The Knowledge Engineering Review, Vol. 15:1, 2000, 1–10
Printed in the United Kingdom

Artificial intelligence and operations research: challenges and opportunities in planning and scheduling

CARLA P. GOMES

Computer Science Department, Cornell University, Ithaca, NY 14853, USA. Email: gomes@cs.cornell.edu

1 Introduction

Both the Artificial Intelligence (AI) and the Operations Research (OR) communities are interested in developing techniques for solving hard combinatorial problems, in particular in the domain of planning and scheduling. AI approaches encompass a rich collection of knowledge representation formalisms for dealing with a wide variety of real-world problems. Some examples are constraint programming representations, logical formalisms, declarative and functional programming languages such as Prolog and Lisp, Bayesian models, rule-based formalism, etc. The downside of such rich representations is that in general they lead to intractable problems, and we therefore often cannot use such formalisms for handling realistic size problems. OR, on the other hand, has focused on more tractable representations, such as linear programming formulations. OR-based techniques have demonstrated the ability to identify optimal and locally optimal solutions for well-defined problem spaces. In general, however, OR solutions are restricted to rigid models with limited expressive power. AI techniques, on the other hand, provide richer and more flexible representations of real-world problems, supporting efficient constraint-based reasoning mechanisms as well as mixed initiative frameworks, which allow the human expertise to be in the loop. The challenge lies in providing representations that are expressive enough to describe real-world problems and at the same time guaranteeing good and fast solutions.

Below I present some of the main themes in OR followed by a discussion on several topics that, in my opinion, represent opportunities for integration of AI and OR techniques. The papers in this special issue elaborate on these topics, with an emphasis on planning and scheduling.

2 Main themes in operations research

2.1 *Optimization and linear programming*

Traditionally, the operations research community has focused on solving optimization problems. Linear programming plays a major role in OR methods. Work done by Leonid Kantorovich in 1939 is considered the main precursor to Linear Programming (LP). In 1947, George Dantzig developed LP and the simplex method, initially conceived to speed up the process of providing a time-staged deployment, training and logistical program in military applications.[1] Interestingly, the word "programming" in linear programming has nothing to do with computer programming, but rather with the notion of "program" as used by the military to refer to plans of military operations. The simplex method made it possible to consider larger problems in areas as diverse as transportation, production, resource allocation, and scheduling problems.

[1] Ironically, Dantzig was not considered for the Nobel Prize in Economics for work related to the discovery and application of LP. The prize was given to Koopmans and Kantorivich for their work applying LP to problems in economics.

The main extensions of LP are Integer Programming (IP), Mixed Integer Programming (MIP) and stochastic programming. IP and MIP extend LP to deal with integrality constraints and are the "bread and butter" of OR. Stochastic programming addresses issues dealing with uncertainty. Common approaches to solving IP and MIP entail solving several LPs, which are relaxations of the original IP or MIP that provide guidance and tighten bounds for branch and bound techniques. Similarly, Stochastic programming entails solving several LP's that represent different scenarios in the future—one can determine the best course of action in the present by optimizing the expected performance for the different scenarios.

The complexity of LP was not known for a long time. In the 1970s, Klee and Minty created an example that showed that the simplex method can require exponential time. However, despite its worst-case exponential complexity, the simplex method generally performs very well in practice. In the late 1970s, Khachian developed a polynomial-time algorithm for linear programming. On practical problems, however, this method was much less efficient than the simplex method. In 1984, Karmarkar devised an interior point method that is more efficient and can outperform the simplex method on certain large problems instances. Still, the simplex method is often the method of choice. During the announcement of the new release of CPLEX at the main OR conference, INFORMS, simplex based methods were shown to be very competitive or even outperforming interior point based methods on several benchmarks.[2]

Successful solutions of large-scale MIPs require formulations whose LP relaxations give a good approximation to feasible solutions. For instance, it is known that the Knapsack problem is relatively easy to solve if using the "right" LP formulations whose relaxations are very insightful for a branch and bound algorithm. However, some formulations of the Knapsack problem lead to poor relaxations of the corresponding LP, in the sense that they do not provide much information for a branch and bound algorithm.

