Compute-Intensive Methods for Artificial Intelligence

Bart Selman
Computer Science Department
Cornell University

Synopsis of Proposed Integrated Research and Educational Activities

My overall objectives are threefold: (a) to develop new formalisms and methods for artificial intelligence by combining a sound theoretical approach with a principled experimental component, (b) to provide a practical evaluation of the proposed models and relate them to real-world applications and challenges, and (c) to educate the next generation of CS students about the ambitious and challenging goals of artificial intelligence.

The focus of my research is on compute-intensive methods for artificial intelligence (AI). Traditionally, general search and reasoning is largely avoided in AI by explicitly incorporating large amounts of domain-specific knowledge. While such a knowledge-intensive approach has been successful in certain domains, as demonstrated by expert system applications, in other areas, such as planning or general reasoning, progress has been disappointing. However, recent advances in general search and reasoning methods combined with faster hardware and better implementations provide strong evidence that a compute-intensive approach is not only suitable for dealing with the combinatorial nature of many AI formalisms, but may also be required to supplement domain-specific knowledge, especially considering the knowledge-acquisition bottleneck in terms of encoding highly specific domain knowledge.

I propose to study fast, general reasoning and search methods, with an emphasis on stochastic procedures, which are a promising recent development for solving computationally hard problems. I will also investigate the various sources of complexity in hard problems, using both theoretical and experimental methods. This work explores interesting connections between computer science, artificial intelligence, and statistical physics. In addition, I will study issues in problem representation, including the robustness of encodings, abstraction, compilation, and approximation methods. These issues are critical to the successful application in realistic domains of reasoning and search methods. In terms of applications, I'm interested in challenge problems from planning, learning, and knowledge representation. I will also study applications from other disciplines, such as operations research and computational biology.

As part of my educational objectives, I plan to enrich the AI curriculum by emphasizing a more central role for AI with respect to the rest of computer science. In addition, I will establish links with other disciplines, such as physics, operations research, and computational biology. I plan to achieve these objectives by incorporating in the AI course curriculum concrete challenge problems from areas such as game playing, planning, learning, and reasoning, developing an advanced course on compute-intensive methods in AI, supervising student research projects, and organizing inter-disciplinary seminars. My research program provides a close match with these objectives. For example, by focusing on computational issues and problem representation, the students will encounter concepts from computational complexity theory, algorithm design, and advanced data-structures. Also, various stochastic and systematic search methods developed in AI for solving hard computational problems are directly applicable to computational challenges encountered in other areas, such as in operations research and computational biology.

Compute-Intensive Methods for Artificial Intelligence

Bart Selman
Computer Science Department
Cornell University
Ithaca, NY 14853
SELMAN@CS.CORNELL.EDU

Proposal for NSF Faculty Early Career Development Award (1998-2002). This document provides a compresensive survey of my research program.

1 Introduction

In the 70's and 80's the success of knowledge-intensive approaches to problem solving eclipsed earlier work on compute-intensive weak methods. However, in recent years, compute-intensive methods have made a surprising comeback. One of the most prominent examples is the success of IBM's Deep Blue in defeating Gary Kasparov in the 1997 ACM Challenge match. Deep Blue's performance led Kasparov to exclaim, "I could feel — I could smell — a new kind of intelligence across the table." Deep Blue derives its strength mainly from highly optimized search (Kasparov 1997, McDermott 1997). Another dramatic development in the compute-intensive approach was the recent computer proof resolving the Robbins problem (Kolata 1996). The Robbins problem is a well-known problem in Boolean algebra, and was open for over sixty years. The computer proof was found by applying powerful search techniques guided by general search tactics. Several aspects of the computer proof could be called "creative" by mathematicians' standards. Deep Blue's performance and the resolution of Robbin's theorem are good examples of a *qualitative* change in performance of compute-intensive approaches compared to just a few years ago.

In my own work, I've focused on compute-intensive methods with rich underlying representations. SATPLAN (Kautz and Selman 1996) is a planner that challenges the widespread belief in the AI community that planning is not amenable to general theorem-proving techniques. SATPLAN shows that general propositional satisfiability algorithms can outperform specialized planning systems on a range of benchmark problems. At the core of SATPLAN lies a powerful model finding procedure. This procedure was developed as part of my work on stochastic model finding methods for propositional theories, such as GSAT (Selman *et al.* 1992, Selman 1995a, 1997a). These stochastic methods have significantly extended the range and size of constraint and satisfiability problems that can be solved effectively.

It has now become feasible to solve problem instances with tens of thousands of variables and up to several million constraints. Being able to tackle problem encodings of this size leads to a qualitative difference in the kinds of tasks that one might consider for general search and reasoning methods.

I predict that this is just the beginning of a shift to compute-intensive methods in AI. Given the tremendous difficulty of supplying a program with certain kinds of domain-specific knowledge, we may in fact be better-off attacking the inherent combinatorics head on. I don't mean to suggest that simply faster machines and implementations will be sufficient. Further research on search and reasoning procedures and on problem encodings will certainly be needed. Also, there are still many examples where general methods quickly become ineffective. Nevertheless, we now have concrete evidence that the battle against combinatorics can be won, at least in certain interesting cases. The challenge is to continue pushing the compute-intensive approach into areas that have previously been considered off-limits.

In my research on compute-intensive methods for AI, I will concentrate on the following areas:

- *The nature of hard computational problems and its connections to statistical physics.* During the past five years, we have obtained a much better understanding of the nature of computationally hard problems, mainly by studying distributions of random problem instances (Hogg *et al.* 1996, Kirkpatrick and Selman 1994, and Selman and Kirkpatrick 1996). A better understanding of more structured instances, as they occur in actual applications, is needed. This is a rapidly emerging area of research incorporating concepts and methods from statistical physics.

- *Robustness, uncertainty, and the brittleness of knowledge representation schemes.* Subtle differences in problem representations can lead to large differences in our ability to solve the underlying problem. Moreover, many problem encodings used in, for example, planning and reasoning, lead to solutions that are quite brittle when it comes to small changes in the original problem. A *principled* approach for designing "good" problem representations is needed.

- *Experimentation with close ties to algorithmic and theoretical work.* Worst-case and theoretical average-case results for search and reasoning procedures often tell us surprisingly little about their performance on real-world problems. In general, detailed *experimentation* is necessary to analyze search and reasoning strategies. Such studies often give rise to new algorithms, but there is also room for the development of theoretical models to explain the empirically observed phenomena.

