Problem Structure in the Presence of Perturbations

Carla P. Gomes
Rome Laboratory*
25 Brooks Rd.
Rome Lab, NY 13441-4505
gomes@ai.rl.af.mil

Bart Selman
AT&T Laboratories
Murray Hill, NJ 07974
selman@research.att.com

Abstract

Recent progress on search and reasoning procedures has been driven by experimentation on computationally hard problem instances. Hard random problem distributions are an important source of such instances. Challenge problems from the area of finite algebra have also stimulated research on search and reasoning procedures. Nevertheless, the relation of such problems to practical applications is somewhat unclear. Realistic problem instances clearly have more structure than the random problem instances, but, on the other hand, they are not as regular as the structured mathematical problems. We propose a new benchmark domain that bridges the gap between the purely random instances and the highly structured problems, by introducing perturbations into a structured domain. We will show how to obtain interesting search problems in this manner, and how such problems can be used to study the robustness of search control mechanisms. Our experiments demonstrate that the performance of search strategies designed to mimic direct constructive methods degrade surprisingly quickly in the presence of even minor perturbations.

Introduction

In recent years, we have seen significant progress in the area of search and constraint satisfaction. Using random instance distributions, hard search problems have been identified. Such instances have pushed the development of new search methods both in terms of systematic and stochastic procedures (Hogg et al. 1996). The study of highly structured problems, such as those from various finite algebra domains, has also driven the development of search procedures. For example, the question of the existence and non-existence of certain discrete structures with intricate mathematical properties gives rise to some of the most challenging search problems (Fujita et al. 1993).

An important question is to what extent real-world search and reasoning tasks are represented by such problems. It seems clear that random instances lack certain structure that is often present in realistic problems. On the other hand, the highly structured mathematical problems contain too much structure from the perspective of realistic applications.

We propose a new benchmark domain that bridges the gap between the purely random problems and the highly structured problems.[1] We consider structured problems from the area of combinatorics for which direct constructive solution methods are known. Our goal is to study the properties of such structured domains in the presence of perturbations to their structure. The properties of the resulting problems are closer to those of real-world problem instances, in the sense that the underlying structure of practical problem instances is also often perturbed by an element of randomness or uncertainty.

As our starting point, we selected the quasigroup, a discrete structure that can be characterized by a set of simple properties. As we will see, finding basic quasigroups is a relatively easy problem, for which direct constructions are known. However, the nature of the problem changes dramatically if we impose additional constraints that are locally consistent but not necessarily globally consistent. In particular, we *perturb* the structure of the quasigroup by requiring that it satisfies an incomplete initial pattern. Even though this initial pattern is consistent with the properties of quasigroups, there are no guarantees that a complete quasigroup can be derived from it.

The quasigroup completion problem enables us to study the impact of perturbations on the complexity of the underlying well-structured problem. Our experiments will show that for general search procedures small perturbations actually facilitate the search for solutions. However, when the perturbations become larger, the search problems become exponentially hard on average.

Using this spectrum of difficulty of the quasigroup completion problem, we study the performance of various forms of search control. In particular, we investigate the robustness of search procedures, i.e., whether procedures that work well on the structured problems

*Carla P. Gomes works for Rome Laboratory as a Research Associate.

[1] See also http://www.ai.rl.af.mil/quasi/quasi.html.

degrade *gracefully* in the presence of perturbation. We find that the performance of a search heuristic that mimics a direct constructive method for quasigroups degrades rapidly in the presence of even the smallest possible perturbations. On the other hand, more generic heuristics, which do not perform so well on the original problem, degrade more gracefully in the presence of perturbations, and quickly outperform the more specialized search method.

This finding suggests that it can be counterproductive to design search control strategies that specifically mimic constructive methods. Of course, specialized search control has been shown to be quite effective in certain domains. Our results merely question the robustness of such customized search methods. Apparently, even small perturbations can "throw off" the specialized search control, making it inferior to a more general method. These results are consistent with the empirical finding that simple generic search methods often outperform more sophisticated ones when applied to a range of problem instances.

The paper is structured as follows. In the next section, we introduce quasigroups and define the quasigroup completion problem. We also discuss the theoretical complexity of the problem. We then present empirical results on the quasigroup completion task, followed by the evaluation of search strategies and their scaling properties. And, finally, we summarize our results and discuss future directions.

