

# A Decade of Combinatorial Optimization

Karen Aardal<sup>1</sup>

*Utrecht University*

Stan van Hoesel

*University of Maastricht*

Jan Karel Lenstra

*Eindhoven University of Technology; CWI, Amsterdam*

Leen Stougie

*University of Amsterdam*

## Abstract

This paper offers a brief overview of the developments in combinatorial optimization during the past decade. We discuss improvements in polynomial-time algorithms for problems on graphs and networks, and review the methodological and computational progress in linear and integer optimization. Some of the more prominent software packages in these areas are mentioned. With respect to obtaining approximate solutions to NP-hard problems, we survey recent positive and negative results on polynomial-time approximability and summarize the advances in local search.

## 1 Introduction

Combinatorial optimization is involved with models and methods for optimization over discrete choices. It is rooted in the theory of linear programming, and has strong links with discrete mathematics, probability theory, algorithmic computer science, and complexity theory. Some problems in the area are relatively well understood and admit solution to optimality in polynomial time. Many others are NP-hard, and one is forced to go one of three ways. Either one chooses an enumerative method that is guaranteed to produce an optimal solution. Or one applies an approximation algorithm that runs in polynomial time. Or one resorts to some type of heuristic search technique, without any a priori guarantee in terms of solution quality or running time.

In the past decade we have seen significant progress on all these fronts. Network flow algorithms became more efficient, and so did algorithms for linear and convex optimization. For the hard problems, advances in polyhedral techniques extended the realm of true optimization methods. Performance bounds that can – or probably cannot – be met in polynomial time were tightened. And there has been a surge in the development of local search approaches. Remarkable aspects were the use of randomization in the design and analysis of algorithms, and the attention paid to on-line planning models.

---

<sup>1</sup>This research was partially supported by ESPRIT Long Term Research Project 20244 (project ALCOM IT: *Algorithms and Complexity in Information Technology*).

Two developments outside the area stimulated research in combinatorial optimization. First, the continued increase in computing power strengthened the need for efficient algorithms. The ability to handle bigger problems made the distinction between low and high order running times more pronounced and diminished the power of brute force. Second, at the application side, there has been an increasing confidence in the practical potential of optimization techniques. Large and difficult real-world problems that were out of reach ten years ago are now being solved. Notable examples occurred in airline crew scheduling, train timetabling, time-constrained vehicle routing, telecommunication network design, frequency allocation, VLSI layout synthesis, and statistical disclosure control.

Combinatorial optimization has established itself as a mature discipline of scientific interest and practical relevance. The selection of topics and references presented below has been governed by space constraints and personal bias. We apologize for all omissions.

## 2 Graphs and networks

Major improvements in running times of algorithms for specific graph related problems have been obtained during the last ten years. We highlight a few here. Quite a number of these results have been obtained via randomized algorithms, for which we refer to the book by Motwani and Raghavan [39].

Much research on designing faster algorithms for the maximum flow problem and the minimum cost flow problem was initiated by the work of Tardos [49], who found the first strongly polynomial algorithm for the minimum cost flow problem. Scaling of the input parameters and prefixing flows are the main ingredients of most of these new algorithms, but the design of efficient data structures has also had an important impact. For a network with  $n$  nodes and  $m$  arcs, the best known strongly polynomial algorithms for finding a maximum flow and a minimum cost flow have running times  $O(nm \log n)$  and  $O(m \log n(m + n \log n))$ , respectively. A thorough treatment of these results is given by Ahuja et al. [6]. This book also describes improvements in solution times for problems such as shortest path and matching problems.

The minimum cut problem of finding a minimum weight set of arcs in a network whose removal would disconnect the network is dual to the maximum flow problem. Recently, new algorithms have been developed for this problem that do not exploit this duality. Nagamochi and Ibaraki [40], for instance, use edge contraction in their algorithm. Randomized edge contraction, introduced by Karger and Stein [32], leads to the fastest algorithm so far. An overview of these algorithms with a computational study is given by Chekuri et al. [17].

Interesting results have been obtained in determining polynomially solvable subclasses of generally NP-hard problems. Robertson and Seymour [45] proved an old conjecture of Wagner: for each set of graphs that is closed under taking minors, there exists a finite set of graphs that are forbidden to be minors of any graph in the set. This *obstruction set* can be enormously large, but its finiteness allows Robertson and Seymour to prove the existence of a polynomial-time algorithm for determining the tree-width of any graph in a class that is known to contain graphs of bounded tree-width only. The proof is non-constructive, and the algorithm may involve a large

constant (of the order of the number of forbidden minors). Bodlaender [15] gives a good overview of the techniques involved together with some applications.