- *Reasoning from first-principles: Compilation, approximation, and abstraction.* Compute-intensive methods operate from "first-principles." That is, little or no domain-specific search control knowledge is used. This kind of reasoning or search generally requires substantial computational resources. In building real-time systems, it is often desirable to shift the computational cost to an off-line, pre-processing (compilation/abstraction) phase. There are interesting tradeoffs to be explored between the cost of preprocessing and on-line savings.

Research on abstraction and reformulation may also shed new light on human problem solving abilities. The success of compute-based methods in reaching expert-level performance on certain tasks revives the intriguing question of how human cognition is able to avoid the inherent combinatorics of the underlying problem domains. Humans appear to employ clever abstractions, problem reformulations, or perhaps heuristic rules to dramatically change or reduce the search space. However, exactly how this is done is still very much an open research issue. Questions about human problem solving abilities were first formulated by some of the AI pioneers, for example, by Newell and Simon (1971) in their work on problem solving.

- *Challenge applications in planning, operations research, learning and computational biology.* AI planning is a notoriously hard combinatorial search problem. The recent work on reformulating planning in terms of a large constraint satisfaction problem has brought a novel perspective on AI planning, and there is still much room for improvement in this area. Other promising areas of application of compute-intensive methods are in learning, operations research, and computational biology.

Work on compute-intensive methods also provides exciting educational opportunities, particularly because of the many connections to other areas of computer science and to other disciplines. This fits well with my educational goal of enriching the AI curriculum at Cornell by emphasizing a more central role for AI with respect to the rest of computer science. I strongly believe that AI presents some of the most fascinating problems in computer science. After all, the search for an understanding of the human mind is considered by many to be one of the final frontiers of modern science. Many of the questions on how to create some form of machine intelligence are as pertinent today as they were several decades ago when they were first posed by some of the early pioneers of computer science, such as Turing, von Neuman, and Shannon. I believe that with the proper presentation of the achievements of AI combined with a discussion of its many open research challenges, we can educate the next generation of AI researchers, while, in addition, provide students in other areas with a solid appreciation of AI. I will also establish links with other disciplines, in particular with statistical physics, operations research, and computational biology. My own inter-disciplinary research interests are a natural fit with these educational objectives. I will provide more concrete details of my educational plans below.

2 The Nature of Hard Computational Problems

Complexity results and connections to statistical physics

Many formal intractability results in AI are based on a worst-case analysis, and hence, there has been much debate about their practical relevance. Average-case complexity analysis, on the other hand, would appear to be more directly applicable. However, such analysis requires a precise model of the distribution of input instances. Since we do not have a good understanding of real-world problem distributions, average-case results generally assume relatively simple distributions of randomly generated problem instances. Average-case results based on such distributions show that almost all randomly generated instances of combinatorial problems are surprisingly easy to solve. This has led some AI researchers to dismiss many of the negative worst-case complexity results. Recent work, however, has shown that the positive average-case results are largely due to the particular choice of input distributions. By being more careful in generating instances, one can in fact quite easily obtain extremely hard search and reasoning problems. The key property of such hard random instances is that they have to be *critically constrained*. That is, they occur at a certain critical ratio of variables to constraints. At this ratio, the problem distributions undergo dramatic changes. An analogous phenomenon occurs in phase transitions studied in physics, and tools from statistical mechanics can be used to analyze the transition phenomenon. (See, for example, Cheeseman *et al.* 1991, Delahaye 1995, Hayes 1997, Hogg *et al.* 1996, Kirkpatrick and Selman 1994, and Mitchell *et al.* 1992.)

In Mitchell *et al.* (1992) and Selman *et al.* (1996), we identified the connection between the computational difficulty of the propositional satisfiability problem and a phase transition phenomenon occurring at a critical ratio of around 4.3 clauses (constraints) per variable. Below this ratio almost all (CNF) formulas are satisfiable; above this ratio almost all formulas are unsatisfiable. Subsequently, in Kirkpatrick and Selman (1994), we showed how methods from statistical physics can be used to analyze the phase transition phenomenon in the satisfiability problem. This work provided some of the first evidence that the connection between the nature of combinatorial search problems and systems studied in statistical physics goes beyond just superficial similarities. In fact, some of the mathematical models from statistical physics can be used to capture key properties of combinatorial search problems.

In our most recent work (Monasson *et al.* 1996), we study a mixture of instance distributions from a tractable and an intractable class. More specifically, we consider a mixture of binary and ternary propositional clauses. The goal is to explore the *transition* from a tractable problem (2-SAT) to an intractable one (3-SAT) in a controlled manner by slowly increasing the fraction of 3-SAT clauses. Instead of seeing the distributions become increasingly harder when increasing the fraction of 3-SAT, we observe that up to a certain fraction (around 40%) the system essentially behaves as pure 2-SAT, whereas above this threshold the ternary clauses determine the overall behavior. This finding provides further insights into why the combinatorial explosion does not occur in practice in certain systems that are intractable in the worst-case: *a tractable subclass may dominate the overall behavior*. In addition, we show, using the tools from statistical physics, that the nature of the combinatorial search space changes dramatically at the critical fraction.

Work on the application of models from statistical physics to the study of combinatorial problems has just begun, and it is reasonable to expect significant further developments in this area in the near future. This area has been central in my research agenda and I plan to actively pursue it. From the perspective of applications in AI, possibly the most exciting opportunities involve the development of new search methods, either systematic or stochastic, based on the new insights gained about the nature of the combinatorial search space. For example, using our work on typical instances from mixed problem distributions, I will investigate *hybrid methods* that smoothly combine the best algorithms for the general intractable case with fast specialized algorithms for tractable subclasses.

Random versus structured problem instances

Randomly generated critically constrained problem instances have been used extensively in the study and development of algorithms. A key question is whether the results obtained for such instances are at all indicative of the behavior of the algorithms on more structured, real-world instances. The results of the recent DIMACS Challenge on Satisfiability Testing suggest that the behavior of algorithms on hard random problems can indeed be representative of the behavior on more structured problems. However, there is the concern that we may be reaching a point where this is no longer the case, and that the simple random distributions now used for testing may be driving us in the wrong direction in our research (Johnson 1996).

In Gomes and Selman (1997a, 1997b), we introduce an alternative to pure random instances. In our model, we take a highly structured problem from the domain of finite algebra, and introduce an element of randomness to perturb its structure. The resulting problem provides a good framework for studying issues such as the robustness of search strategies. Hard instances can be generated at a certain critical level of perturbation of the original problem. Using these problem distributions, we showed that the runtime distribution of a stochastic backtrack-style procedure is highly non-standard (Gomes *et al.* 1997). More specifically, we encountered so-called *heavy-tailed* distributions, characterized by infinite mean and infinite variance. We have since also found such non-standard distributions when solving real-world planning and scheduling problems. These results suggest a possible fractal dimension and self-similar characteristics of the combinatorial search space (Mandelbrot 1983). Our work also provides a first step in the direction of obtaining more realistic benchmark problems for evaluating reasoning and search procedures. An obvious next step is to design problem generators for instances with properties that are even closer to those of real-world instances. (See also Dean *et al.* 1996, and Zhang and Korf 1996).

