Supermodels and Robustness

Matthew L. Ginsberg
CIRL
1269 University of Oregon
Eugene, OR 97403-1269
ginsberg@cirl.uoregon.edu

Andrew J. Parkes
CIRL and CIS Dept.
1269 University of Oregon
Eugene, OR 97403-1269
parkes@cirl.uoregon.edu

Amitabha Roy
CIS Dept.
University of Oregon
Eugene, OR 97403
aroy@cs.uoregon.edu

Abstract

When search techniques are used to solve a practical problem, the solution produced is often brittle in the sense that small execution difficulties can have an arbitrarily large effect on the viability of the solution. The AI community has responded to this difficulty by investigating the development of "robust problem solvers" that are intended to be proof against this difficulty.

We argue that robustness is best cast not as a property of the problem solver, but as a property of the solution. We introduce a new class of models for a logical theory, called *supermodels*, that captures this idea. Supermodels guarantee that the model in question is robust, and allow us to quantify the degree to which it is so.

We investigate the theoretical properties of supermodels, showing that finding supermodels is typically of the same theoretical complexity as finding models. We provide a general way to modify a logical theory so that a model of the modified theory is a supermodel of the original. Experimentally, we show that the supermodel problem exhibits phase transition behavior similar to that found in other satisfiability work.

Introduction

In many combinatorial optimization or decision problems our initial concern is to find solutions of minimal cost, for example, a schedule with a minimal overall length. In practice, however, such optimal solutions can be very brittle. If anything out of our control goes wrong (call this a "breakage"), repairing the schedule might lead to a great increase in its final cost. If breakages are sufficiently common, we might well do better on average to use a suboptimal solution that is more robust. The difficulty with trading optimality for robustness is that robustness is difficult to quantify, and especially difficult to quantify in a practical fashion.

In building definitions that are useful for quantifying robustness, we need to be aware of the requirements of both the users and the producers of robust solutions.

A user of robust solutions might be motivated by two distinct demands on the possibilities for repair:

1. Fast repair: A small set of changes must be repairable in polynomial time.
2. Small repair: The repaired solution should be close to the original model. In other words, it must be possible to repair a small set of changes with another small set of changes.

The condition of fast repair arises, for example, when something goes wrong in a production line and halting the line to perform exponential search might be far too costly. Demanding that small flaws can be addressed with small repairs is also common. A production line schedule might involve many people, each with a different list of tasks for the day. Constantly changing everyone's task list is likely to lead to far too much confusion. The ability to repair flaws with a small number of changes is a goal in itself, independent of the fact that this means repair is also likely to be fast.

As a producer of robust solutions, it might well be helpful if the measure of robustness were independent of the repair algorithm. An algorithm-independent characterization of robustness is useful not only because of its greater simplicity, but because it might support the use of intelligent search methods to find solutions with guaranteed levels of robustness. In contrast, algorithm-dependent notions of robustness imply that the search for robust solutions is likely to reduce to generate and test. This is because partial solutions might not carry enough information to determine whether the repair algorithm will succeed. For example, if we were to use local search for repair, it is already difficult to characterize the repairability of full solutions. Deciding whether a partial solution will extend to a repairable full solution might well be completely impractical. We are not implying that algorithm independence is essential, only that it might be very useful in practice.

This paper introduces the concept of *supermodels* as models that measure inherent degrees of robustness. In essence, a supermodel provides a simple way to capture the requirement that "for all small breakages there exists a small repair;" that repairs are also fast will be seen to follow from this. The supermodel definition also has the advantage that the robustness is inherently a property of the supermodel, and does not rely on assumptions about repair algorithms.

We will also see that despite the simplicity of the supermodel concept, there appears to be a surprisingly rich associated theory. Most importantly, there are many different interrelated classes of supermodel characterized by the amounts of breakage and repair that are allowed. This richness of structure with various degrees of robustness allows us to propose a framework under which robustness can be quantified, thereby supporting an informed tradeoff between optimality and robustness.