Next to these classical graph problems, there are several problems in planning and scheduling that can be viewed as problems on graphs. One well-studied problem of this kind is the uncapacitated lot sizing problem. In 1958 Wagner and Whitin proposed an  $O(T^2)$  dynamic programming algorithm, where  $T$  is the number of time periods in the planning horizon. It lasted more than thirty years before a better algorithm was found. In the early 1990's three groups simultaneously developed algorithms with running time  $O(T \log T)$ ; see [5], [24], [52].

Important for the implementation of graph related algorithms is the availability of software packages. The most prominent software library is LEDA, A Library of Efficient Datatypes and Algorithms, developed by Melhorn and Näher [38]. It is implemented by a C++ class library, and incorporates many efficient data structures and algorithms. LEDA is available at `ftp://ftp.mpi-sb.mpg.de` in directory `pub/LEDA`.

### 3 Linear optimization

The main developments in linear optimization have sprouted from the work of Karmarkar [33], who started a wave of research on so-called *interior point methods*. Both theoretical and practical advances were accomplished over the past ten years, and by now some interior point methods are competitive with the celebrated simplex method. An interesting overview and discussion of the use of simplex and interior point methods can be found in the *ORSA Journal on Computing* 6.1 (1994). The book by Roos et al. [46] gives a comprehensive treatment of interior point methods for linear optimization. Interior point methods have also been developed for convex optimization problems. The application of interior point type methods to semidefinite optimization has led to results that have proved particularly useful in the design of approximation algorithms for certain combinatorial optimization problems; see Section 5.

A new line of research is the development of randomized algorithms for the search of an optimal basic feasible solution. The main open question here is if there exist randomized algorithms that solve linear optimization problems in strongly polynomial expected running time. Though this question has not been resolved yet, major steps have been taken. The fastest randomized algorithm is due to Kalai [31], and has expected running time  $O(n^2m + b\sqrt{n \log n} \log m)$ , where  $n$  is the number of variables,  $m$  the number of constraints, and  $b$  a constant independent of the input. It is in essence a randomized simplex algorithm. For a review of research in this direction we refer to Chapter 9 of the book by Motwani and Raghavan [39].

With respect to deterministic simplex algorithms, many improvements in practical performance have been achieved. Many of these improvements have been implemented in the state-of-the-art software package CPLEX [21]. CPLEX also contains an interior point method.

To enhance user-friendliness of software for linear and integer optimization, modeling languages that allow for representation of variables and constraints in a set-based format are very useful. Leading computer packages for modeling are AMPL [25] and AIMMS [14].

## 4 Integer optimization

The most commonly used technique for solving (mixed) integer programs is still branch-and-bound. The quality of the available upper and lower bounds on the optimal value of the considered instance is the decisive factor for success of this tree search technique. A lower bound on the optimal value (assuming a minimization problem) is obtained from a relaxation of the integer program. In the past ten to fifteen years attention has shifted from Lagrangian relaxation to linear programming relaxation, since the latter type of relaxation can be strengthened more easily by using cutting planes. Combining cutting planes and Lagrangian relaxation usually causes convergence problems. Moreover, good LP solvers, such as CPLEX, that allow for addition of rows are nowadays available.

The theory of cutting planes in the form of valid inequalities that define facets of the convex hull of feasible solutions to an instance, was mainly developed prior to this past decade. During the past ten years, however, an enormous amount of more problem specific results have been obtained. Moreover, surprisingly large instances have been solved using a mixture of cutting plane algorithms and branch-and-bound. For recent surveys we refer to Aardal and Van Hoesel [1], [2], and to Chapter 3 of [22]. Similar developments have been attained for column generation methods, which can be viewed as dual to cutting plane techniques. For a survey we refer to Barnhart et al. [11].

A new development of the last decade is the theoretical quality analysis of cutting planes. Negative results for some classes of cutting planes have been reported by Goemans [26]. He evaluated the worst-case improvement resulting from adding several of the known classes of facets for the traveling salesman polytope to the subtour polyhedron, i.e., the set of vectors satisfying the so-called subtour elimination constraints.