The Quasigroup Completion Problem

A quasigroup is an ordered pair $(Q, \cdot)$, where Q is a set and $(\cdot)$ is a binary operation on Q such that the equations $a \cdot x = b$ and $y \cdot a = b$ are uniquely solvable for every pair of elements a, b in Q. The *order* N of the quasigroup is the cardinality of the set Q. A good way to understand the structure of a quasigroup is to consider the N by N multiplication table as defined by its binary operation. (For each pair of elements x and y, the table gives the result of $x \cdot y$.) The constraints on a quasigroup are such that its multiplication table defines a *Latin square*. This means that in each row of the table, each element of the set Q occurs exactly once; similarly, in each column, each element occurs exactly once (Denes and Keedwell 1974).

An *incomplete* or *partial latin square* P is a partially filled N by N table such that no symbol occurs twice in a row or a column. The *Quasigroup Completion Problem* is the problem of determining whether the remaining entries of the table can be filled in such a way that we obtain a complete latin square, that is, a full multiplication table of a quasigroup. We view the pre-assigned values of the latin square as a *perturbation* to the original problem of finding an arbitrary latin square. As we will discuss below, there are direct constructive methods for generating a latin square of any order. However, the situation is quite different for completing a partial latin square.

Evans (1960) conjectured that every N by N partial latin square with at most $N-1$ cells occupied can be completed to a latin square of order N. This is known as the *Evans conjecture*. Despite the fact that the problem received much attention, and many partial solutions were published, it took until 1981 for the conjecture to be proved correct (Smetaniuk 1981).

Andersen and Hilton (1983), through independent work, proved Evans conjecture with stronger results. They give a complete characterization of those partial latin squares of order N with N non-empty cells that *cannot* be completed to a full latin square. However, it appears unlikely that one can characterize non-completable partial latin squares with an *arbitrary* number of pre-assigned elements. This is because the completion problem has been shown to be NP-complete (Colbourn 1983, 1984). Of course, this makes the problem computationally interesting from the perspective of search and constraint satisfaction.

An interesting application area of latin squares is the design of statistical experiments. The purpose of latin squares is to eliminate the effect of certain systematic dependency among the data (Denes and Keedwell 1974). Another interesting application is in scheduling and timetabling. For example, latin squares are useful in determining intricate schedules involving pairwise meetings among the members of a group (Anderson 1985). A natural perturbation of this problem is the problem of completing a schedule given a set of pre-assigned meetings.

Quasigroups with additional mathematical properties are studied in the area of automated theorem proving (Fujita et al. 1993; Lam et al. 1989). One interesting question is to what extent special search heuristic that can guide the search to find general unrestricted quasigroups is also of use in finding special quasigroup with interesting mathematical properties.

Finally, the notion of completing partial solutions is useful in exploring the total space of solutions, since the completion problem provides insights into the density of solutions. For example, easy completion of partial structures does suggest a high density of solutions.

Computational Results for the Completion Problem

We now consider the practical computational difficulty of the quasigroup completion problem. There is a natural formulation of the problem as a Constraint Satisfaction Problem. We have a variable for each of the N^2 entries in the multiplication table of the quasigroup, and we use constraints to capture the requirement of having no repeated values in any row or column. All variables have the same domain, namely the set of elements Q of the quasigroup. Pre-assigned values are captured by fixing the value of some of the variables.

We encoded this problem in C++ using the ILOG SOLVER, a powerful C++ constraint programming library (Puget 1994). ILOG's underlying backtracking mechanism allows us to keep track of variables and their domains explored at each decision choice point while maintaining arc-consistency.

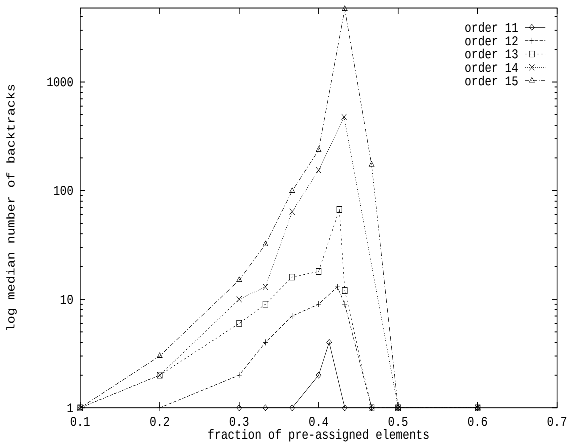

Figure 1: The Complexity of Quasigroup Completion (Log Scale)