The first sections in the paper define supermodels and explore some theoretical consequences of the definition. For satisfiability, finding supermodels is in NP, the same complexity class as that of finding models. We give an encoding that allows us to find a particular kind of supermodel for SAT using standard solvers for SAT. Using this encoding, we explore the existence of supermodels in Random 3SAT, finding evidence for the existence of a phase transition, along with the standard easy-hard-easy transition in search cost.

Overall, the supermodel concept makes the task of finding robust solutions similar to that of finding solutions, rather than necessarily requiring special-purpose search technology of its own.

Supermodel Definitions

A first notion of solutions that are inherently robust to small changes can be captured as follows:

> **Definition 1:** An (a, b)-supermodel is a model such that if we modify the values taken by the variables in a set of size at most a (breakage), another model can be obtained by modifying the values of the variables in a disjoint set of size at most b (repair).

The case $a = 0$ means that we never have to handle any breakages, and so all models are also $(0, b)$-supermodels. A less trivial example is a $(1, 1)$-supermodel: This is a model that guarantees that if any single variable's value is changed, then we can recover a model by changing the value of at most one other variable.

We will typically take a and b to be small, directly quantifying the requirement for small repairs. Definition 1 also has a variety of attractive properties. Firstly, if finding models is in NP, and if a is taken to be a constant (independent of the problem size n) then finding (a, b)-supermodels is also in NP. This is because the number of possible breakages is polynomial $O(n^a)$. Secondly, if b is a constant, finding the repair is possible in polynomial time, since there are only $O(n^b)$ possible repairs. These observations are independent of the method used to make the repairs, depending only on the bounded size of the set of possible repairs.

We thus see that with a and b small constants, (a, b)-supermodels quantify our conditions for robustness and do so without worsening the complexity class of the problem (assuming we start in NP or worse).

In practice, the definition needs to be modified because not all variables are on an equal footing. We might not be able to account for some variables changing their value: some breakages might simply be irreparable, while others might be either very unlikely or impossible, and so not worth preparing for. To account for this, we use a "breakage set" that is a subset of the set of all variables, and will only attempt to guarantee robustness against changes of these variables. Similarly, repairs are likely to be constrained in the variables they can change; as an example, it is obviously impossible to modify a variable that refers to an action taken in the past. We therefore introduce a similar "repair set" of variables. We extend Definition 1 to

> **Definition 2:** An (S_1^a, S_2^b)-supermodel is a model such that if we modify the values taken by the variables in a subset of S_1 of size at most a (breakage), another model can be obtained by modifying the values of the variables in a disjoint subset of S_2 of size at most b (repair).

It is clear that an (a, b)-supermodel is simply a (S_1^a, S_2^b)-supermodel in which the breakage and repair sets are unrestricted. We will use the term "supermodel" as a generic term for any (S_1^a, S_2^b)- or (a, b)-supermodel.

Different degrees of robustness correspond to variation in the parameters S_1, S_2, a and b. As we increase the size of the breakage set S_1 or the number of breaks a, the supermodels become increasingly robust. Robustness also increases if we decrease the size of the repair set S_2 or number of repairs b. Supermodels give us a flexible method of returning solutions with certificates of differing but guaranteed robustness. As an example,[1] consider the simple theory $p \vee q$. Any of the three models (p, q), $(\neg p, q)$ and $(p, \neg q)$ is a $(1, 1)$-supermodel. Only the first model, however, is a $(1, 0)$-supermodel. The supermodel ideas correctly identify (p, q) as the most robust model of $p \vee q$.

[1]for which we would like to thank Tania Bedrax-Weiss.

Theory

Let us now restrict our discussion to the case of propositional theories, so that breakage and repair will correspond to flipping the values of variables in the model from true to false or vice versa. We also focus on (a, b)-supermodels as opposed to the more general (S_1^a, S_2^b)-supermodels, and so breakage or repair might involve any variable of the theory.