Another surprising theoretical result in polyhedral combinatorics is due to Lovász and Schrijver [37], who developed an algorithm for obtaining a sequence of tighter and tighter relaxations of integer 0-1 programs. The algorithm iterates the following steps. First, each constraint of the considered problem is multiplied by each variable  $x_j$  and its complement  $1 - x_j$  ( $j = 1, \dots, n$ ). The resulting quadratic program is then linearized by replacing the nonlinear terms  $x_i x_j$  by new variables  $y_{ij}$ . This linear formulation is finally projected onto the space of the original variables. Lovász and Schrijver showed that this procedure needs to be repeated at most  $n$  times before the convex hull of feasible solutions is obtained. Balas et al. [10] showed that it is sufficient to multiply each constraint by a single variable  $x_j$  and its complement at a time.

For branch-and-bound algorithms powerful and quite flexible software packages have been developed. We mention MINTO [47] and ABACUS [50]. MINTO contains more tools such as preprocessing and generic valid inequalities, whereas ABACUS has the advantage that it is written in C++.

Apart from the further development of existing solution techniques, also two new techniques for integer optimization received much attention in the last decade. The first algorithm we mention, developed by H.W. Lenstra [35], is older than ten years, but

served as an inspiration for further developments. Lenstra’s algorithm was developed to show that the problem of determining whether the polyhedron  $K = \{x \in \mathbb{R}^n : Ax \leq b\}$  contains an integer vector  $x$ , can be solved in polynomial time if  $n$  is fixed. First, the algorithm finds a transformation  $\tau$  such that the polyhedron  $\tau K$  has a “spherical” appearance. If the basis of the lattice  $\tau\mathbb{Z}^n$  has short and near-orthogonal vectors, then the membership problem can be solved recursively by branching on a number of parallel hyperplanes. The number of such hyperplanes can be proved to be bounded by a constant depending only on  $n$ . For any lattice such a basis exists and can be found in polynomial time starting from an arbitrary basis by using *basis reduction*; see Lenstra et al. [34]. Lovász and Scarf [36] designed a “generalized” basis reduction algorithm, which works directly on the polyhedron instead of using approximations such as Lenstra does. The advantage of their method is that less information is lost, the disadvantage is that it uses considerably more computational steps. Cook et al. [20] implemented the Lovász-Scarf algorithm and solved some previously unsolved integer programming problems. Barvinok [12] generalized Lenstra’s result and proved that the number of integral points in a polyhedron can be *counted* in polynomial time if the dimension is fixed.

Another new technique, based on the theory of *Gröbner bases*, was already known in computational algebraic geometry, and was introduced for solving integer optimization problems by Conti and Traverso [19]. It amounts to translating the integer programming problem into an algebraic membership problem. The Gröbner bases are used to guide the generalized division that decides the membership. Advances in applicability of these methods are due mainly to Thomas [51]. Their current practical power is restricted by the size of the Gröbner bases, which is large for most problems. Due to their structure such methods have advantages over other more conventional IP methods in solving stochastic integer programming problems; see Schultz et al. [48]. Computer packages for computing Gröbner bases are available, e.g., CoCoa [16] and MACAULAY [13].

## 5 Polynomial-time approximation

As an alternative to solving NP-hard combinatorial optimization problems to optimality, which may be very time consuming, a stream of research has concentrated on designing polynomial-time algorithms that aim at good approximations for such problems. A widely accepted quality measure of such approximations is the *performance guarantee*, i.e., an upper bound on the ratio between the approximate solution value and the optimal one. A comprehensive and up-to-date survey of the theory of approximation algorithms is provided in the book edited by Hochbaum [28].

Some of the major achievements in this field are based on a combination of relaxation and randomization. Goemans and Williamson (see [27] and Chapter 11 of [28]) designed approximation algorithms that solve appropriately chosen relaxations of mathematical programming formulations of the considered combinatorial problems, and then round the obtained solution in a randomized way. The rounding can be derandomized yielding deterministic approximation algorithms. In particular, Goemans and Williamson use semidefinite optimization relaxations to design algorithms with very good performance guarantees for the problem of finding a

maximum cardinality cut in a graph and the problem of finding the maximum number of simultaneously satisfiable clauses in a Boolean expression with at most two literals per clause.

A remarkable result was obtained by Arora [8]. He developed a polynomial-time approximation scheme for the traveling salesman problem (TSP) in the Euclidean space. Here we notice that Christofides' algorithm of 1976 [18], with its performance guarantee of  $3/2$ , is still the best polynomial approximation algorithm for the TSP whose distances are symmetric and satisfy the triangle inequality.

