,443

4 authors, including:

Stefan Arnborg
KTH Royal Institute of Technology
83 PUBLICATIONS **4,611** CITATIONS
SEE PROFILE

Viggo Kann
KTH Royal Institute of Technology
98 PUBLICATIONS **4,996** CITATIONS
SEE PROFILE

Jens Lagergren
KTH Royal Institute of Technology
126 PUBLICATIONS **3,896** CITATIONS
SEE PROFILE

Theoretical Computer Science, TCS

Stefan Arnborg and Johan Håstad

The objective of the research in Theoretical Computer Science is to investigate methods of efficient computation in a mathematically precise sense, and to find lower bounds on the computational resources required for a computation. The types of computations studied are either chosen for their tractability or for their importance in practical applications. The different areas of study are described below.

During the period 1993-1999 the group has been supported by Nutek, BFR, TFR, NFR, SSF, and HSFR.

The objective of the research in Theoretical Computer Science is to investigate methods of efficient computation ...

http://www.nada.kth.se/theory/

Approximation algorithms

Johan Håstad, Viggo Kann, Jens Lagergren

A large number of the known NP-complete problems are in fact optimization problems, and for some of these optimization problems there are fast approximation algorithms, i.e. algorithms guaranteed to find close to optimal solutions. For example, the travelling salesperson problem in the plane is NP-complete, but in polynomial time it can be solved approximately within every constant, i.e. for any $\varepsilon>0$ one can find a trip of length at most $1+\varepsilon$ times the shortest trip possible. On the other hand, some NP-complete problems are extremely hard to approximate. For example, unless P=NP the maximum independent set problem cannot in polynomial time be approximated within $n^{1-\varepsilon}$ for any $\varepsilon>0$, where n is the number of vertices in the input graph.

The main topic of this project is to investigate to what extent the optimum value of important NP-complete problems can be approximated effectively. These investigations are naturally divided into two types of activities, namely to prove positive and negative approximation results. To get a positive result—an upper bound of the approximability—one constructs an algorithm, proves that it is efficient and approximates the problem within a certain accuracy. To get a negative result—a lower bound—one usually proves that approximating a given problem remains NP-hard.

We have studied the approximability of several basic and important combinatorial problems. Positive or negative results have been found for e.g. some generalizations of the maximum cut problem, maximum number of satisfied linear equations mod p, maximum p-section, and the travelling salesperson problem with weights 1 and 2, see e.g. [Amaldi and Kann, 1998], [Andersson, 1999], [Andersson and Engebretsen, 1998], [Andersson, Engebretsen, and Håstad, 1999], [Engebretsen, 1999], [Håstad, 1999], [Kann, Lagergren, and Panconesi, 1998].

Viggo Kann and Pierluigi Crescenzi have compiled a list of the best lower and upper bounds known for more than two hundred well-studied NP-complete optimization problems. We try to collect all new results in the wide area of approximation in order to keep the problem list updated. The list is included in a new text book on approximation [Ausiello *et al*., 1999], and is also available for everybody on the web as http://www.nada.kth.se/~viggo/problemlist/. It has frequently been used by researchers throughout the world in the last five years, see [Kann and Crescenzi, 1998].

Complexity

Johan Håstad, Mikael Goldmann

Computational complexity involves studying the amount of computational resources required to solve certain types of problems. Resources typically considered are, for instance, the time (number of steps) of an algorithm, or the size and depth of Boolean circuits.

Computational complexity involves studying the amount of computational resources required to solve certain types of problems.

The general area in which we have worked is that of establishing lower bounds for natural functions in restricted computational models. Two models of computation where this program has been especially successful are small-depth circuits and monotone circuits. The most significant steps in both models were taken in the 1980's but developments during the last decade have increased our understanding considerably.