The objective function is very important in OR models. In fact, in a recent review of Mathematical Programming, Dantzig (1991) emphasizes that, apart from LP and the simplex method, one of his main contributions was the formulation of an explicit *goal or objective function* to guide the search for feasible solution, instead of ad hoc ground rules. The objective function is essential in OR models, for two reasons: on the one hand, it provides a criterion for optimization and it guides the search for solutions. Furthermore, it is a way of considering soft constraints. OR experts dealing with real-world applications use the approach of encoding constraints through the objective function, avoiding the use of hard constraints as much as possible. A goal constraint is an objective that is desirable but, if necessary, it can be violated. Goal Programming involves different techniques to produce solutions involving goal constraints. An example is the use of penalties associated with variables that measure the deviation between the desired goal and the actual value.

The work of Dantzig and Wolfe on solving LP by means of *decomposition* has had a major impact on solving large-scale problems. In fact, even though the simplex method can handle sparse problems with several thousands of rows quite comfortably, it does not scale up when it comes to truly huge problems. For such problems, the simplex method is out of the question, and the Dantzig–Wolfe decomposition is needed. An example of the application of such decomposition methods is column generation techniques. They have been successfully applied, for example, in Airline Crew Scheduling (see Barnhart et al., 1994). Branch-and-price is an example of a column generation technique.

Crew scheduling is the problem of assigning crews to a given set of flights, in general all the flights of a specific fleet type. In this problem, sequences of flights (pairings) are assigned to crews so that each flight is assigned to exactly one crew. Since pairings are subject to complicated rules (safety and contractual rules) it would be difficult to express constraints and costs if a direct encoding were used.[3] Instead, valid pairings are enumerated and the problem is formulated as a Set Partioning

[2] CPLEX presentation of the new release at Informs, 1998.

[3] By direct encoding we mean a formulation with variables x_{ij}, where $x_{ij} = 1$ if crew i is assigned to flight j.

Problem (SPP). In this formulation, each column or variable corresponds to a pairing and the objective is to partition all of the flights into a set of minimum cost pairings.

The main drawback of such an approach is that the number of pairings grows exponentially with the number of flights. For example, Vance (1993) found more than five million valid pairings in a daily problem with 253 flights. Problems with 1000 flights, a typical size for a U.S. domestic carrier, are likely to have billions of pairings.

The approach used to solve this formidable problem uses Dantzig–Wolfe decomposition. The LP relaxation of the SPP is solved, but only a subset of columns are initially considered. This problem is called the *restricted master problem*. New columns are generated only as needed, and if needed, and based on the information provided by the solution to the *restricted master problem*, i.e., the *dual prices*. These *dual prices* allow one to determine which flights should be included in ''good'' columns for the master problem. The problem of generating new columns is called the *subproblem* or *pricing problem*.

2.2 Duality

Duality plays an important role in OR. The theory of duality is elegantly developed in the context of LP. The basic idea is that every problem can be considered from a dual perspective—maximizing the profit is equivalent to minimizing costs. Every maximization LP problem gives rise to a minimization LP problem, its *dual*. Interestingly, every feasible solution of one problem provides a bound on the optimal value of the other problem, and if one of them has an optimal solution, so does the other and their optimal values are the same. This is what the famous Duality Theorem states, formally proved by Gale et al. (1951). Its notions originated in conversations between Dantzig and von Neumann in the fall of 1947.

The theory of duality is also used to perform sensitivity analysis and parametric analysis, i.e., the study of the impact on the objective function when the level of resources (the right-hand sides of the linear constraints) vary or when the coefficients of the objective function vary. The technique of *penalties* uses sensitivity analysis to tighten bounds during branch-and-bound search.