Apart from the above positive sounds on approximation, there has also been a breakthrough on the negative side, in the sense of non-approximability of optimal solutions of some problems. Papadimitriou and Yannakakis [42] defined a class of maximization problems for the purpose of distinguishing problems whose optimal solutions are hard to approximate within arbitrarily small ratio. This class called MAXSNP has a two-sided polynomial reduction defined on it under which it is closed. Given a Boolean expression in conjunctive normal form, the problem MAXSAT of finding a truth assignment to the variables that satisfies the maximum number of clauses is complete for this class.

Arora et al. [9] gave a strong justification for investigating these concepts. They showed that there cannot exist a polynomial-time approximation scheme for MAXSAT unless  $P = NP$ . The proof is based on an alternative definition of NP in terms of randomized certificate verification based on fingerprinting methodology. This important result implies that for any MAXSNP-complete problem there must be a threshold value strictly greater than 1 on the achievable polynomial-time performance guarantee. For an overview of specific results in this direction, we refer to Chapter 10 of [28].

In sequencing and scheduling, techniques based on linear programming and rounding led to surprising performance guarantees for the off-line and on-line minimization of total (weighted) completion time on a single machine and on parallel machines, and for the minimization of makespan on parallel machines subject to communication delays. An investigation of the complexity of finding very short schedules yielded lower bounds on the polynomial-time approximability of several scheduling problems, including the job shop scheduling problem. For specific results and references, we refer to Chapter 12 of [22].

The previous paragraphs concerned the *worst-case* approach to approximation. A complementary approach is *average-case* or *probabilistic* analysis, a research field that started more than twenty years ago. The main developments in this field during the last decade were based on discovering the possibility to exploit existing results from probability theory. *Empirical process* theory provided tools for the analysis of the optimal solution value of a series of number problems; see Piersma [43]. *Martingale* theory allowed for relatively elegant asymptotic characterizations of optimal solution values of several problems; see Rhee and Talagrand [44]. Finally we mention the rather complete probabilistic analysis of *bin-packing* algorithms, presented in Chapter 2 of [28].

Next to these developments for optimization problems, a breakthrough in approximation was accomplished for *counting* problems, again based on randomization. Counting combinatorial structures such as the number of Hamiltonian cycles in a graph is obviously harder than just deciding on the presence of the structure. Jerrum et al.

[30] showed the equivalence between approximate counting and approximate sampling for a wide class of combinatorial structures. Building on work by Aldous [7], they use Markov chains to simulate random (uniform) sampling of the structures, and proved that these “mix rapidly”. As a first result Jerrum and Sinclair [29] devised a fully polynomial randomized approximation scheme (FPRAS) for counting perfect matchings in dense graphs, whose vertices have degree at least half of the total number of vertices. The non-dense graph case is still open. Another prominent result in this direction is an FPRAS for computing volumes of convex bodies by Dyer et al. [23]. A series of subsequent papers have given schemes with increasingly better running times. For an overview we refer to Chapter 12 of [28].

## 6 Local search

For many years heuristic search approaches have been used throughout science and engineering. Their performance was generally considered to be satisfactory, partly based on experience, partly based on a belief in some physical or biological analogy, which was not always supported by familiarity with what has been achieved in mathematics.

Still, in the past decade local search has reinforced its position as a standard approach in combinatorial optimization. Problem size or lack of analytical insight may prohibit the application of true optimization algorithms. Polynomial-time approximation algorithms may give inferior solutions, and their performance bounds, if they can be obtained at all, may be meaningless in practice. Local search is a robust way to obtain good solutions to real problems in reasonable time.

Simulated annealing has established itself as a relatively straightforward technique that performs very well when given enough time. Tabu search requires more tuning but often less running time. Genetic algorithms are not known to perform well in a pure form, even when problem solutions allow a natural string representation, but hybrid forms in which offspring are subjected to iterative improvement are promising. Neural networks have many applications, which, however, seem to fall outside the realm of optimization.

Many aspects of local search are discussed in the book edited by Aarts and Lenstra [3]. We see three main lines of advance. First, a theory of the computational complexity and performance analysis of local search is now emerging; see Chapters 2 and 3 of [3]. Second, neighborhoods embodying problem-specific knowledge and data structures supporting incremental computations are being used in rather sophisticated implementations. Third, some of the more successful search strategies are hybrids, which combine local search with a constructive method, with tree search or, again, with local search. The *shifting bottleneck procedure* for job shop scheduling of Adams et al. [4] is a constructive rule that reoptimizes partial schedules along the way. The shop scheduling algorithms of Nowicki and Smutnicki (see, e.g., [41] and Chapter 11 of [3]) apply tabu search and jump back to previously considered promising but rejected moves; many other combinations of local search and tree search have been proposed. Johnson’s *iterated Lin-Kernighan* algorithm for the TSP (see Chapter 8 of [3]) is a nested form of local search, which applies 4-exchanges to local optima resulting from variable-depth search.