Our main results in the area of monotone complexity are the symmetric approximation method introduced by Berg and Ulfberg (see Ulfberg's thesis, [Ulfberg, 1999]), and the lower bounds for the depth of monotone circuits computing connectivity by [Goldmann and Håstad, 1998]. The former very much simplifies the rather complicated combinatorial arguments that were previously used for proving lower bounds for monotone circuits.

For small-depth circuits a number of results were established using similar methods. [Lai, Chen and Håstad, 1998] studied how bottom fan-in affected the computational power of small-depth circuits, and Berg and Ulfberg used circuit complexity techniques to construct oracles that separate the levels of the PP^{PH} hierarchy (see [Ulfberg, 1999]).

As for more general models, [Håstad, 1998] also established the strongest lower bound on the formula size of any function in NP.

Cryptography

Johan Håstad, Mikael Goldmann, Mats Näslund

Our work has been mostly foundational. A basic concept in cryptography is that of a one-way function, a function that is easy to compute and hard to invert.

It has been known for a very long time that given a one-way function with suitable structure (e.g., a function that is one-to-one) it is possible to construct a strong crypto system. Results by [Håstad *et al*., 1998] show that in fact no structure is needed, and any one-way function is sufficient to construct a strong crypto system.

Strong crypto systems can be constructed through the use of good pseudorandom generators

Strong crypto systems can be constructed through the use of good pseudorandom generators. A very convenient way to construct a pseudorandom generator is to iterate a one-way function f and on each iteration output one or more bits obtained from the current value of the iterate. That is, starting from a seed x_0 one iterates the function producing $x_1 = f(x_0)$, $x_2=f(x_1)$ and so on, each time outputting a bit $B(x_0)$, $B(x_1)$, $B(x_2)$.... To prove that such a scheme is secure (i.e., that the sequence of bits "looks random") one would typically need that the bit output is a so-called hard-core predicate. This means that $B(x)$ is easy to compute given x but hard when only given $f(x)$.

Näslund proved that for any one-way function the standard pairwise independent hash-functions do provide hard-core predicates. These are very simple functions and Goldmann and Näslund proved that this is about as simple as possible for functions which one can hope to have this property (see Näslund's thesis, [Näslund, 1998]). To be more specific, they proved that any predicate that is hard-core for any one-way function requires, when computed by unbounded fan-in circuits, circuits of almost logarithmic depth. This work is currently being extended to a wider class of functions by [Goldmann *et al*., 1999].

For specific functions one can of course have a simpler hard-core predicate. The most famous function in cryptography is RSA-encryption and it has been known for over a decade that the least significant bits give hard-core predicates for this function. Settling a long standing conjecture this was extended to each individual bit by [Håstad and Näslund, 1998]. The same result applied to the discrete logarithm function, the second most used function in public-key cryptography.

Decomposability

Stefan Arnborg, Jens Lagergren

The theory of tree-decomposable graphs became a foundation for, for instance, BDD technology and Bayesian networks, two areas which have numerous applications. Our recent work includes better bounds for obstruction sizes in [Lagergren, 1998], and an improved way of handling independence assumptions in imprecise probability theory [Arnborg, 1999a]. Studies of dense decomposable graph families have been made, like graphs of bounded clique-width and NLC-width (see [Johansson , 1998]). The first progress in the (since 1985) open problem of complexity of decomposition of dense decomposable families except co-graphs was obtained in [Johansson, 1999], with a polynomial time algorithm for NLC2-decomposition.

Formal methods based software testing

Karl Meinke

In software engineering, the classical approach to validating the correctness of software consists of testing a system implementation against its requirements. The testing process involves: (i) collecting an ad-hoc, often random, collection of sample points from the entire input space, known as the test set, (ii) executing the system on each input from the test set, and (iii) evaluating the outcome of each such execution with reference to the software requirements.

Step (ii) is uncontroversial. However, step (i) leads to a fundamental problem known as the coverage problem – this is essentially the likelihood that an implementation error is not discovered by the test set. Furthermore, step (iii) leads to an equally fundamental problem, interpreting the outcome of each test, and reaching one of the judgements: pass/fail/don't know. This is known as the oracle problem.