2.3 Structure

Another theme in OR is to exploit inherent problem structure. *Trans-Shipment Problems* or *Network Flow Problems* are notable examples of the importance of exploiting structure. The special structure of these problems allows for very efficient (polynomial) algorithms. An interesting aspect of Network Flow Problems is that the optimal solution of instances involving only integral constraints are guaranteed to be also integer-valued. Many combinatorial problems, well beyond cases that deal with physical shipments of some commodity, such as scheduling problems, can be efficiently formulated as Network Flow Problems.

Typically, when using OR methods, one starts by categorizing the problem into a class of problems for which efficient solution methods have been developed such as LP or Network Flow. If the problem does not fit into such a class, one uses a more general formulation such as IP or MIP. At a second level, in general using an automated process, structure is detected using inference methods. For example, when solving IP's or MIP's, the derivation of ''cutting planes'' is very important to eliminate parts of the search space that are guaranteed not to contain the optimal solution. Cutting planes are linear inequalities that can be added to the original formulation of an IP with the guarantee that no integer solution will be eliminated, but with the advantage of eliminating fractional solutions generated by the linear relaxation of the problem. The addition of cutting planes leads to tighter relaxations, and therefore their solutions are better approximations of the IP's solution. Gomory (1963) pioneered this approach, showing how to systematically generate ''cuts'' that lead to an integer solution.

Related to the approach of exploiting structure is the strategy of decomposing complex problems into simpler problems for which there are good algorithmic approaches or, at least, relaxed versions of the subproblems can be solved using efficient algorithms. Network Flow Problems play an important role in decomposition strategies since they represent a large class of problems after abstracting away some "details".[4]

Decomposition is often used to get tighter bounds for branch-and-bound methods. Such an approach is used, for example, by one of the fastest jobshop scheduling algorithms (Carlier and Pinson, 1990). This algorithm bounds its search with Jackson's preemptive schedule algorithm for a single machine. Another example is the Knapsack problem. Even though this problem is NP-complete, it is relatively easy to solve in practice. It is used in several approaches, for example to solve the generalized machine assignment problem.

3 Opportunities for integration of AI/OR

Solving large real-world scheduling problems has so far been almost exclusively the domain of operations research, but recent developments in constraint-satisfaction techniques have shown that they can be competitive on real-world problems. The constraint-satisfaction approach brings a novel perspective to planning and scheduling. Constraint-based methods provide a richer representational formalism compared to the traditional OR methods. Furthermore, constraint satisfaction techniques have developed powerful inference methods that lead to efficient variable domain reductions.

For example, the constraint programming language ILOG is now being used in actual fielded applications, in areas such as manpower and service scheduling, airline scheduling, cutting-stock in the steel industry, manufacturing scheduling for the auto industry, supply chain management, etc. Companies such as SAP, Peoplesoft, and I2, leading developers of software solutions for managing human resources, accounting, materials management, distribution, and manufacturing, across different industries, combine different optimization techniques such as constraint programming, mathematical programming, and local search methods. These new developments have created a unique opportunity to investigate the integration of AI, primarily constraint-satisfaction methods, and OR techniques. Some key issues are outlined in the following paragraphs.

3.1 Hybrid solvers

This is an important emerging area of research combining CSP techniques with OR techniques. Work in this area began with CLP(R), Prolog III and Chip, combining constraint satisfaction (CSP) methods with linear programming. The ILOG system integrates a finite domain propagation solver for discrete variables with CPLEX, for continuous variables. Promising results have been obtained using such *hybrid* approaches, which allow for more powerful constraint reasoning: consistency checking and domain reduction techniques enforce efficient constraint propagation, while linear programming relaxations provide infeasibility, or bounds on the objective function. For example, a research team at Imperial College reports that *only* by using a hybrid approach were they able to solve to optimality hoist scheduling problems (Rodosek and Wallace, 1998). These problems could not be solved optimally in the OR literature. McAloon et al. (1998) also report similar benefits of using hybrid solvers to solve a multicommodity integer network flow problem of the Dutch Railways which is greatly complicated by additional constraints on the coupling and decoupling of trains. See also the paper by Hooker et al. in this issue.