We shall say that a theory Γ belongs to the class of theories $\mathrm{SUPSAT}(a, b)$ if and only if Γ has an (a, b)-supermodel. We will also use $\mathrm{SUPSAT}(a, b)$ to refer to the associated decision problem:

$\mathrm{SUPSAT}(a, b)$

Instance: A clausal theory Γ
Question: Is $\Gamma \in \mathrm{SUPSAT}(a, b)$?

We first prove that $\mathrm{SUPSAT}(a, b)$ is in NP for any constants a, b. Given an instance of a theory Γ with n variables, a nondeterministic Turing machine guesses a model and a table of which variables to repair for each set of variables flipped. The table has at most n^a entries, one for each possible possible breakage, and each entry is a list of at most b variables specifying the repair. It is obviously possible to check in polynomial time whether the assignment is a model and that all the repairs do indeed work.

In principle, a supermodel-finding algorithm could produce such a table as output, storing in advance all possible repair tuples. This would take polynomial space $O(n^a b)$ and reduce the time needed to find the repair to be a *constant*, $O(a)$. In practice, however, usage of $O(n^a b)$ memory is likely to be prohibitive.

We also have:

Theorem: $\mathrm{SUPSAT}(1, 1)$ is NP-hard.

Proof: We reduce SAT to $\mathrm{SUPSAT}(1, 1)$.

Let the clausal theory $\Gamma = C_1 \wedge C_2 \ldots \wedge C_m$ over n variables $V = \{x_1 \ldots x_n\}$ be an instance of SAT. We construct an instance of $\mathrm{SUPSAT}(1, 1)$ as follows: construct the theory Γ' over $n + 1$ variables $V' = \{x_1, x_2 \ldots x_n, \alpha\}$ where

$$\Gamma' = (C_1 \vee \alpha) \wedge (C_2 \vee \alpha) \ldots (C_m \vee \alpha)$$

and α is a new variable not appearing in Γ. We prove that Γ has a model iff Γ' has a $(1, 1)$-supermodel.

Suppose Γ had a model m. We construct a model for Γ' which will be a $(1, 1)$-supermodel. Extend the assignment m to an assignment of Γ' by setting α to false. Clearly this assignment satisfies all clauses of Γ'. Suppose now we flip the value of a variable in V'. If we flip the value of some variable in $\{x_1 \ldots x_n\}$, we can repair it by setting α = true. If instead we flip the value of α from false to true, no repair is needed. Hence this assignment is indeed a model of Γ' such that on flipping the value of 1 variable, at most 1 repair is needed. Hence $\Gamma' \in \mathrm{SUPSAT}(1, 1)$.

Next, suppose $\Gamma' \in \mathrm{SUPSAT}(1, 1)$. By definition, it has a model m. If α is false in m observe that the restriction of m to $V = \{x_1, x_2 \ldots x_n\}$ is a model of Γ. If α is true, flip it to false. Since $\Gamma' \in \mathrm{SUPSAT}(1, 1)$ we can repair it by flipping some other variable to get a model where α remains false. Restricting the repaired model to $V = \{x_1, x_2 \ldots x_n\}$ once again gives us a model for Γ. Thus Γ is satisfiable. **QED**.

It follows immediately from the definition that

$$\mathrm{SUPSAT}(a, b) \subseteq \mathrm{SUPSAT}(a, b + 1) \quad (1)$$

and

$$\mathrm{SUPSAT}(a + 1, b) \subseteq \mathrm{SUPSAT}(a, b) \quad (2)$$

since b repairs suffice for up to $a + 1$ breaks, b repairs suffice for up to a breaks. In many cases we can prove that the inclusions in the above supermodel hierarchy are strict.