It seems fair to say that both the coverage and oracle problems are still extremely poorly understood scientifically. This is despite the fact that software testing is carried out intensively and widely within the IT industry.

The methodology of program testing seems to oppose the methodology of formal methods, which advocates that programs be mathematically proved correct (for all possible input values). Although formal program verification has a completely sound scientific basis, it is often difficult to put into practise for real programs, and for certain data types, notably the floating point type.

In fact, it is possible to see program verification and program testing as complementary approaches. Starting from a set R of program require-

ments, expressed in some logic, verification seeks a proof of R while testing seeks a satisfying assignment for the negation of R. Thus (by Gödel's Completeness Theorem) testing and verification are in a strong sense dual.

During the period in question, we have carried out research into formal methods based black box software testing. This research includes:

(i) Case studies of software requirements, particularly in the domains of computational finance (Black-Scholes) and control systems design (automotive systems). Characteristic of these areas is the need for floating point computation, and thus the limited applicability of present day formal verification methods.

(ii) Algorithm design for automated testing. We develop decision algorithms for the satisfiability problem for first-order logic over finite large cardinality data types. First order formulas are used to define program pre- and postconditions which can provide an oracle during testing. Our approach to algorithm design is based upon function approximation theory. The basic strategy is to iteratively search for a satisfying assignment, using the outcome of searches to construct an approximate model of the system, from which we can prune the search space.

Future research in this area will include:

(iii) Theoretical study of the coverage problem. We are attempting to find an appropriate stochastic model for the variance between the functional behaviour of programs and their approximate models in the sense of (ii). For example, is a random walk model appropriate? Such models could be used to estimate coverage in a mathematically precise way.

Results in this project are presented in [Abdulla *et al*.,1999], [Berg *et al*., 1999], and in [Meinke and Nielsen, 1999].

A description of our research project for the layman can be found on the web at http://www.nada.kth.se/\~karlm/Testing.htm .

Computational biology

Jens Lagergren, Henrik Eriksson

In biology, perhaps the most important discovery during this century is that of DNA and the genetic code.

Algorithms on strings (sequences of characters) have been studied since the birth of computer science. In biology, perhaps the most important discovery during this century is that of DNA and the genetic code. For a computer scientist it is natural to formulate the most fundamental conclusion of these discoveries as: the genetic information of any being is a string over a four letter alphabet. Also proteins can be viewed as strings. Many other structures are common to biology and computer science. For instance, evolutionary trees are important in biology, and trees in general are fundamental to computer science.

Recent advances in molecular biology have lead to a large number of

biologically motivated algorithmic problems. For instance, sequencing of DNA constitutes a partly algorithmic problem. Moreover, it gives rise to an enormous amount of discrete data, and the analysis of this data often gives algorithmic problems. Many of the most interesting problems are NP-complete, and thus expected to be difficult to solve, but sometimes there are ways around this dilemma. One can consider approximation algorithms, fixed parameter variations etc.

One fundamental problem in computational biology is to measure the similarity between to genomes.

One fundamental problem in computational biology is to measure the similarity between two genomes. The Sequence Alignment problem is to, given two DNA sequences, insert blanks into the sequences in such a way that they become as similar as possible.

Say that we are given two genomes as sequences of genes. One can measure the distance between these two genomes by counting the least number of operations that transform one into the other. One example of an operation that is reasonable to use (since it mimics an operation on the genome that takes place during evolution) is reversal, that is any subsequence of the gene-sequence may be reversed. We have studied the algorithmic part of a procedure to obtain the gene sequences of a chromosome – namely Radiation Hybrid mapping see [Håstad *et al*., 1998], and [Ivansson, and Lagergren, 1999].

Another fundamental problem in computational biology is that of inferring the evolutionary history of a set of species, given some representation of the species.