3.2 Duality

The notion of duality expresses the fact that there are two complementary ways of looking at a

[4] Unfortunately, Network Flow Algorithms cannot be used when there are global constraints on the nodes of the network. An example of a global constraint would state that the amount of goods shipped through certain nodes corresponds to 30% of the total amount of goods shipped.

problem. Duality is a powerful concept that has been extensively exploited by the OR community in linear programming. Duality can be exploited to solve problems, by considering simultaneously two perspectives—the primal and dual view of the problem. Such approaches, in general, allow for stronger inferences, both in terms of cutting planes as well as variable domain reductions. Recently there have been several promising results in the CSP community using a dual formulation approach (e.g., to solve hard timetabling problems; see McAloon et al., 1997; Gomes et al., 1998). However, in general, duality is not yet well understood for problems involving constraints other than inequality constraints. Research in this area, coupled with the study of the design of global constraints and good relaxation schemes for primal and dual formulations, is very promising. The study of new ways for performing sensitivity analysis based on duality is also a promising research area. See also the paper by Hooker et al. in this issue.

3.3 Problem structure

In general, structured models are easier to understand and compute with. The OR community has identified several classes of problems with a very interesting, *tractable*, structure. LP and network flow problems are good examples of such problems. OR also exploits the structure of problems during inference by generating "cutting planes", which allow for tighter relaxations that are therefore closer to the optimal integer solution. The CSP community, on the other hand, has identified the special structure of several global constraints that are ubiquitous in several problems, which allow for the development of efficient constraint propagation techniques for the reduction of the variable domains.

In general, however, the notion of structure is very hard to define, even though we recognize structure when we see it. For example, there is not a methodology that shows how to construct good cutting planes. Formalizing the notion of structure and understanding its impact in terms of search is a big challenge for both AI and OR. AI has made some progresses in this area, namely in the study of phase transition phenomena, correlating structural features of problems. For example, in the Satisfiability problem it is known that the difficulty of problems depends upon the ratio between number of clauses and number of variables (Kirkpatrick and Selman, 1994). Recently, it has also been shown that random satisfiability instances that are a mixture of 2-Sat and 3-Sat clauses, the 3-Sat clauses with weight p, scale linearly as long as $p <= 0.4$. Another structural feature that has been recently formalized is the concept of *backbone*. The backbone of an instance corresponds to the shared structure of all the solutions of a problem instance. In other words, the set of variables and corresponding assignments that are common in all the solutions of a problem instance is the backbone (Monasson et al., 1999). For the study of phase transition phenomena and backbone in a structured domain see, for example, Achlioptas et al. (2000) and Gomes and Selman (1997).

As a final remark, it is important to mention the trade-off between highly structured models, which tend to be very specific and therefore fit a narrow class of problems, and more unstructured models that are more flexible, and therefore easier to fit real world problems.

3.4 Local search

Local search methods or meta-heuristics are often used to solve challenging combinatorial problems. Such methods start with an initial solution, not necessarily feasible, and improve upon it by performing small "local" changes. One of the earliest applications of local search was to find good solutions for the Traveling Salesman Problem (TSP) (Lin, 1965; Lin and Kernighan, 1973). Lin and Kernighan showed that by performing successive swaps of cities to an arbitrary initial tour of cities, until no such swaps are possible, one can generate solutions that are surprisingly close to the shortest possible tour. There are several ways of implementing local search methods, depending on the choice of the initial solution, types of "local" changes allowed, and feasibility and cost of (intermediate) solutions.

There is a great deal of overlap in research on local search by the AI and OR communities, namely in simulated annealing (Kirkpatrick et al., 1983), tabu search (Glover, 1989) and genetic algorithms (Holland, 1975). A recent new area of application for local search methods is in solving NP-complete *decision problems*, such as the Boolean satisfiability (SAT) problem. In 1992, Selman et al. showed that a greedy local search method, called GSAT, could solve instances with up to 700 hundred variables. Currently, GSAT and variants (e.g., WALKSAT) are among the best methods for SAT, enabling us to solve instances with up to 3000 variables (Selman et al., 1994). Closely related work in the area of scheduling is the technique of "MinConflicts" proposed by Minton et al. (1992).