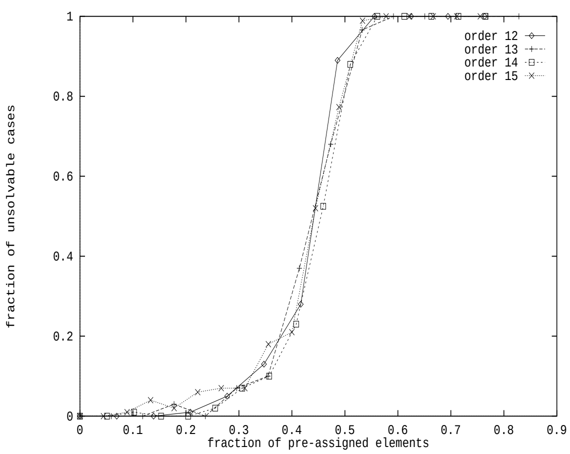

Figure 2: Phase Transition for the Completion Problem

A natural question to consider is how the difficulty of the quasigroup completion problem depends on the number of pre-assigned values. Figure 1 gives the median number of backtracks needed to find a completion or to show none exists (log-scale). (We used the specialized control heuristic described below.) Along the horizontal axis we give the fraction of pre-assigned values (out of a total of N^2 values, where N is the order of the quasigroup). From the figure, we observe that the costs peak roughly around the same ratio (approximately 42% pre-assignment) for different values of N.

What we observe here is a clear phase-transition in the problem domain. Figure 2 further confirms this. It plots the fraction of unsolvable cases as a function of the fraction of pre-assigned elements in the quasigroup. We see that the transition — from almost all instances solvable to almost all unsolvable — occurs around the same ratio as the peak in the computational difficulty. (Each data point is generated using 1,000 problem instances. The pre-assigned values were randomly generated.)

The median cost curves in Figure 1 show an interesting asymmetry. The right-hand side of the median cost peak is very steep, which shows that instances become easily solvable or easily proved unsolvable. The slope on the left-hand side is much less abrupt. Similar asymmetries have been observed in for example the work on random Boolean satisfiability (SAT) problems (Crawford and Auton 1996; Kirkpatrick and Selman 1994; Mitchell *et al.* 1992). However, the slope in those curves is much steeper on the left-hand side, *i.e.*, the satisfiable area. For the SAT problem, the slope on the right-hand side is much less steep, because proving unsatisfiability of random Boolean expressions requires large search trees (Chvatal and Szemeredi 1988). It's interesting to see the opposite phenomenon here. That is, for the quasigroup completion problem it becomes relatively easy to show unsolvability to the right of the phase transition; whereas finding solutions, to the left of the transition, appears to be harder, relatively speaking.

The difference with the situation for SAT may be due to the fundamentally different nature of our instances. In the random SAT problems, all the constraints (Boolean disjunctions) are randomly generated. This is in contrast to our approach, where we start with a highly regular original set of constraints, defining the quasigroup structure. We then perturb this structure by randomly pre-assigning a set of values. Our instances combine a high degree of structure with an element of irregularity or uncertainty due to the pre-assigned values. The fact that we again observe a clear phase transition is evidence for the practical relevance of the phase transition phenomenon in general. There is a large body of recent work on the phase transitions observed in random distributions of constraint satisfaction problems. See, for example, Cheeseman et al. (1991), Crawford and Auton (1996), Gent and Walsh (1996), Mitchell et al. (1992), and Smith and Dyer (1996). Hogg *et al.* (1996) contains a collection of recent papers in the area.

Some important related work on hard problem instances involving underlying structure is that of Gent and Walsh (1995), and Zhang and Korf (1996). Both teams use the structure of combinatorial optimization problems, such as the Travelling Salesman Problem and real-world Timetabling problems. By varying the constraint density of their problem instances they obtain varying degrees of difficulty. One important difference is that in our approach we start with an initial problem structure for which a direct construction is known. As we will discuss below, this allows us to consider search control mechanisms that mimic known constructive methods and thus study their robustness under perturbations. We also believe that our domain fits more naturally with reasoning style problems ("satisfaction problems") as opposed to combinatorial optimization style problems.

In the next section, we discuss how our problem instances enable us to evaluate the robustness of different search strategies. We will show that the relative performance of search strategies varies, depending on the location of the problem instances with respect to the phase transition. This suggests ways of chosing search control depending on the statistical properties of the problems under consideration.