Another fundamental problem in computational biology is that of inferring the evolutionary history of a set of species, given some representation of the species. The evolutionary history is represented by a phylogenetic tree. The vertices of a phylogenetic tree are labelled with species, given species or inferred species (representing ancestors of the given species). Two adjacent nodes of the phylogenetic tree should be labelled with species such that one is the ancestor of the other.

The following strategy is often used for reconstructing the evolutionary relationships of a set of species (i.e. their *species tree*): One begins by constructing phylogenetic trees for a set of distinct gene families (i.e. *gene trees*). Typically, these gene trees are built using one of the standard techniques for constructing phylogenetic trees from molecular sequence data. However, for many gene families, the gene tree differs from the species tree (using another terminology, their topologies disagree). Hence, a single gene tree is not considered sufficient for inferring a species tree. For this reason, a set of gene trees is often used in order to increase the reliability of the resulting species tree. We have studied this algorithmic problem, that is, given a set of disagreeing gene trees find the species tree [Hallet and Lagergren, 1999].

Algorithms in language engineering

Viggo Kann

The goal of this project is to combine algorithms that have a solid mathematical base with linguistic knowledge in order to construct tools for Swedish text processing; tools that are both accurate and capable of handling large amounts of data without speed degradation. We have earlier successfully constructed tools for Swedish hyphenation and Swedish spelling error detection and correction.

The goal of this project is to combine algorithms ... to construct tools for Swedish text processing

Currently we are focusing on Swedish grammar checking and proof reading. We are constructing Granska, a hybrid system using both statistical and linguistical methods for checking and correcting Swedish text. Much work has been devoted to constructing a part-of-speech tagger that tags each word in a sentence with its word class and inflectional features, see [Carlberger and Kann, 1999]. By attacking the problems of disambiguation of ambiguous words and tagging of unknown words we have managed to obtain a tagger that tags 97% of the words correctly.

We have also constructed and implemented an efficient and powerful artificial language to express grammatical errors and corrections. In the IPLab part of the project, many rules in this language have been written and evaluated, and a graphical user interface has been constructed. There is a web interface to Granska: http://www.nada.kth.se/theory/projects/granska/demo.html

http://www.nada.kth.se/theory/projects/granska/demo.html

Intelligent Data Analysis and Visualization

Stefan Arnborg

In a project within the Centre for Geoinformatics, Johannes Keukelaar develops methods for rough set classification using a visual language. This technique is useful, for instance, when a number of qualitatively different classifications of geographical areas are merged, see [Keukelaar, 1999] and [Ahlqvist *et al*., 1999].

Design and development of brain image databases

Per Svensson

The project is to perform research on architecture, design and usage quality aspects of large-scale brain image databases and related data analysis management systems. It combines research objectives from the subject areas of database technology and brain image analysis so as to contribute to the ECHBD image database and the BINS neuroinformatics analysis center projects (see below), investigate the application of advanced object-oriented

The project is to perform research on architecture, design and usage quality aspects of large-scale brain image database...

methodology in heterogeneous, networked scientific image analysis systems, and develop improved technology for meta-research and data mining in large-scale brain image databases.

The European Computerized Human Brain Database (ECHBD) project is supported by an EC Biotech grant and the Brain Image Neuroinformatics System (BINS) by an SSF grant. Both projects were initiated and are led by professor Per Roland at the Division of Human Brain Research of the Karolinska Institute in Stockholm.

There is no mature methodology for management of 3D spatial and spatiotemporal image databases. Therefore, one objective of the project is to evaluate and refine emerging such methodology, in particular the RasDaMan system. We will also investigate the use of mediator technology (the AMOS II system) in a large and complex scientific database environment.

The project has a neuroinformatics usage perspective involving management of very large, inhomogeneous data collections consisting mainly of 3D raw data images, logical and statistical operations and queries on sequences of such images, performance requirements arising from interactive display 3D images, and web distribution and access of images.