Local search, and mixtures of local and global search strategies have proved to be very effective to tackle real world problems, in general beyond the reach of pure complete search methods.

3.5 Randomization

Stochastic strategies have been very successful in the area of local search. However, local search procedures are inherently incomplete methods. An emerging area of research is the study of Las Vegas algorithms, i.e., randomized algorithms that always return a model satisfying the constraints of the search problem or prove that no such model exists (Motwani and Raghavan, 1995). The running time of a Las Vegas style algorithm can vary dramatically on the same problem instance. The extreme variance or "unpredictability" in the running time of complete search procedures can often be explained by the phenomenon of "heavy-tailed cost distributions". The understanding of these characteristics explains why "rapid restarts" and portfolio strategies are very effective. Such strategies eliminate the heavy-tailed behavior and exploit *any significant probability mass early on in the distribution*. Restarts and portfolio strategies therefore reduce the variance in runtime and the probability of failure of the search procedures, resulting in more robust overall search methods (Frost et al., 1997; Gomes and Selman, 1999; Gomes et al., 1998, 2000; Hoos, 1999).

3.6 Cutting planes and constraint propagation

OR's inference method of choice, during search, is "cuts". "Cuts" or "cutting planes" are *redundant* constraints, in the sense that they do not eliminate feasible solutions. However, although these constraints are redundant in terms of the solution, they can play a major role during the search process. A classical example of the importance of cutting planes involves the pigeonhole problem: by adding the appropriate redundant constraints to a linear programming formulation, its relaxation immediately returns infeasibility. Without such redundant constraints, the results of the LP relaxation are useless. Gomory (1963) pioneered the study of cutting planes, showing how to systematically generate "cuts" that lead to an integer solution. The OR community has developed several techniques for the generation of cuts, but in general, it is not clear how to construct such cuts.

The CSP's community, on the other hand, mainly relies on domain reduction techniques for inference during search. A very successful strategy is to exploit the structure of special constraints and treat them as a global constraint (Beldiceanu and Contejean, 1994; Caseau and Laburthe, 1997; Regin, 1994, 1996). Some examples of such propagation methods are the constraint that guarantees that all elements of a vector are different (all-different constraint) and the constraint that enforces that certain values occur a given number of times in a given vector of variables (cardinality constraint). The implementation of such constraints is an interesting use of Network Flow algorithms (Régin, 1994, 1996).

A direction of research is the study of techniques that will lead to the generation of better cuts as well as efficient domain reduction techniques, and the combination of cuts with domain reduction techniques. Relevant work in this area is that of Lovasz and Schrijver (1991) and Balas, Ceria and Cornuejols (1993). They have developed the *lift-and-project* technique. Hooker (1992) has developed cutting plane algorithms for IP and resolution methods in propositional logic. Work on the automated generation of cutting planes for problems such as the pigeonhole problem has been done by Barth (1996). The work done at Kestrel Institute using a transformational approach to scheduling encompasses the generation of very efficient constraint propagation techniques (Smith and Parra, 1993). Relevant work on exploiting the structure of global constraints for domain reduction is that of, for example, Beldiceanu and Contejean (1994), Caseau and Laburthe (1997) and Régin (1994, 1996). (See also the papers by Hooker et al. and Dixon and Ginsberg in this issue.)

3.7 Coupling of column generation with CSP

As described above, the column generation formulation involves two phases: (1) generating columns, and (2) solving the corresponding LP relaxation problem. In general, the process of generating columns is quite ‘‘messy’’, since complicated constraints are involved. OR methods are not suitable for such a task. A combination of AI and OR techniques can enhance this phase considerably. Leconte et al. (1997) have reported very good results for solving a column generation problem applied to bin-packing configuration problems using hybrid solvers.