## References

- [1] K. Aardal, C.P.M. van Hoesel (1996). Polyhedral techniques in combinatorial optimization I: Theory. *Statist. Neerlandica* 50, 3–26.
- [2] K. Aardal, S. van Hoesel (1995). *Polyhedral Techniques in Combinatorial Optimization II: Computations*, Report UU-CS-1995-42, Utrecht University.
- [3] E.H.L. Aarts, J.K. Lenstra (eds.) (1997). *Local Search in Combinatorial Optimization*, Wiley, Chichester.
- [4] J. Adams, E. Balas, D. Zawack (1988). The shifting bottleneck procedure for job shop scheduling. *Management Sci.* 34, 391-401.
- [5] A. Aggarwal, J.K. Park (1993). Improved algorithms for economic lot size problems. *Oper. Res.* 41, 549-571.
- [6] R.K. Ahuja, T.L. Magnanti, J.B. Orlin (1993). *Network Flows; Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs.
- [7] D.J. Aldous (1981). Random walks on finite groups and rapidly mixing Markov chains. *Séminaire de Probabilités XVII*, Lecture Notes in Mathematics 986, Springer, New York, 243–297.
- [8] S. Arora (1996). Polynomial time approximation schemes for Euclidean PST and other geometric problems. *Proc. 37th Annual Symp. Foundations of Computer Science*, 2–11.
- [9] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy (1992). Proof verification and hardness of approximation problems. *Proc. 33rd Annual Symp. Foundations of Computer Science*, 14–23.
- [10] E. Balas, S. Ceria, G. Cornuéjols (1993). A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Programming* 58, 295–324.
- [11] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance (1994). Branch-and-price: column generation for solving huge integer programs. In: J.R. Birge, K.G. Murty (eds.). *Mathematical Programming: State of the Art 1994*, University of Michigan, Ann Arbor, 186-207.
- [12] A.I. Barvinok (1994). A polynomial time algorithm for counting integral points. *Math. Oper. Res.* 19, 769–779.
- [13] D. Bayer, M. Stillman. *MACAULAY: A Computer Algebra System for Algebraic Geometry*, Available by anonymous ftp from [zariski.harvard.edu](http://zariski.harvard.edu).
- [14] J. Bisschop, R. Entriken (1993). *AIMMS: The Modeling System*, Available at <http://www.paragon.nl/aimms220.html>.
- [15] H.L. Bodlaender (1993). A tourist guide through treewidth. *Acta Cybernet.* 11, 1–21.
- [16] A. Capani, G. Niesi (1995). *CoCoa User's Manual, Release 3.0b, Edition 1995*, Department of Mathematics, University of Genova.
- [17] C.S. Chekuri, A.V. Goldberg, D.R. Karger, M.S. Levine, C. Stein (1996). *Experimental study of minimum cut algorithms*, Technical report, NEC Research Institute, Princeton.