From this usage perspective, the project investigates multidatabase architecture and data modeling issues, data mining methods for functional PET and MRI brain imagery, use of a supercomputing center as a backend storage and computing resource, and performance issues in storage, access, query, and display of brain imagery.

A brain image database system, based on the RasDaMan raster database manager and the O2 object-oriented database management system, was developed for the ECHBD project. The first working version of the system was made available to the ECHBD project partners in September 1999.

A research group has been formed whose first task is to design the general architecture of the BINS raw image database system.

Some of the groups results can be found in [Fredriksson, 1999] and in [Fredriksson *et al.*, 1999].

Object-relational language and system for spatial analysis (AMOROSE)

Per Svensson

This project researches state-of-the-art spatial database technology to improve the interface between users and Geographical Information Systems (GIS), as well as between GIS and databases.

Current GIS's lack efficient facilities for integrating the base system and a domain model for spatial analysis. Such facilities may be provided by an object-oriented database language, whose primitives represent fundamental operations of spatial analysis.

The use of a declarative object-oriented query language for domain modelling offers several advantages over conventional imperative programming techniques. It permits clear expression of ad-hoc analysis problems on a high level of abstraction. Declarative models are easy to define, inspect and understand and lead to compact, easily reusable, and powerful domain models. Object-oriented query languages provide object views which are invoked independently of whether they represent stored or derived data. This feature supports data independence and schema evolution.

The Amorose project aims to develop a prototype spatial analysis and database management system...

The Amorose project aims to develop a prototype spatial analysis and database management system based on the AMOS II main-memory object mediator system (Risch et al., Uppsala University) with spatial extensions from the ROSE library (Gueting et al., Fernuniversitaet Hagen) , the prototype will be used to evaluate the prototype over a representative set of spatial data analysis tasks.

In the first phase of the project (1996–98), the Rose library was brought to a state where it satisfies the basic needs of the Amorose project. As a result of this work, Rose was used as the basic computational tool in a study of the concept of rough sets for uncertainty representation in spatial data classification, see [Ahlqvist *et al*., 1999].

The second phase involved the preliminary integration of Rose into Amos. This first integrated Amorose prototype showed how complex spatial queries could be formulated in AmosQL and executed within Amorose.

The ongoing third development phase aims at close integration of the modified ROSE library and the Windows NT-based Amos II mediator system. The resulting Amorose system will provide a scalable memory integration mechanism extending the Amos query optimizer and making use of query optimization techniques for efficient query evaluation.

Inference from data

Stefan Arnborg

Under this heading we study fundamental and applied inference problems. The fundamental studies are related to ongoing discussions about the role and possible adaptation of Bayesianism to applications in semi-intelligent computer systems. The applied studies concern the possibilities for advanced data mining techniques to help investigating major medical puzzles, like explanation and treatment of schizophrenia. See [Arnborg, 1999a] and Arnborg [1999b].

www.ki.se/cns/hubin/.

References–TCS

Book

[Ausiello et al., 1999] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti Spaccamela, A., and Protasi, M. (1999), *Complexity and Approximation— combinatorial optimization problems and their approximability properties,* Springer Verlag.

Theses

[Engebretsen, 1998] Engebretsen, L. (1998), *Approximating Generalizations of Max Cut*, Licentiate thesis.

[Keukelaar, 1999] Keukelaar, J. (1999), *A Visual Programming Language for the Analysis of Uncertain Spatial Data,* Licentiate thesis.

[Näslund, 1998] Näslund, M. (1998), *Bit Extraction, Hard-Core Predicates, and the Bit Security of RSA,* PhD thesis.

[Ulfberg, 1999] Ulfberg, S (1999), *On Lower Bounds for Circuits and Selection,* PhD thesis.