3.8 Robustness

Ideally, we would like to find not only *good* but also *robust* solutions. The intuition behind robustness is: given a set C of changes to the initial formulation of the problem instance, a solution A is more robust than solution B, w.r.t. set C, if the number of changes required to fix solution A is less than the number of changes required to fix solution B. There are very few results on the study of robustness. Most results emphasize generation of solutions from scratch completely ignoring issues on robustness. This is an area that requires substantial research, starting with a good definition of the the notion of robustness.

4 The papers in this special issue

The purpose of this special issue is to compare and contrast the different techniques from artificial intelligence and Operations Research (OR) for planning and scheduling, highlighting potential synergistic benefits from combining the techniques. We asked the contributors to this special issue to be visionary and articulate their own perspectives on the current state-of-the-art and future developments of the integration of AI and OR.

The first paper of the issue, by Hooker et al., proposes a unifying framework for optimization and constraint satisfaction techniques, based on exploiting the duality of search vs. inference and the duality of strengthening vs. relaxation.

The paper by Dixon and Ginsberg introduces the AI community to the pseudo-Boolean representations and the cutting plane proof system from OR, and the OR community to restricted learning methods, such as relevance-bounded learning. The paper proposes a new cutting plane proof for the pigeonhole principle of size n^2, and show how to implement intelligent backtracking techniques using pseudo-Boolean representation.

The remaining papers in this issue focus on planning and scheduling. Smith et al. give an overview of AI planning and scheduling methods, focusing on their similarities, differences, and limitations. Vossen et al. introduce several Integer Linear Programming (ILP) formulations of AI planning problems and compare their computational properties. Kautz and Walser describe ILP-PLAN, a framework for solving AI planning problems represented as ILP. ILP-PLAN extends the planning satisfiability approach to handle resources, action costs, and complex objective functions.

5 Forthcoming issue

We are currently in preparation of a second special issue on the integration of AI and OR techniques. The issue will feature the following articles: Ismael de Farias and George Nemhauser, on branch-and-cut for combinatorial optimization without auxilliary binary variables, Pascal van Hentenryck and Bart Selman, on automated synthesis of local search techniques, Jean-François

Puget and Irv Lustig, on combining constraint programming with mathematical programming techniques, Doug Smith and Stephen Westfold, on transformational approaches applied to the synthesis of fast schedulers, and Steven Wolfman and Daniel Weld, on combining linear programming and satisfiability solving for resource planning.

Acknowledgements

This work is funded by the Air Force Research Laboratory, Information Directorate and the Air Force Office of Scientific Research, under the New World Vistas Initiative (F30602-99-1-0005, F30602-99-1-0006, and AFOSR NWV project 2304, LIRL 97RL005N25).

References

Achlioptas, D, Gomes, C, Kautz, H and Selman, B, 2000. "Generating satisfiable instances" *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00)*.

Alt, H, Guibas, L, Mehlhorn, K, Karp, R and Wigderson A, 1996. "A method for obtaining randomized algorithms with small tail probabilities" *Algorithmica* **16** 543–547.

Balas, E, Ceria, S and Cornuejols, G, 1993. "A lift and project cutting plane algorithm for mixed 0-1 programs" *Mathematical Programming* **58** 295–324.

Barnhart, C, Johnson, E, Nemhauser, G, Savelsbergh, M and Vance, P, 1994. "Branch-and-Price: Column generation for solving huge integer programs" In: J Birge and K Murty (eds.), *Mathematical Programming. State of the Art*, 186–207.

Barth, P, 1996. *Logic based 0-1 Constraint Programming*. Kluwer.

Beldiceanu, N and Contejean, E, 1994. "Introducing global constraints in CHIP" *Mathl. Comput. Modelling* **20**(12) 97–123.

Carlier, J and Pinson, E, 1990. "A practical use of Jackson's preemptive schedule for solving the job shop problem" *Annals of Operations Research* **26** 269–287.