It is easy to show that the inclusion (1) is strict. i.e. $\mathrm{SUPSAT}(a, b) \neq \mathrm{SUPSAT}(a, b + 1)$. For example the following theory which consists of a chain of $b + 2$ variables,

$$(x_1 \to x_2) \wedge (x_2 \to x_3) \ldots (x_{b+1} \to x_{b+2}) \wedge (x_{b+2} \to x_1)$$

belongs to $\mathrm{SUPSAT}(a, b+1) - \mathrm{SUPSAT}(a, b)$. The only models of this theory are those with all $b + 2$ variables set to true or with all of them set to false. For any set of a flips, we need at most $b + 1$ repairs, hence this theory is in $\mathrm{SUPSAT}(a, b + 1)$. If one variable value is flipped, we need exactly $b + 1$ repairs. Since b repairs do not suffice, this theory is not in $\mathrm{SUPSAT}(a, b)$.

Using multiple chains and similar arguments, one can prove that the inclusion in (2) is strict whenever $a \leq b$. In general, however, the question of whether or not (2) is strict for all a, b is open.

Equations (1) and (2) induce a hierarchical structure on the set of all satisfiable theories. This gives a rich set of relative "strengths" of robustness with a fairly strong partial order among them.

Finding (1,1)-supermodels

We have shown the task of finding (a,b)-supermodels of a SAT problem to be in NP. It should therefore be possible to encode the supermodel requirements on a theory Γ as a new SAT CNF instance Γ_{SM} that is at most polynomially larger than Γ. In this section, we do this explicitly for (1,1)-supermodels in SAT, so that a model for Γ_{SM} has the property that if we are forced to flip any variable i there is another variable j that we can flip in order to recover a model of Γ. In other words, we will show how to construct Γ_{SM} such that Γ has a (1,1)-supermodel if and only if Γ_{SM} has a model. A model of Γ_{SM} will be a supermodel of Γ.

We are working with CNF so $\Gamma = \wedge_a C_a$ is a conjunction of clauses C_a. The basic idea underlying the encoding is to allow the assignment to remain fixed, instead flipping the variables as they appear in the theory.

Thus let Γ_i denote Γ in which all occurrences of variable i have been flipped, and let Γ_{ij} denote Γ in which all occurrences of the variables i and j have been flipped. Denoting the clauses with flipped variables similarly, $\Gamma_i = \wedge_a C_{ai}$ and $\Gamma_{ij} = \wedge_a C_{aij}$.

Now for a model to be a (1,1)-supermodel, if we flip a variable i, one of two conditions must hold. Either the model must be a model of the flipped theory, or there must be some different variable j for which the current model is a model of the doubly flipped theory. Hence, we must enforce

$$\forall i.\ \ \Gamma_i \vee (\ \exists j.\ \ j \neq i \wedge \Gamma_{ij}) \qquad (3)$$

Converting this to CNF by direct expansion would result in an exponential increase in size, and we therefore introduce new variables c and y that reify the flipped clauses and theories:

$$\begin{aligned} c_{ai} &\longleftrightarrow C_{ai} \\ c_{aij} &\longleftrightarrow C_{aij} \\ y_i &\longleftrightarrow \Gamma_i \\ y_{ij} &\longleftrightarrow \Gamma_{ij} \end{aligned}$$

These definitions are easily converted to a CNF formula Γ_{defs} via

$$\begin{aligned} \neg y_i &= \vee_a \neg c_{ai} \\ \neg y_{ij} &= \vee_a \neg c_{aij} \end{aligned}$$

The supermodel constraint (3) is now

$$\wedge_i\,(\neg y_i \vee \vee_{j \neq i} \neg y_{ij})$$

which is correctly in CNF. The complete encoding is

$$\Gamma_{SM} = \wedge_i\,(\neg y_i \vee \vee_{j \neq i} \neg y_{ij}) \wedge \Gamma \wedge \Gamma_{\mathrm{defs}} \qquad (4)$$

If the original Γ had n variables and m clauses of length at most k then Γ_{SM} has $O(mn^2)$ variables, and $O(mn^2k)$ clauses of length at most $O(n)$.

As an example, consider once again the trivial theory $p \vee q$. The only clause is $C_0 = p \vee q$.