Journal publications

[Ahlqvist *et al*., 1999] Ahlqvist, O., Keukelaar, J. H. D., and Oukbir, K. (1999), *Rough classification and accuracy assessment, accepted to Internat*. J. Geographic Information Science.

[Amaldi and Kann, 1998] Amaldi, E. and Kann, V. (1998), *On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems*, Theoretical Comput. Sci. 209, 237–260.

[Andersson and Engebretsen, 1998] Andersson, G. and Engebretsen, L. (1998), *Better approximation algorithms and tighter analysis for Set Splitting and Not-All-Equal Sat*, Inform. Process. Lett., 65:6, 305–311.

[Berg and Ulfberg, 1998] Berg, C. and Ulfberg, S. (1998), *A lower bound for perceptrons and an oracle separation of the* PP^{PH} *hierarchy*, J. Comput. and Syst. Sci. 56, 263–271.

[Cai *et al*., 1998] Cai, L., Chen, J., and Håstad, J. (1998), *Circuit bottom fan-in and computational powe*r, SIAM J. Computing 27:2, 341–355.

[Carlberger and Kann, 1999] Carlberger, J. and Kann, V. (1999), *Implementing an efficient part-of-speech tagger*, Software Practice and Experience, 29, 815–832.

[Crescenzi and Kann, 1998] Crescenzi, P. and Kann, V. (1998), *How to find the best approximation results — a follow-up to Garey and Johnson*, SIGACT News 29:4, 90—97.

[Fernández-Baca and Lagergren, 1998] Fernández-Baca, D. and Lagergren, J. (1998), *On the approximability of the Steiner tree problem in phylogeny*, J. Discrete and Applied Math., Special issue on comp. molecular biology, 88:127–143.

[Goldmann and Håstad, 1998] Goldmann, M. and Håstad J. (1998). *Monotone Circuits for Connectivity Have Depth (log n)$^{2-o(1)}$* SIAM J. Computing, 27:5, 1283–1294.

[Goldmann and Karpinski, 1998] Goldmann, M. and Karpinski, M. (1998), *Simulating threshold circuits by majority circuits*, SIAM J. Computing, 27:1, 230–246.

[Goldreich and Håstad, 1998] Goldreich, O. and Håstad, J. (1998), *On the complexity of interactive proof with bounded communication*, Inform. Process. Lett. 67:4, 205–214.

[Håstad, 1998] Håstad, J. (1998), *The shrinkage exponent of De Morgan formulas is 2*, SIAM J. Computing, 27:1, 48–64.

[Håstad, 1999] Håstad, J. (1999), *Clique is hard to approximate within $n^{1-\varepsilon}$*, Acta Mathematica, 182, 105–142.

[Håstad *et al*., 1999] Håstad, J., Imagliazzo, R., Levin, L., and Luby, M. (1999), *A Pseudorandom Generator from any one-way function*, SIAM J. Computing, 28:4, 1364–1396.

[Kann *et al*., 1999] Kann, V., Domeij, R., Hollman J., and Tillenius M. (1999), *Implementation aspects and applications of a spelling correction algorithm,* in R. Koehler, L. Uhlirova, G. Wimmer: Text as a Linguistic Paradigm: Levels, Constituents, Constructs. Festschrift in honour of Ludek Hrebicek.

[Kann *et al*., 1998] Kann, V., Lagergren, J., and Panconesi, A. (1998), *Approximate MAX k-CUT with subgraph guarantee*, Inform. Pro cess. Lett. 65, 145–150.

[Lagergren, 1998] Lagergren, J. (1998), *Upper bounds on the size of obstructions and intertwines*. Journal of Combinatorial Theory Series B, 73:1, 7–40.

[Johansson, 1998] Johansson, Ö. (1998), *Clique-decomposition, NLC-decomposition, and modular decomposition — relationships and results for random graphs*, CGTC 1998, Congressus Numerantium, 132, 39–60.