Caseau, Y and Laburthe, F, 1997. "Solving various weighted matching problems with constraints" *Principles and Practice of Constraint Programming* **1330**. *Lecture Notes in Computer Science* Springer-Verlag, 17–31.

Caseau, Y, Laburthe, F, Le Pape, C and Rottembourg, B, 2000. "Combining local and global search in a constraint programming environment" *The Knowledge Engineering Review* **15**.

Clements, D, Crawford J, Joslin, D, Nemhauser, G, Puttlitz, M and Savelsbergh, M, 1997. "Heuristic optimization: a Hybrid AI/OR approach" *Workshop of Constraint Programming* Austria.

Dantzig, GB, 1991. "Linear programming" In: J Lenstra, R Kan, A Schrijver (eds.), *History of Mathematical Programming, A collection of Reminiscences*. CWI, Amsterdam, 19–31.

Dantzig, GB and Wolfe, P, 1960. "Decomposition principle for linear programs" *Operations Research* **8** 101–111.

Dixon, H and Ginsberg, M, 2000. "Combining satisfiability techniques from AI and OR" *The Knowledge Engineering Review* **15**(1).

Frost, D, Rish, I and Vila, L, 1997. "Summarizing CSP hardness with continuous probability distributions" *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*.

Gale, D, Kuhn H and Tucker, A, 1951. "Linear programming and the theory of games" *Activity Analysis of Production and Allocation*. Wiley, New York, 317–329.

Glover, F, 1989. "Tabu search – part I" *ORSA Journal on Computing*, **1**(3) 190–206.

Gomes, CP, Kautz, H and Selman, B, 1998. "Boosting combinatorial search through randomization" *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*.

Gomes, CP and Selman, B, 1997. "Problem structure in the presence of perturbations" *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)* New Providence, RI, 221–226.

Gomes, CP and Selman, B, 1999. "Search strategies for hybrid search spaces" *Proceedings of the Eleventh International Conference on Tools with Artificial Intelligence (ICTAI-99)*.

Gomes, CP, Selman, B, Crato, N and Kautz, H, 2000. "Heavy-tailed phenomena in satisfiability and constraint satisfaction problems" *Journal of Automated Reasoning* to appear.

Gomes, CP, Selman, B, McAloon, K and Tretkoff C, 1998. "Randomization in backtrack search: Exploiting heavy-tailed profiles for solving hard scheduling problems" *Proc. AIPS-98*.

Gomory, R, 1958. "An outline of algorithm for integer solutions to linear programs" *Bulletin of the American Mathematical Society* **64** 275–278.

Gomory, R, 1963. "An algorithm for integer solutions to linear programs" In: R Graves and P Wolfe (eds.), *Recent Advances in Mathematical Programming*. McGraw-Hill, 260–302.

Holland, JH, 1992. *Adaptation in natural and artificial systems*. University of Michigan Press.

Hooker, J, 1992. "Generalized resolution for 0-1 linear inequalities" *Annals of Mathematics and Artificial Intelligence* **6** 271–286.

Hoos, H, 1999. "On the run-time behaviour of stochastic local search algorithms for SAT" *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-99)* 661–666.

Hooker, J, Ottosson, G, Thorsteinsson, E and Kim, H, 2000. "A scheme for unifying optimization and constraint satisfaction methods" *The Knowledge Engineering Review* **15**(1).

Jeroslow, R, 1980. "A cutting plane game for facial disjunctive programs" *SIAM J. Control and Optimization* **18** 264–280.

Kantorovich, V, 1939. "Mathematical methods in the organization of planning of production" *Management Science* **6** 366–422 [English translation].

Karmarkar, N, 1984. "A new polynomial time algorithm for linear programming" *Combinatorica* **4** 373–395.

Kautz, H and Walser, J, 2000. "Integer optimization models of AI planning problems" *The Knowledge Engineering Review* **15**(1).