The study of more structured benchmark problems has a direct payoff in terms of the design of algorithms. I plan to study further the heavy-tailed and self-similar nature of combinatorial search and its implications for algorithm design. For example, the long tails of the cost distributions suggest that one could use parallel or inter-leaved runs of several stochastic methods to obtain a lower expected computational cost for the overall ensemble. I plan to investigate provably optimal ways of building such portfolios, and strategies for optimally combining independent runs of the same stochastic methods. The objective is to obtain a low expected computational cost for the overall approach while minimizing the risk of not finding a solution at all (*i.e.*, we want to maximize the "predictability" of the ensemble). Optimal strategies will depend on the available resource bounds and the particular problem instance(s) under consideration. Such strategies will therefore need to take into account the uncertainty in the availability of resources combined with the uncertainty about the exact nature of the instances under consideration. For related work, see the work on flexible computation (Huberman *et al.* 1997, Horvitz and Zilberstein 1996) and on bounded rationality (Russell and Norvig 1995, Russell and Subramanian 1993).

3 Robustness, Uncertainty, and the Brittleness of Encodings

Encodings

The success of compute-intensive approaches depends on our ability to find suitable encodings of real-world problems. There has been significant recent progress along this front using general constraint-based representations and propositional encodings. Examples include work on planning (Blum and Furst 1995; Kautz and Selman 1996), problems in finite algebra (Fujita *et al.* 1993), verification of hardware and software protocols, scheduling (Crawford and Baker 1994), circuit synthesis and diagnosis (Larrabee 1992), and many other domains. Experience has shown that different encodings of the same problem can have vastly different computational properties. For example, in planning, "causal" encodings appear to be harder to solve than "state-based" encodings (Kautz *et al.* 1996).

The challenge that I'm interested in addressing is how to develop a general characterization of encodings that can be *efficiently* solved. This characterization may involve, for example, understanding the relationship between the encoding and the shape of the combinatorial search space. Note that the characterization cannot be as simple as stating that the search space has no local minima. For example, the propositional encodings of blocks-world planning problems do have deep local minima, yet can be solved by local search. We need to understand why search can escape from local minima in some encodings but not in others. Perhaps one can find easily measurable statistical properties of the encoding that predict the behavior of various search algorithms on a given instance.

The evaluation of progress in this area cannot be purely objective. There will be some subjective judgement necessary in determining whether the characterization is useful and enlightening. *Predictions* of the theory can be put to an empirical test: in the best case, the general principles would suggest new encodings which can be solved more easily than previous ones.

Robustness

Another problem with the logical encodings for real-world problems that have been suggested in the literature is that they are extremely "brittle". If we look at formulations in which models that satisfy the encoding correspond to solutions to the original problem instance, we see little relationship between models that "almost" satisfy the encoded problem and candidate solutions that "almost" satisfy the original problem instance. For example, in Kautz and Selman (1992), we noted that it was easy to find truth-assignments that satisfied all but a *single* clause of the propositional encodings of blocks-world planning problems, even though these "near models" had no resemblance to valid plans. (The near models represent an initial series of random moves, followed by a final step, in which the blocks "magically" re-arrange themselves into the correct position.)

I plan to study the robustness of encodings, and in particular, how the robustness of local search algorithms applied to those encodings could be improved by finding ways to more closely align the syntactic form of the encodings with the semantics of the source domain. Some of the early work in AI on *analogical representations* may be applicable here. Analogical representations or "analogues" are representations in which there is a close relation between the syntactic form of the representation and its semantic content. Pictorial representations are an example of analogues. The study of such representations is related to a large body of research on the use of visual images in human cognition (Kosslyn 1981, Levesque 1986, Selman 1997c, and Sloman 1971).

Uncertainty

The real world is a highly uncertain place, and most reasoning involves determining what is likely, not just what is possible. Another research challenge is therefore to find proper ways of extending purely constraint-based and propositional encodings to incorporate notions of probability and uncertainty. In recent years, there has been tremendous activity in the area of uncertainty and AI, as witnessed by the increasing prominence of the UAI community (Pearl 1988). With respect to knowledge representation schemes, there has also been significant progress in incorporating notions of probability and uncertainty in more standard logical representations (*e.g.,* Halpern 1997 and Nilsson 1986, 1998). In general such proposals increase the computational complexity of the underlying representations. I plan to study how to extend purely logical representations, while maintaining good *practical* computational properties. This will require not only the adaptation of the representations but also some adaptation of the search and reasoning algorithms in current use.

4 Experimentation with Close Ties to Algorithmic and Theoretical Work

An important recent advance in the area of reasoning and search was the introduction of stochastic local search methods as a viable alternative to systematic procedures. For constraint satisfaction and scheduling, Minton *et al.* (1992) proposed the Min-Conflict procedure, and, in my own work on propositional reasoning, I introduced GSAT, a greedy local search method for finding models of propositional theories (Selman *et al.* 1992, Selman and Kautz 1993a, 1993b, Selman *et al.* 1994, 1996, and Selman 1995a, 1997a). These stochastic local search procedures have significantly extended the range and size of constraint and satisfiability problems that can be solved effectively. For example, between 1991 and 1996 the size of hard satisfiability problems that could be feasibly solved grew from problems involving less than 100 variables with up to 400 constraints (propositional clauses) to problems involving over 10,000 variables and up to 1,000,000 constraints. Being able to handle problem encodings of this size has significantly extended the practical value of constraint satisfaction and satisfiability procedures.

Progress in this area has been spurred by the interaction of researchers in AI, operations research, and theory, by making available benchmark problems (DIMACS, see Trick and Johnson 1996), holding joint workshops (Ginsberg 1995), and sharing algorithms and code. There is a growing consensus, however, that certain key technical challenges must be addressed in order to continue to increase the range of problems that can be practically solved. In the next two sections, I'll discuss the challenges that I will pursue in this area.