[Johansson, 1999] Johansson, Ö. (1999), *Simple distributed Δ+1-coloring of graphs,* Inform. Process. Lett. 70:5, 229–232.

Conference publications

[Abdulla et al., 1999] Abdulla, P., Ciapessoni, E., Marmo, P., Meinke, K., and Ratto, E. (1999), *FAST: an integrated tool for verification and validation of real time system requirements*, in P.G. Larsen (ed) Proc. Fifth FMRail Workshop, electronic publication with LNCS 1708/ 1709, Proc. FM '99, Springer Verlag.

[Andersson, 1999] Andersson, G. (1999), *An approximation algorithm for Max p-section*, STACS 1999.

[Andersson and Engebretsen, 1998] Andersson, G. and Engebretsen, L. (1998), *Sampling methods applied to dense instances of non-Boolean optimization problems*, RANDOM 1998, LNCS 1518, 357–368.

[Andersson *et al*., 1999] Andersson, G., Engebretsen, L., and Håstad, J. (1999), *A new way to use semidefinite programming with applications to linear equations mod* p, SODA 1999, 41–50.

[Arnborg, 1999a] Arnborg, S. (1999a), *Learning in Prevision Space,* ISIPTA-99.

[Aumann *et al*., 1999] Aumann, Y., Håstad, J., Rabin, M., and Sudan, M. (1999), *Linear consistency testing, Proceedings of Workshop on Randomization and Approximation Techniques* in Computer Science, Berkeley CA.

[Engebretsen, 1999] Engebretsen, L. (1999), *An explicit lower bound for TSP with distances one and two*, STACS 1999.

[Fredriksson, 1999] Fredriksson, J. (1999), *Design of an Internet accessible visual human brain database system*, IEEE International Confer-ence on Multimedia Computing and Systems (ICMCS '99), Florence, Italy.

[Fredriksson *et al*., 1999] Fredriksson, J., Roland, P. and Svensson, P. (1999), *Rationale and design of the European computerized human brain database*, System, Scientific and Statistical Database Management (SSDBM '99), Cleveland.

[Goldmann and Russell, 1999] Goldmann, M. and Russell, A. (1999), *The complexity of solving equations over finite groups,* CCC 1999.

[Håstad *et al*., 1998] Håstad, J., Ivansson, L., and Lagergren, J. (1998), *Fitting points on the real line and its application to RH mapping*, ESA 1998, 465–476.

[Håstad and Näslund, 1998] Håstad, J. and Näslund, M. (1998), *The security of individual RSA bits*, FOCS 1998, 510–519.

[Johansson, 1999] Johansson, Ö., *NLC_2-decomposition in polynomial time*, WG'99.

[Näslund and Russell, 1998] Näslund, M. and Russell, A. (1998) *Extraction of optimally unbiased bits from a biased source,* IEEE ITW '98, 90–91.

Misc

[Arnborg, 1999b] Arnborg, S. (1999b), *A survey of Bayesian Data Mining –Part I: Discrete and semi-discrete Data Matrices,* SICS Technical Report.

[Berg *et al*., 1999] Berg, C., Ekström, R., and Meinke, K. (1999), *Automatic test case generation for derivative trading software*, submitted to: TEST Congress 2000.

[Goldmann et al., 1999] Goldmann, M., Näslund, M., and Russell, A. (1999), *Spectral bounds on general hard core predicates,* manuscript.

[Hallet and Lagergren, 1999] Hallet, M.T. and Lagergren, J, (1999), *New Algorithms for the Duplication- Loss Model, manuscript.*

[Ivansson and Lagergren, 1999] Ivansson, and Lagergren (1999), *Algorithms for RH Mapping: New Ideas and Improved Analysis*, manuscript.

[Meinke and Nielsen, 1999] Meinke, K. and Nielsen, J. (1999), *User requirements capture with tabular TRIO: a case study of a cruise controller,* submitted to Formal Aspects of Computing.