Khachian, V, 1979. "A polynomial time algorithm for linear programming" *Math. Doklady* **20** 191–194. [English translation].

Kirkpatrick, S, Gelatt, CD and Vecchi, MP, 1983. "Optimization by simulated annealing" *Science* **220** 671–680.

Kirkpatrick, S and Selman, B, 1994. "Critical behavior in the satisfiability of random boolean expressions" *Science* **264** 1297–1301.

Klee, V and Minty, G, 1972. "How good is the simplex algorithm?" In: O Shisha (ed.), *Inequalities-III*. Academic Press, New York, 159–175.

Leconte, M, Couronne, P and Vergamini, D, 1996. "Column generation using cooperating constraint-based solvers" *Manuscript*.

Lin, S, 1965. "Computer solutions of the traveling salesman problem" *BSTJ* **44**(10) 2245–2269.

Lin, S and Kernighan, BW, 1973. "An effective heuristic for the traveling-salesman problem" *Oper. Res.* **21** 498–516.

Lovasz, L and Schrijver, A, 1991. "Cones of matrices and set functions and 0-1 optimizations" *SIAM J. Control and Optimization* 166–190.

Luby, M, Sinclair, A and Zuckerman, D, 1993. "Optimal speedup of Las Vegas algorithms" *Information Process. Lett.* **17** 173–180.

McAloon, K, Tretkoff, C and Wetzel, G, 1997. "Sports league scheduling" *Proceedings of Third Ilog International Users Meeting*.

McAloon, K, Tretkoff, C and Wetzel, G, 1998. "Disjunctive programming and and cooperating solvers" In: D Woodruff (ed.), *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*. Kluwer, 75–96.

Minton, S, Johnston, M, Philips, AB and Laird, P, 1992. "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems" *Artificial Intelligence* **58** 161–205.

Monasson, R, Zecchina, R, Kirkpatrick, S, Selman, B and Troyansky, L, 1996. "Determining computational complexity from characteristic 'phase transitions' " *Nature* **400**(8).

Motwani, R and Raghavan, P, 1995. *Randomized Algorithms*. Cambridge University Press.

Nemhauser, G and Wolsey, L, 1988. *Integer and Combinatorial Optimization*. Wiley, New York.

Nemhauser, G and Trick, M, 1997. "Scheduling a major college basketball conference" Georgia Tech., Technical Report.

Oddi, A and Smith, S, 1997. "Stochastic procedures for generating feasible schedules" *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)* New Providence, RI.

Puget, J-F and Leconte, M, 1995. "Beyond the Black Box: Constraints as objects" *Proceedings of ILPS'95* MIT Press, 513–527.

Régin, JC, 1994. "A filtering algorithm for constraints of difference in CSPs" *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-94)* Seattle, WA.

Régin, JC, 1996. "Generalized arc consistency for global cardinality constraint" *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)* Oregon.

Rodosek, R and Wallace, M, 1998. "One model and different solvers for hoist scheduling problems" in preparation.

Selman, B, Kautz, H and Cohen, B, 1994. "Noise strategies for improving local search" *Proc. AAAI-94* Seattle, WA, 337–343.

Selman, B, Levesque, HJ and Mitchell, D, 1992. "A new method for solving hard satisfiability problems" *Proc. AAAI-92* San Jose, CA, 440–446.

Smith, D, Frank, J and Jónsson, A, 2000. "Bridging the gap between planning and scheduling" *The Knowledge Engineering Review* **15**(1).

Smith, D and Parra, E, 1993. "Transformational approach to transportation scheduling" *Proceedings of the Eighth Knowledge-Based Software Engineering Conference* Chicago, IL.

Vance, P, 1993. "Crew Scheduling, Cutting Stock, and Column Generation: Solving Huge Integer Programs" Georgia Tech., PhD Thesis.

Vossen, T, Ball, M, Lotem, A and Nau, D, 2000. "Applying integer programming techniques to AI planning" *The Knowledge Engineering Review* **15**(1).