Stochastic Methods

An important research issue for stochastic search methods is the problem of distinguishing "dependent" from "independent" variables. Propositional or constraint-based encodings of structured problems such as planning and diagnosis often contain large numbers of variables whose values are a simple function of other variables. We call these *dependent* variables. Variables whose values cannot be easily determined to be a simple function of other variables are called *independent*. For a given propositional theory there may be many different ways to classify the variables as dependent and independent. But for most encodings of real-world problems there is a natural division between dependent and independent variables. Since an assignment to the independent variables determines a truth value for each dependent variable, the number of assignments that need to be considered in systematic search is at most 2^n, where n is the number of independent variables. (The distinction between dependent and independent variables has also received much attention in the area of probabilistic reasoning. See, *e.g.*, Pearl (1988).)

One can improve the performance of systematic search methods by first identifying the set of independent variables, and then branching only on those variables in the backtrack search. In local search methods, it appears much harder to exploit the distinction between dependent and independent variables. The challenge is to improve local search for problems with many dependent variables: the search should concentrate only (or mainly) on the independent variables, and some fast mechanism should then set the dependent variables.

Another issue in the use of stochastic local search methods is their inherent incompleteness: if they fail to find a model for a theory (or formula) one cannot be *certain* that the theory is inconsistent. This has led to an asymmetry in our ability to solve satisfiable and unsatisfiable instances drawn from the same problem distribution. Stochastic algorithms can solve hard random satisfiable formulas containing thousands of variables, but we cannot solve unsatisfiable instances of the same size. The challenge is to design a local search method for proving unsatisfiability. This apparently would require searching in the space of refutation proofs, rather than in the space of truth assignments. Each state would be an incomplete proof tree, *e.g.*, a proof tree that rules out some fraction of the truth assignments. The neighborhood of a state would be similar proof trees. Each step in the local search would try to transform the proof into one that rules out a larger fraction of assignments.

Systematic and local search procedures outperform each other on different problem classes. Even putting the issue of incompleteness aside, there exist classes of problems for which one or the other approach is clearly the winner. This leads to the general question: Can we develop a procedure that leverages the strengths of each? The obvious way, of course, is to simply run good implementations of each approach in parallel. But is there a more powerful way of combining the two? Recently there has been some intriguing work on using local search to implement the variable ordering heuristic for systematic search (Boufkhad 1996; Mazure *et al.* 1996). These methods and other ways of combining systematic and stochastic search need to be further developed and compared with previous approaches on a range of benchmark problems.

Systematic search

In addition to the developments in the area of stochastic local search methods, there has also been significant progress in terms of systematic search procedures. Most systematic procedures employ some form of backtrack-style search. In order to speed up these procedures, research has focused on reducing the size of the backtrack search tree, which can be achieved by using better variable and value selection heuristics. Other strategies involve preprocessing of the constraints (work on various forms of K-consistency), the derivation of new constraints during the backtrack search, and strategies for dynamic variable ordering (Dechter 1991, Dechter and Meiri 1994 Freuder and Mackworth 1994, Ginsberg 1993b, and Korf 1991, 1993).

New strategies continue to be discovered in this area but researchers are faced with a fundamental tradeoff between the computational savings achieved through the reduction in the overall size of the search tree and the additional computational cost of applying more sophisticated variable and value selection strategies combined with the cost of deriving new constraints. Somewhat surprisingly, it appears that the simplest backtrack strategies with only a limited amount of preprocessing are the most effective overall (Dubois *et al.* 1996). Nevertheless, these simple but extremely fast procedures (exploring 20,000 to 30,000 nodes per second) do not necessarily generate the most compact search trees. I believe that it should be possible, using careful algorithmic design guided by rigorous experimental evaluations, to develop more sophisticated backtrack style methods that do outperform — *in terms of overall computational cost* — the very fast simple procedures. (See also Ginsberg 1993a, and Freuder and Mackworth 1994.)

Another fundamental limitation of the best current backtrack-style search procedures is that they are all based on creating some form of resolution-style proof for inconsistency or unsatisfiability. This includes depth-first search algorithms such as the Davis-Putnam procedure, where the proof tree can be recovered from the trace of the algorithm's execution, but is not explicitly represented in a data structure (the algorithm only maintains a single branch of the proof tree in memory at any one time). Unfortunately, there are known fundamental limitations on the size of the shortest resolution proofs for many classes of problems (Haken 1985; Chvatal and Szemeredi 1988). For example, "pigeon hole" problems (showing that n pigeons cannot fit in $n-1$ holes) are intuitively easy, but shortest resolution refutation proofs are of exponential length, and therefore the Davis-Putnam procedure will require exponential time to solve a pigeon-hole formula. Shorter proofs do exist in more powerful proof systems. Examples of proof systems more powerful than resolution include extended resolution, which allows one to introduce new defined variables, and resolution with symmetry-detection, which uses symmetries to eliminate parts of the tree without search. Assuming

NP$\neq$co-NP, even the most powerful propositional proof systems would require exponential long proofs worst-case — nonetheless, such systems provably dominate resolution in terms of minimum proof size.

Attempts to mechanize more powerful proof systems usually yield no computational savings, because it appears *harder* to find the small proof tree in the new system, than to simply crank out a large resolution proof. In essence, the overhead in dealing with more powerful rules of inference consumes all the potential savings. There is promising work in this area (Crawford *et al.* 1996; de la Tour and Demri 1995), but not yet convincing empirical results on a variety of benchmark problems. Given the potentially large payoff in obtaining search methods that are not affected by the limitations of standard resolution, further research is warranted to find practical procedures based on more powerful proof systems.

5 Compilation, Approximation, and Abstraction

In real-time systems a fast response to a dynamically changing environment is of the essence. A promising strategy in building such systems is to shift the computational cost of reasoning and search to an an off-line compilation phase. The general idea is to take the underlying domain theory — written in an unrestricted representation language — and compile this theory in a more restricted, tractable sublanguage. An exact translation is often not possible, because of the more limited expressive power of the sublanguage. The challenge is then to find the best possible *approximation* of the original domain theory.

In Selman and Kautz (1996), we introduce a general framework for knowledge compilation and theory approximation. The idea is to generate two bounding approximations of the original theory. The bounds are given in a tractable sublanguage, for example, a propositional Horn theory. Given that there are many logical theories that cannot be written in an equivalent Horn form, our proposal is based on generating two approximating theories: one logically weaker ("more abstract") and one logically stronger ("more specific"). We show that many queries about the original theory can be handled efficiently by using the two bounding approximations. However, computing the best bounds is computationally expensive. The compilation should be viewed as an off-line process, whose cost can be amortized over a large number of subsequent on-line queries. We demonstrate on a series of background theories that the approach indeed saves time compared to a general reasoning method running on the original theory.

In the area of diagnosis, Williams and Nayak (1996) demonstrate the effectiveness of compilation by showing how a general declarative representation of the engine system controlling the next generation of NASA spacecrafts can be compiled into a theory with good computational properties. The system uses the compiled theory to perform efficient on-line diagnosis during flights.