A Comparison of Search Heuristics

There are several direct methods for generating basic quasigroups of any order N. For example, consider placing the elements of the quasigroup in the first row of the multiplication table in some arbitrary order. Now, we can generate the next row, by simply shifting the elements from the first row, one place to the right (the rightmost element wraps around and moves to the leftmost position of the second row). We repeat this process for the third row by shifting the pattern of the second row another cell to the right. The remaining rows are filled in a similar manner. By this construction, it is clear that there will be no repeated elements in any row, nor in any column. We therefore end up with a latin square of order N, which defines the multiplication table of a quasigroup.

We can mimic this constructive method using a backtrack procedure with appropriate search control. In particular, we can add a tie-breaking criterion to the generic first-fail heuristic. In the first-fail heuristic, one selects as the next branching variable the one with smallest remaining domain. We add the following tie-breaking criterion: select from among the variables with the smallest current domains, the variable with the minimal value in its current domain. (We assume some arbitrary fixed order on the quasigroup elements.) Our experiments below show that this simple tie-breaking heuristic will generate quasigroups of any order with zero backtracks. So, our tie-breaking rule effectively mimics a constructive method. The key question is how robust such a heuristic is in the presence of perturbations on the structure.

In Figure 3, we compare the performance of this specialized heuristic with the more general first-fail strategy on the quasigroup completion problem with no pre-assigned values. The figure uses a logarithmic scale. In order to show the tie-breaking heuristic on these instances, we added "2" to its median number of backtracks. We see from the points on the horizontal line that this heuristic indeed does not create any

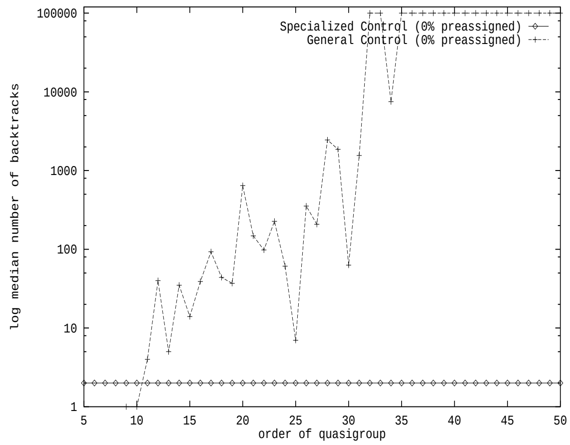

Figure 3: General vs. Specialized Search Control (0% Pre-assignment).

backtracks. However, the number of backtracks for the general heuristic increases sharply with increasing N, reaching our cutoff of $100,000$ when N is 32.

We now consider the change in behavior when we slightly perturb the quasigroup structure by pre-assigning 5% of the elements of its multiplication table. See Figure 4. The figure shows the performance of our tie-breaking heuristic degrades dramatically. In fact, the general first-fail heuristic without tie-breaking actually scales substantially better. (Note the logarithmic scale.) In other words, the tie-breaking heuristic is surprising "fragile" in the presence of small perturbations.

Figure 5 further confirms the difference in performance between running with and without the tie-breaking rule. This time we consider instances in the hardest area of the phase transition (about 40% pre-assignment). We see that the specialized tie-breaking rule performs consistently worse on the hardest quasigroup completion problems, near the phase transition. Even on a log scale the gap between the heuristic actually widens for larger N. (Our search cutoff actually hides some of this difference.) So, except for completely unperturbed versions of our search problem, the first-fail heuristic with tie-breaking is consistently worse than the first-fail heuristic by itself. This despite the fact that the tie-breaking rule mimics a construction method for quasigroups, in the unperturbed version.

Figure 6 illustrates in more detail the performance of the general heuristic. With a small number of pre-assigned values (1%), the heuristic performs better than with no pre-assigned values at all. However, its performance deteriorates again in the hard area of the phase transition. Apparently, the small number of randomly pre-assigned values helps the first-fail heuristic in finding a completion of the quasigroup. At an in-

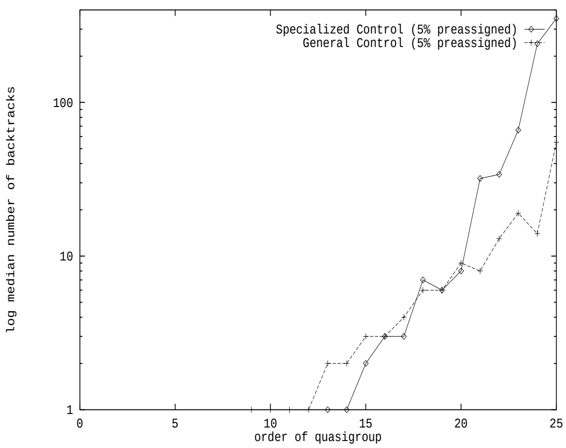

Figure 4: General vs. Specialized Search Control (5% Pre-assignment).