- [18] N. Christofides (1976). *Worst-case analysis of a new heuristic for the travelling salesman problem*, Report 388, GSIA, Carnegie Mellon University.
- [19] P. Conti, C. Traverso (1991). Buchberger algorithm and integer programming. *Proceedings of AAECC 9, New Orleans*, Lecture Notes in Computer Science 539, Springer, Berlin, 130–139.
- [20] W. Cook, T. Rutherford, H.E. Scarf, D. Shallcross (1993). An implementation of the generalized basis reduction algorithm for integer programming. *ORSA J. Comput.* 5, 206–212.
- [21] CPLEX Optimization, Inc. (1994). *Using the CPLEX callable library, Version 3.0*.
- [22] M. Dell'Amico, F. Maffioli, S. Martello (eds.) (1997). *Annotated Bibliographies in Combinatorial Optimization*, Wiley, Chichester.
- [23] M.E. Dyer, A.M. Frieze, R. Kannan (1991). A random polynomial algorithm for approximating the volume of convex bodies. *J. Assoc. Comput. Mach.* 38, 1–17.
- [24] A. Federgruen, M. Tzur (1991). A simple forward algorithm to solve general dynamic lot sizing models with  $n$  periods in  $O(n \log n)$  or  $O(n)$  time. *Management Sci.* 37, 909–925.
- [25] R. Fourer, D.M. Gay, B.W. Kernighan (1993). *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press.
- [26] M.X. Goemans (1995). Worst-case comparison of valid inequalities for the TSP. *Math. Programming* 69, 335–349.
- [27] M.X. Goemans, D.P. Williamson (1994). .878-Approximation algorithms for MAX CUT and MAX 2SAT. *Proc. 26th Annual ACM Symp. Theory of Computing*, 422–431.
- [28] D.S. Hochbaum (ed.) (1996). *Approximation Algorithms for NP-Hard problems*, PWS Publishing Company, Boston.
- [29] M.R. Jerrum, A. Sinclair (1988). Approximating the permanent. *SIAM J. Comput.* 18, 1149–1178.
- [30] M.R. Jerrum, L.G. Valiant, V.V. Vazirani (1986). Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.* 43, 169–188.
- [31] G. Kalai (1992). A subexponential randomized simplex algorithm. *Proc. 24th Annual ACM Symp. Theory of Computing*, 475–482.
- [32] D.R. Karger, C. Stein (1993). An  $O(n^2)$  algorithm for minimum cuts. *Proc. 25th Annual ACM Symp. Theory of Computing*, 757–765.
- [33] A.V. Karmarkar (1984). A new polynomial-time algorithm for linear programming. *Combinatorica* 4, 373–395.
- [34] A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász (1982). Factoring polynomials with rational coefficients. *Math. Ann.* 261, 515–534
- [35] H.W. Lenstra, Jr. (1983). Integer programming with a fixed number of variables. *Math. Oper. Res.* 8, 538–548.
- [36] L. Lovász, H.E. Scarf (1992). The generalized basis reduction algorithm. *Math. Oper. Res.* 17, 751–764.

- [37] L. Lovász, A. Schrijver (1991). Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization* 1, 166–190.
- [38] K. Mehlhorn, S. Näher (to appear). *The LEDA Platform of Combinatorial and Geometric Computing*, Cambridge University Press, Cambridge.
- [39] R. Motwani, P. Raghavan (1995). *Randomized Algorithms*, Cambridge University Press, New York.
- [40] H. Nagamochi, T. Ibaraki (1992). Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math.* 5, 54–66.
- [41] E. Nowicki, C. Smutnicki (1996). A fast taboo search algorithm for the job shop problem. *Management Sci.* 42, 797–813.
- [42] C.H. Papadimitriou, M. Yannakakis (1991). Optimization, approximation, and complexity classes. *J. Comput. System Sci.* 28, 425–440.
- [43] N. Piersma (1993). *Combinatorial Optimization and Empirical Processes*, Tinbergen Institute Research Series 52, Amsterdam.
- [44] W.T. Rhee, M. Talagrand (1989). Martingale inequalities, interpolation and NP-complete problems. *Math. Oper. Res.* 14, 91–96.
- [45] N. Robertson, P.D. Seymour (1996). Graph minors XV: giant steps. *J. Combin. Theory Ser. B* 68, 112–148.
- [46] C. Roos, T. Terlaky, J.P. Vial (1997). *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, Wiley, Chichester.
- [47] M.W.P. Savelsbergh, G.C. Sigismondi, G.L. Nemhauser (1994). Functional description of MINTO, a Mixed INTEGER Optimizer. *Oper. Res. Lett.* 15, 47–58.
- [48] R. Schultz, L. Stougie, M.H. van der Vlerk (1995). *Solving stochastic programs with integer recourse by enumeration: a framework using Gröbner basis reductions*, Discussion paper TI 95-216, Tinbergen Institute, Amsterdam.
- [49] É. Tardos (1985). A strongly polynomial minimum cost circulation algorithm. *Combinatorica* 5, 247–255.
- [50] S. Thienel (1995). *ABACUS, A Branch-And-Cut System*, PhD thesis, Institut für Informatik, Universität zu Köln, Germany.
- [51] R.R. Thomas (1995). A geometric Buchberger algorithm for integer programming. *Math. Oper. Res.* 20, 864–884.
- [52] A. Wagelmans, S. van Hoesel, A. Kolen. Economic lot sizing: an  $O(n \log n)$  algorithm that runs in linear time in the Wagner-Whitin case. *Oper. Res.* 40, S145–S156.