So far, most work in this area has focused on compilation using a polynomially decidable target language. Unfortunately, in certain cases the resulting approximations are exponentially longer than the originally theory, thus defeating the initial purpose of the compilation (Selman and Kautz 1996). I plan to investigate compilation and approximation methods that transform a theory in one that is not necessarily polynomially decidable, but nevertheless has good computational properties for subsequent processing with the recent fast stochastic and systematic search and reasoning procedures. Insights obtained from studying the characteristics of hard computational problems and the properties of various encodings, as discussed above, should be useful in developing such compilation and approximation strategies.

6 Challenge Applications

Applications are an integral part of the study and development of compute-intensive methods for AI. In fact, many of the research issues I've discussed above first arose when considering specific applications. Challenge applications expose the limitations of compute-intensive methods in terms of the effectiveness of the reasoning and search methods as well as reveal shortcomings in terms of problem representations. In my research, I will continue to pursue simultaneously the development of compute-intensive methods and their application. Below are examples of some of the most promising challenge application areas, including a brief statement of my research objectives for each area. I will restrict myself here to three example domains. However, I also plan to explore potential applications in other AI domains with hard computational problems, such as learning.

Let's consider planning as a first example of an application domain. SATPLAN (Kautz and Selman 1996) showed that general propositional satisfiability algorithms can be used to build efficient general planners. Our approach, related to the work on constraint-based planning by Blum and Furst (1995), provided a novel perspective on AI planning. See, for example, Kambhampati (1996, 1997) and Ernst *et al.* (1997). Critical to the success of our approach is the use of concise propositional representations of planning problems. In Kautz *et al.* (1996), we describe several general polynomial-time reductions from STRIPS-style planning (Fikes and Nilsson 1971) to propositional theories. We found that the most concise encodings do not necessarily provide the best performance in practice. In particular, the highly compact lifted causal encodings are surprisingly difficult to solve for current satisfiability procedures. I plan to investigate the kinds of encodings that lead to the best computational results on planning problems. There are also good opportunities in the planning domain for the study of compilation and preprocessing methods.

For our second application domain, we consider the area of scheduling as studied in operations research (OR). Linear programming and integer programming are the methods of choice in the OR community to tackle scheduling tasks. Constraint-satisfaction techniques and general search methods are the preferred tools in the AI community. OR methods excel at handling numeric information as part of the problem formulation, whereas AI techniques appear better suited for purely combinatorial challenges.

In Jiang *et al.* (1995), we provide some initial evidence that one can successfully incorporate numeric information, used to capture *soft constraints* or *preference information*, into a Boolean representation scheme. However, much remains to be done to synergetically combine techniques from AI and OR. For example, at least for certain kinds of problems, such as the pigeon hole problem, the OR style representations offer a distinct computational advantage (Ginsberg 1995). I plan to investigate ways to extend constraint-based and propositional representations, to incorporate features from OR-style representations based on linear and mixed integer programs.

Another interesting application domain for compute-intensive methods from AI is in the area of computational biology. In particular, I will consider the protein folding problem. Protein folding is one of the most challenging problems in computational biology. The goal in protein folding is to determine the three-dimension structure of a protein. The problem can be formulated as an energy minimization problem, where the final configuration of the molecule corresponds to a low energy state. Unfortunately, for proteins of practical interest, finding good energy minima is much beyond the state-of-the-art of current optimization methods. Given that the three-dimensional structure of the protein determines its function, being able to predict the structure would have a tremendous impact on methods for designing drugs.

Current methods for protein folding are generally based on numerical optimization techniques, such as simulated annealing (Kirkpatrick *et al.* 1983). An interesting question is whether some of the enhancements of stochastic search methods developed for purely combinatorial problems, such as the mixed random walk technique (Selman *et al.* 1994, 1996), can be of use in the protein folding problem. This is an open question at this point, but in other numerical optimization problems there has been success in using combinatorial techniques, such as graph coloring methods, as effective subroutines in numerical procedures. Cornell provides an excellent environment for studying protein folding as an application domain because of its excellent program in computational biology with strong connections to the numerical analysis group in Computer Science.

7 Educational Plans

Role within the department

A key factor in my decision to join Cornell was the opportunity to become involved in the education of the next generation of computer scientists, and to introduce them to the achievements and challenges of AI. My goal is to bring the AI curriculum closer to the general CS curriculum, by emphasizing the many connections between problems and issues in AI and other areas of computer science. It is a unique opportunity for me to join Cornell at this point in time. Cornell is committed to creating a strong presence in core AI. This gives us the opportunity to build up and shape the AI curriculum in the years to come. Last year, Joseph Halpern, a world-renowned expert in knowledge representation and reasoning about uncertainty, joined the department. I will complement his expertise with more emphasis on computational issues in AI combined with a strong experimental component. Together with the already strong presence in computer vision (Daniel Huttenlocher and Ramin Zabih), natural language (Claire Cardie and Lillian Lee), and learning theory (Ronitt Rubinfeld), we will be able to provide students with an in-depth coverage of all areas of AI.

Another important factor in my decision to join Cornell is the strong interdisciplinary nature of the department. The department plays a central role in many of the activities at the university. A recent report outlines the key role computer science plays within the university (CS Faculty 1997). The report starts with the observation that computers and computer network are rapidly changing our educational system, and will impact virtually all aspects of our lifes. Future generations of students, independent of their chosen field, will therefore need to develop a good understanding of the opportunities provided by the rapid developments in information technology. The goal of the department is to reach as many students as possible within the university by playing an active interdisciplinary role. My own research and educational plans match well with these objectives. I will pursue connections to the programs in physics, operations research, and computational biology. I plan to build these connections through joint supervision of students, and by organizing interdisciplinary research seminars. I will also develop a course on Computational Challenges in artificial intelligence, in which I hope to draw students from the various departments dealing with hard computational problems. Together with Joseph Halpern, I will also be actively involved in creating an interdisciplinary Cognitive Science program, drawing students from philosophy, psychology, and computer science.

Teaching, experimental computer science, and student support

I believe that an early introduction to basic research problems is a key component of the undergraduate curriculum. It provides the students with the necessary hands-on experience needed to understand both the reach and the limitations of the current methods of computer science. For example, in the Foundations of Artificial Intelligence course, I will have students use algorithm visualization techniques to explore various AI methods. The current WWW tools combined with the JAVA programming environment are ideal for such hands-on experience. An introduction to basic research problems will challenge the students to venture beyond the established material and explore non-traditional methods. Computational problems, such as game-playing and planning problems, are often used to introduce traditional AI methods, but my goal is to use these problem domains to challenge the students to explore *novel* techniques. My view of the interaction between research and undergraduate education resonates well with the position expressed by Donald Kennedy (Stanford) in his paper, "The improvement of teaching." Kennedy writes "the undergraduate program should be enriched by interaction with ongoing research at the edge of present knowledge."