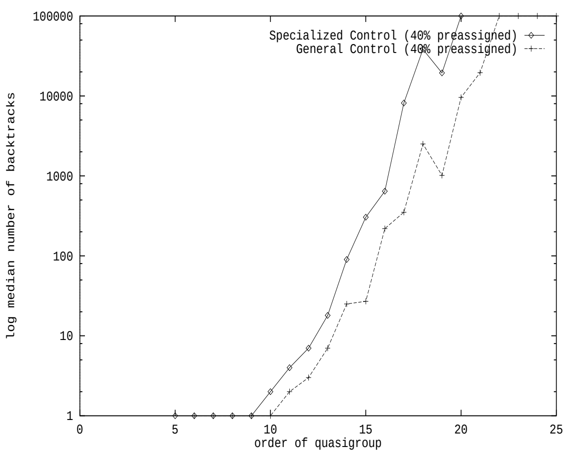

Figure 5: General vs. Specialized Search Control with 40% Perturbation

tuitive level, this is consistent with the fact that the specialized heuristic actually performs worse. The tie-breaking strategy in this heuristic makes it more deterministic, which appears to hurt its performance on the completion task.

To summarize, we observed that the performance of specialized search control — designed to mimic a construction method — degrades rapidly in the presence of small perturbations. More general heuristics appear more robust. We also saw how small perturbations can actually improve the performance of a general heuristic. We believe that randomness versus determinism plays a key role in these phenomena. Our findings suggest care should be taken in the use of tailored heuristics: their performance can degrade dramatically in the presence of minor perturbations. In presence of such

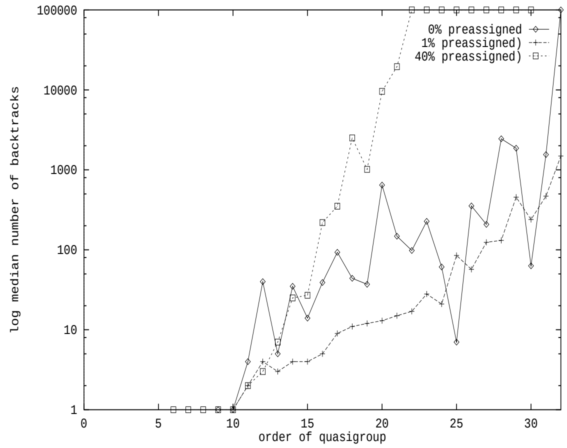

Figure 6: General Search Control. Effect of Pre-assignment.

perturbations, less sophisticated, more generic heuristics may very well have better scaling properties.

Closely connected to our work is the study of search control and various constraint processing techniques (Dechter 1991, Freuder *et al.* 1995, Ginsberg and Geddis 1991, Kondrak and van Beek 1995), and the work on selecting appropriate search heuristics (see eg. Minton 1996). We hope that our approach will stimulate more research into the robustness of the various methods.

The idea of completing partial solutions is, of course, applicable in many CSP domains. Therefore, by asserting partial solutions, that are at least locally consistent, one can study whether special search techniques degrade gracefully.

Conclusions and Future Work

The study of the complexity and performance of search procedures when applied to realistic problems is greatly hampered by the difficulty in gathering realistic data. As an alternative, researchers heavily resort to randomly generated instances or highly structured problems from finite algebra. The random instances clearly lack sufficient structure, whereas the finite algebra problems are, in some sense, too regular. In order to bridge this gap, we introduced a new benchmark domain, the quasigroup completion problem.

Our problem instances provide a natural testbed for studying the robustness of search procedures. For example, in the presence of small perturbations, we found that specialized search control can perform substantially worse that a generic control strategy. This discrepancy is even more accentuated when dealing with harder problems near the phase transition. In our domain, a specialized search strategy that mimics a constructive method only performs well when dealing with the problem in its pure, undisturbed, form. Any small level of perturbation makes it degrade rapidly. On the other hand, small perturbations seem to *help* a general search control mechanism.

Acknowledgments
We would like to thank Karen Alguire for developing an exciting tool for experimenting with the quasigroup completion problem. We also would like to thank Nort Fowler for many useful suggestions and discussions, and Neal Glassman for suggesting the domain of combinatorial design as a potential benchmark domain. The first author is a research associate with Rome Laboratory and is funded by the Air Force Office of Scientific Research, under the New World Vistas Initiative (F30602-97-C-0037 and AFOSR NWV project 2304, LIRL 97RL005N25).