AI has a long history of experimental work, often involving the design and study of large systems. I believe that such an experimental research methodology with close connections to theoretical models will be of increasing importance in computer science. In fact, the role of empirical studies may very well become quite similar to that of experimentation in the natural sciences. As is apparent from some of the main results in my work on the connections between computational complexity and statistical physics, the experimental methodology often allows us to study interesting phenomena that are beyond the scope of our rigorous mathematical models. In general, the application of the scientific method from the natural sciences opens up important new research directions in computer science and artificial intelligence. The current computer science curriculum includes little or no training in experimental methods. This situation is in sharp contrast with other disciplines, such as physics and biology, that have a well-defined, rigorous experimental methodology. One of my goals is to provide the next generation of computer scientists with a better appreciation and understanding of rigorous experimental work and its proper role in the development of new theoretical models.

At AT&T Bell Labs, I had the opportunity to supervise several students in our Cooperative Research Fellowship Program for minority students. This was a highly rewarding experience because in dealing with students that are at a disadvantage in the educational process, a mentor has a very immediate impact on the students' career. I found that the key to a successful interaction with the students was a continuing involvement with the student's educational choices, even after they left the program, simply because minority students often lack the guidance that other students receive almost continuously throughout their career. I plan to continue my involvement with AT&T's Cooperative Program and in addition provide support for minority students in the computer science program at Cornell.

Educational activities within the research community

I will continue my involvement in the education of students and researchers within the field of AI itself. I'm co-chair of the AAAI-97 Educational Forum, and will also co-chair the program in '98. The philosophy behind the Educational Forum is to give students and researcher an opportunity to learn about subfields outside of their immediate area of expertise. The Forum achieves this by presenting a broad range of tutorials on diverse topics in AI. The tutorials are given by leading researchers in the field, and provide an introduction to the area as well as a discussion of current research challenges.

Finally, I'm involved in the organization of several interdisciplinary meetings. For example, I'm chairing the '98 Symposium on AI and Operations Research. I'm also involved in the organization of the '98 Symposium on Mathematics and Artificial Intelligence, and I co-chaired Math&AI '96. In addition, I'm planning an interdisciplinary meeting on Statistical Physics, AI, and Theoretical Computer Science. I believe that such interdisciplinary meetings provide some of the best educational opportunities for future generations of computer scientists.

REFERENCES

Bacchus, F. and Kabanza, F. (1995). Using temporal logic to control search in a forward chaining planner. *Proc. EWSP-95*, 157–169.

Backstrom, C. (1992). *Computational complexity of reasoning about plans*, Ph.D. thesis, Linkoping University, Linkoping, Sweden.

Barrett, A. and Weld, D. (1994). Partial-order planning: evaluating possible efficiency gains. *Artificial Intelligence*, 67:71-112, 1994.

Blum, A. and Furst, M.L. (1995). Fast planning through planning graph analysis. *Proc. IJCAI-95*, Canada.

Boufkhad, Y. (1996) Aspects probabilistes et algorithmiques du probleme de satisfiabilite. Ph.D. Thesis, Univ. of Paris, 1996.

Brachman, R.J. and Levesque, H.J. (1985). *Readings in Knowledge Representation*, Los Altos, CA: Morgan Kaufmann, 1985.

Cheeseman, P., Kanefsky, B., Taylor, and William M. (1991). Where the really hard problems are. *Proc. IJCAI-91* (1991) 331–336.

Chvatal, V. and Szemeredi, E. (1988). Many hard examples for resolution. *J. of the ACM*, 35(4) (1988) 759–208.

Clearwater, S.H., Huberman, B.A., Hogg, T. (1991). Cooperative solution of constraint satisfaction problems. *Science*, 254 (1991).

Cook, S. and Mitchell, D. (1997). Finding hard instances of the satisfiability problem: a survey. *DIMACS Series in Discr. Math. and Theoretical Comp. Sci.*. (to appear)

Crawford, J.M. and Auton, L.D. (1993). Experimental results on the cross-over point in satisfiability problems. *AAAI-93* (1993) 21–27. (Ext. version in Artif. Intel.)

Crawford, J. and Baker, A.B. (1994). Experimental results on the application of satisfiability algorithms to scheduling problems. *Proc. AAAI-94*, Seattle, WA.

Crawford, J.M., Ginsberg, M., Luks, E., and Roy, A. (1996). *Proc. KR-96*, Boston, MA, 148–158.

CS Faculty. (1996). Cornell computer science: Vision for the next decade. White paper, Computer Science Department, Cornell University, 1996.

Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *J. of the ACM*, 7 (1960).

Dean, T. and Boddy, M. (1988) An analysis of time-dependent planning. *Proc. AAAI-88*, St. Paul, MI (1988) 49–54.

Dean, T. and Kambhampati, R. (1996). Planning and scheduling. In *CRC Handbook of Computer Science* (1996).

Dean, T., Kirman, K., Lin, S.H., Roussos, K., Weaver, M. (1996). Theory and practice in planning. Techn. report, Brown University, 1996.

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

Dechter, R. and Kask, K. (1995). GSAT and local consistency. *Proc. IJCAI-95*, Montreal, Canada (1995).

Dechter, R. and Meiri, I. (1994). Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. *Artificial Intelligence* 68 (1994) 211–241.

Dechter, R. and Rish, I. (1994). Directional resolution: the Davis-Putnam procedure revisited. *Proc. KR-94*, Bonn, Germany.

Delahaye, J.P. (1995). Les Lois de Tout ou Rien, (0-1 Laws). *Pour La Science* (French ed. of Scientific American), Vol. 213, July 1995.

de la Tour, T. and Demri, S. (1995). On the complexity of extending ground resolution with symmetry rules. *Proc. IJCAI-95*, Montreal, Canada, 289–295.

Doyle, J. and Dean, T. (1996) Strategic directions in computer science. *ACM Computing Surveys*, Vol. 28, no. 4., 1996.

Dubois, O., Andre, P., Boufkhad, Y., and Carlier, J. (1996). SAT versus UNSAT. *DIMACS Series in Discr. Math. and Theoretical Comp. Sci.*, Vol. 26, 1996, 415–433.

Ellman, T. (1993). Abstraction via approximate symmetry. In *Proceedings of IJCAI-93*, 916–921.

Ernst, M. , Millstein, T.D, and Weld, D. (1997). Automatic SAT-compilation of planning problems. *Proc. IJCAI-97*, Nagoya, Japan, 1997.

Erol, K., Nau, D.S., and Subrahmanian, V.S. (1992). On the complexity of domain-independent planning. *Proc. AAAI-92*, 381–386.

Etzioni, O. and Weld, D. S. (1994). A softbot-based interface to the internet. *Comm. ACM*, July 1994.

Fikes, R.E. and Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4), 189–208.

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

Fujita, M., Slaney, J., and Bennett, F. (1993). Automatic generation of results in finite algebra *Proc. IJCAI*, 1993.

Gent, I. and Walsh, T. (1993). Towards an understanding of hill-climbing procedures for SAT. *Proc. AAAI-93*, 28–33.

Ginsberg, M. (1993a). Dynamic Backtracking. *Journal of Artificial Intelligence*, Vol. 1, 1993, 25–46.

Ginsberg, M. (1993b). *Essentials of artificial intelligence.* Morgan Kaufmann Publ., San Francisco, CA, 1993.

Ginsberg, M. (1995). First Intl. Workshop on AI&OR. Timberline, Oregon, OR (1995).

Ginsberg, M.L. and McAllester, D.A. (1994). GSAT and dynamic backtracking. In *Proc. KR-94*, Bonn, Germany, 1994.

Gogic, G., Kautz, H., Papadimitriou, C., and Selman, B. (1995). The comparative linguistics of knowledge representation. *Proc. IJCAI-95*, Montreal, Canada, 1995.

Gomes, C.P. and Selman, B. (1997a). Problem structure in the presence of perturbations *Proc. AAAI-97*, Providence, RI, 1997.

Gomes, C.P. and Selman, B. (1997b). Algorithm portfolio design: theory vs. practice, *Proc. UAI-97*, Providence, RI, 1997.

Gomes, C.P., Selman, B., and Crato N. (1997). Heavy-tailed distributions in combinatorial search. *Proc.*

Constraint Programming '97, 1997.

Haken, A. (1985). The intractability of resolution. *Theoretical Computer Science*, Vol. 39 (1985) 297–308.

Halpern, J.Y. (1997). A logical approach to reasoning about uncertainty: a tutorial. To appear in *Discourse, Interaction, and Communication*, by X. Arrazola, K. Korta, and F.J. Pelletier (Eds.) Kluwer, 1997.

Hampson, S. and Kibler, D. (1993). Plateaus and plateau search in Boolean satisfiability problems: When to give up searching and start again. DIMACS Challenge on Satisfiability Testing, 1993.

Hansen J. and Jaumard, B. (1990). Algorithms for the maximum satisfiability problem. *Computing*, 44, 1990, 279–303.

Hayes, B. (1997). Can't Get No Satisfaction. *American Scientist*, Vol. 85, March/April 1997, pp. 108–112.

Hogg, T., Huberman, B.A., and Williams, C.P. (1996). Phase transitions in problem solving. Artificial Intelligence (special issue), Vol. 81, 1996.

Hogg, T. and Williams, C.P. (1994a). Expected gains from parallelizing constraint solving for hard problems. *AAAI-94* (1994) 331–336.

Hogg, T. and Williams, C.P. (1994b). The hardest constraint problems: a double phase transition. *Artif. Intell.*, 69 (1994) 359–377.

Hooker, J.N. (1988). Resolution vs. cutting plane solution of inference problems: Some computational experience. *Oper. Res. Letter*, 7(1) (1988) 1–7.

Horvitz, E. and Zilberstein S. (1996) (Eds.) *Proc. Flexible Computation*, AAAI Fall Symposium, Cambridge, MA, 1996.

Huberman, B.A., Lukose, R.M., and Hogg, T. (1997). An economics approach to hard computational problems. *Science*, 265, 51–54.

Jiang, Y., Kautz, K., and Selman, B. (1995). Solving problems with hard and soft constraints using a stochastic algorithm for MAX-SAT. AI/OR Wrksh., OR (1995).

Johnson, D. (1996). Experimental analysis of algorithms: The good, the bad, and the ugly. Invited Lecture, *AAAI-96*, Portland, OR.

Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. (1991) Optimization by simulated annealing: an experimental evaluation; part II. *Oper. Res.*, 39 (1991).

Joslin, D. and Pollack, M. (1995). Passive and active decision postponement in plan generation. In the *European Workshop on Planning (EWSP)*, Assisi, Italy, Sept. 1995.

Kambhampati, S. (1996). Refinement planning: status and prospectus. *Proc. AAAI-96*, Portland, OR., 1996, 1331–1336. (Invited Lecture)

Kambhampati, S. (1997). Can we bridge refinement-based and SAT-based planning techniques? *Proc. IJCAI-97*, Nagoya, Japan, 1997.

Kasparov, G. (1997). The day that I sensed a new kind of intelligence. *Time*, May 26, 1997, 66–67. See also *Time*, March 25, 1996, p. 55.

Kautz, H., Kearns, M., and Selman, B. (1995). Horn approximations of empirical data. *Artificial Intelligence*, Vol. 74, 1995, 129-145.

Kautz, H., McAllester, D., and Selman, B. (1996). Encoding plans in propositional logic. *Proceedings of the Fifth International Conference on Knowledge Representation and Reasoning (KR-96)*, Cambridge, MA, 1996, 374–384.

Kautz, H. and Selman, B. (1992). Planning as satisfiability. *ECAI-92*, Vienna, Austria, 1992, 359–363.

Kautz, H. and Selman, B. (1994) An empirical evaluation of knowledge compilation. *Proc. AAAI-94*, Seattle, WA, 1994, 155–161.

Kautz, H. and Selman, B. (1996). Pushing the envelope: planning, propositional logic, and stochastic search. *Proc. AAAI-96*, Portland, OR, 1996, 1194–1201.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 1983, 671–680.

Kirkpatrick, S. and Selman, B. (1994). Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264 (May 1994) 1297–1301. Also p. 1249: "Mathematical Logic: Pinning Down a Treacherous Border in Logical Statements" by B. Cipra.

Kolata, G. (1996). Computer math proof shows reasoning power. *New York Times*, Dec. 10, 1996.

Konolige, K. (1994). Easy to be hard: Difficult problems for greedy algorithms. *KR-94*, Bonn, Germany, 1994, 374–378.

Korf, R. (1991). Search. *Encyclopedia of Artificial Intelligence* John Wiley, New York (1991).

Korf, R. (1993). Linear-space best-first search. *Artificial Intelligence*, Vol. 62(1), 41–78.