References

Andersen, L. (1985). Completing Partial Latin Squares. *Mathematisk Fysiske Meddelelser*, 41, 1985, 23–69.

Andersen, L. and Hilton, J. (1983). Thank Evans! *Proc. London Math. Soc.*, (3) 47 (1983), 507-522.

Cheeseman, Peter and Kanefsky, Bob and Taylor, William M. (1991). Where the Really Hard Problems Are. *Proceedings IJCAI–91,* 1991, 163–169.

Chvatal,V. and Szemeredi, E. (1988). Many hard examples for resolution. *JACM*, val. 35, no. 4, 1988, 759–208.

Colbourn, C. (1983). Embedding Partial Steiner Triple Systems is NP-Complete. *J. Combin. Theory* (A) 35 (1983), 100-105.

Colbourn, C. (1984). The Complexity of Completing Latin Squares. *Discrete Appl. Math.*, 8, (1984), 25-30.

Crawford, J. and Auton, L. (1996) Experimental Results on the Crossover Point in Random 3-SAT. *Artificial Intelligence*, 81, 31–57.

Dechter, R. (1991) Constraint networks. *Encyclopedia of Artificial Intelligence* John Wiley, New York (1991) 276-285.

Denes, J. and Keedwell, A. (1974) Latin Squares and their Applications. *Akademiai Kiado, Budapest, and English Universities Press*, London, 1974.

Evans, T. (1960) Embedding Incomplete Latin Squares. *Amer. Math.*, 67 (1960), 958-961.

Fujita, M., Slaney, J., and Bennett, F. (1993). Automatic Generation of Some Results in Finite Algebra *Proc. IJCAI*, 1993.

Freuder, F., Dechter, R., Ginsberg, M., Selman, B., and Tsang, E. Systematic versus stochastic constraint satisfaction. *Proc. IJCAI-95*, Montreal, Canada (1995).

Freuder, E. and Mackworth, A. (Eds.). *Constraint-based reasoning.* MIT Press, Cambridge, MA, USA, 1994.

Gent, I. and Walsh, T. (1995) The TSP Phase Transition. First Intl. Workshop on AI&OR. Timberline, OR, 1995.

Gent, I. and Walsh, T. (1996) The Satisfiability Constraint Gap. *Artificial Intelligence*, 81, 1996.

Ginsberg, M. and Geddis, D. (1991) Is there a need for domain-dependent control information? *Proc. AAAI-91.*

Hogg, T., Huberman, B.A., and Williams, C.P. (Eds.) (1996). Phase Transitions and Complexity. *Artificial Intelligence*, 81 (Spec. Issue; 1996)

Kirkpatrick, S. and Selman, B. (1994) Critical Behavior in the Satisfiability of Random Boolean Expressions. *Science*, 264 (May 1994) 1297–1301.

Kondrak, G and van Beek, P. (1995) A theoretical evaluation of selected backtracking algorithms. *Proc. IJCAI-95*, 1995, 541–547.

Lam, C., Thiel, L., and Swiercz, S. (1989) The Non-existence of Finite Projective Planes of Order 10. *Can. J. Math.*, Vol. XLI, 6, 1989, 1117-1123.

Mitchell, D., Selman, B., and Levesque, H.J. (1989) Hard and easy distributions of SAT problems. *Proc. AAAI-92*, San Jose, CA (1992) 459–465.

McCune, M. (1996) Solution of the Robbins Problem. Draft, 1996.

Minton, S. (1996) Automatically configuring constraint satisfaction programs: A case study. *Constraints*, 1 (1).

Puget, J.-F. (1994) A C++ Implementation of CLP. *Technical Report 94-01* ILOG S.A., Gentilly, France, (1994).

Smetaniuk, B. (1981) A New Construction on Latin Squares - 1: A Proof of the Evans Conjecture. *Ars Combinatoria* XI (1981), 155-172.

Smith, B. and Dyer, M. Locating the Phase Transition in Binary Constraint Satisfaction Problems. *Artificial Intelligence*, 81, 1996.

Williams, C.P. and Hogg, T. (1992) Using deep structure to locate hard problems. *Proc. AAAI-92,* San Jose, CA, July 1992, 472–277.

Zhang, W. and Korf, R. A Study of Complexity Transitions on the Asymmetric Travelling Salesman Problem. *Artificial Intelligence*, 81, 1996.