Kosslyn, S.M. (1981). The medium and message in mental imagery: a theory. In *Imagery* by N. Block (Ed.), Cambridge MA: Bradford/MIT Press, 1981, 207–244.

Larrabee, T. (1992). Efficient generation of test patterns using Boolean satisfiability. *IEEE Trans. on CAD*, Vol. 11, 1992, pp 4-15.

Larkin, J.H. and Simon, H.A. (1987). Why a picture is (sometimes) worth ten thousand Words. *Cognitive Science*, 1987.

Levesque, H.J. (1986). Making believers out of computers. *Artificial Intelligence*, Vol. 30, 1986, 81–108.

Levesque, H.J. and Brachman, R.J. (1985). A Fundamental Tradeoff in Knowledge Representation and Reasoning (Revised Version). In *Readings in Knowledge Representation* by R.J. Brachman and H.J. Levesque (Eds.), Los Altos, CA: Morgan Kaufmann, 1985, 41–70.

Mandelbrot, B. (1983). *The fractal geometry of nature.* Freeman: New York. 1983.

Mazure, B., Sais, L., and Gregoire E. (1996). Boosting complete techniques thanks to local search methods. *Proc. Math & AI*, 1996.

McAllester, D., Selman, B., and Kautz, H. (1997). Evidence for invariants in local search. *Proc. AAAI-97*, Providence, RI, 1997.

McCune, W. (1996). Solution of the Robbins problem. Draft, 1996.

McDermott, D. (1997). Yes, Computers can think. *New York Times*, May 14, 1997. See ftp://ftp.cs.yale.edu/pub/mcdermott/papers/deepblue.txt.

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

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

Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., and Troyansky, L. (1996). Phase Transition and Search Cost in the 2+p-SAT Problem. *Proceedings of PhysComp-96*, Boston, MA, 1996.

Newell, A. and Simon, H.A. (1971). *Human problem solving.* Prentice-Hall, Englewood Cliffs, NJ.

Nilsson, N.J. (1986) Probabilistic logic. *Artificial Intelligence*, Vol. 28(1), 1986.

Nilsson, N.J. (1998) Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publ., San Mateo, CA, (to appear).

Papadimitriou, C.H. (1993). *Computational Complexity.* Addison Wesley, 1993.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible Inference.* Morgan Kaufmann Publ., San Mateo, CA, 1988.

Russell, S. and Norvig P. (1995). *Artificial Intelligence a Modern Approach.* Prentice Hall, Englewood Cliffs, NJ, 1995.

Russell, S. and Subramanian, D. (1993). Provably bounded optimal agents. *Proc. IJCAI-93*, Chambery, France, 1993.

Selman, B. (1993). Non-systematic search methods for model finding. *Proc. of the 1993 IEEE Intl. Conf. on Tools with AI*, Boston, MA, 1993. (Invited position paper.)

Selman, B. (1994a). Near-optimal plans, tractability, and reactivity. *Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning (KR-94)*, Bonn, Germany, 1994, 521–529.

Selman, B. (1994b). Domain-specific complexity tradeoffs. *Proc. of the Twelfth European Conference on Artificial Intelligence (ECAI-94)*, Amsterdam, The Netherlands.

Selman, B. (1995a). Stochastic search and phase transitions: AI meets physics. *Proc. IJCAI-95*, 1995. (Invited lecture.)

Selman, Bart. (1995b). Randomized search strategies for model finding. *Proc. IJCAI-95*, Montreal, Canada, 1995. (Invited position statement.)

Selman, B. (1996). Computational challenges in artificial intelligence. *ACM Computing Surveys*, 28(4es), 1996.

Selman, B. (1997a). Stochastic Search. *Artificial Intelligence Magazine.* (Invited survey; to appear.)

Selman, B. (1997b). Greedy local search. In *MIT Encyclopedia of the Cognitive Sciences (MITECS)*, Cambridge: MIT Press. (to appear.)

Selman, B. (1997c). Analogical Representations. In *Cognition and Knowledge Representation*, Z. Pylyshyn (Ed.). (to appear)

Selman, B., Brooks, R., Dean, T., and Horvitz, E., Mitchell, T., and Nilsson, N. (1996) Challenge problems for artificial intelligence. *Proc. AAAI-96*, Portland, OR, 1996, 1340–1345.

Selman, B. and Kautz, H.A. (1993a). An empirical study of greedy local search strategies for satisfiability testing. *Proc. AAAI-93,* Wash., DC (1993) 46–51.

Selman, B. and Kautz, H.A. (1993b). Domain-independent extensions to GSAT: Solving large structured satisfiability problems. *IJCAI-93,* France (1993) 290–295.

Selman, B. and Kautz, H. (1996). Knowledge compilation and theory approximation. *Journal of the ACM*, Vol. 43, No. 2, 1996, 193–224.

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

Selman, B., Kautz, H., and Cohen, B. (1996). Local search strategies for satisfiability testing. *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 26, 1996, 521–532.

Selman, B., Kautz, H., and McAllester, D. (1997) Computational challenges in propositional reasoning and search. *Proc. IJCAI-97*, Nagoya, Japan, 1997.

Selman, B. and Kirkpatrick, S. (1996). Finite-size scaling of the computational cost of systematic search. *Artificial Intelligence*, Vol. 81, 1996, 273–295.

Selman, B., and Levesque, H. (1996). Support set selection for abductive and default reasoning. *Artificial Intelligence*, Vol. 82, 1996, 259–272.

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

Selman, B., Mitchell, D., and Levesque, H. (1996). Generating hard satisfiability problems. *Artificial Intelligence*, Vol. 81, 1996, 17–29.

Smith, B. and Grant S.A. (1996). Sparse constraint graphs and exceptionally hard problems. *IJCAI-95*, 646–651, 1995. Full version in AIJ (Hogg et al. 1996).

Sloman, A. (1971). Interactions between philosophy and AI — the role of intuition and non-logical reasoning in intelligence. *Artificial Intelligence*, **2**, 1971, 209–225.

Smith, B. and Dyer, M. (1996). Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence*, 81, 1996.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM,* March 1995, Vol. 38, No. 3.

Trick, M. and Johnson, D. (Eds.) (1996) Cliques, Coloring and Satisfiability. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 26. For benchmarks, see http://dimacs.rutgers.edu/Challenges/index.html.

van Hentenryck, P., Deville, Y., and Teng Choh-Man (1992) A generic arc consistency algorithm and its specializations. *Artificial Intelligence*, 57, 1992.

Williams, B.C. and Nayak, P. (1996) A model-based approach to reactive self-configuring systems. *Proceedings AAAI-96*, Portland, OR. 971–978.

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