Flipping p, we get $C_{0p} = \neg p \vee q$. Flipping both gives $C_{0pq} = \neg p \vee \neg q$. For the defined variables, we have

$$c_{0p} \leftrightarrow (\neg p \vee q)$$

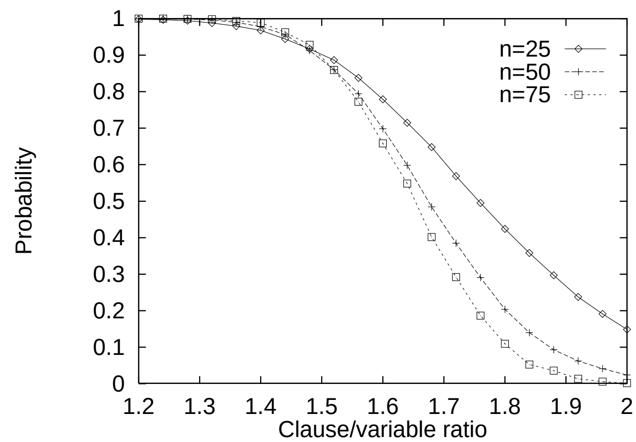

Figure 1: Probability of a Random 3SAT instance having a (1,1)-supermodel.

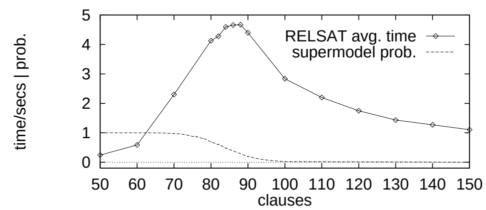

Figure 2: Easy-hard-easy transition at n=50. Time is for `relsat(4)` (Bayardo & Schrag 1997). For comparison we give the probability of finding a supermodel.

and similarly, together with $\neg y_p = \neg c_{0p}$ and similarly. The complete theory Γ_{SM} can now be constructed using (4). Note also that the general construction can easily be extended to (S_1^1, S_2^1)-supermodels simply by restricting the allowed subscripts in the c_{ai} and y_i.

Restricting a model of Γ_{SM} to the original variables will produce a (1,1)-supermodel of Γ. Since Γ_{SM} is just another SAT-CNF problem, we can solve it using a standard SAT solver. This solver itself need not know anything about supermodels, and can apply intelligent search techniques that are likely to be significantly more efficient than would be the case if were were to test for robustness in retrospect.

Phase Transitions

Phase transition, or “threshold”, phenomena are believed to be important to the practical matter of finding solutions (Huberman & Hogg 1987, and others). This is in part because of the similarities to optimization: As we change the system, we change from many to relatively few to no solutions, and the cost of finding solutions simultaneously changes from easy to hard to easy again. The average difficulty peaks in the phase transition region, matching the intuition about finding

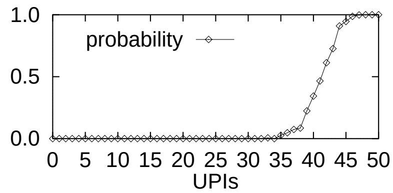

Figure 3: Probability of residual theory having a supermodel, as a function of number of UPIs. Instances are from the phase transition for satisfiability at n=50.

optimal solutions.

In this section, we briefly study the issue of supermodels and phase transitions for the case of Random 3SAT (Mitchell, Selman, & Levesque 1992). Instances are characterized by n variables, m clauses, and a clause/variable ratio $\alpha = m/n$. There is strong evidence that for large n, this system exhibits a phase transition at $\alpha \approx 4.2$ (Crawford & Auton 1996). Below this value, theories are almost always satisfiable; above it, they are almost always unsatisfiable.

We first consider whether or not $\text{SUPSAT}(a, b)$ has similar phase transitions. Using the encoding of the previous section, we studied the empirical probability of Random 3SAT instances having a (1,1)-supermodel. Figure 1 gives the results, leading us to expect a phase transition at $\alpha \approx 1.5$. The apparent $\text{SUPSAT}(1, 1)$ transition thus occurs for theories that are very underconstrained. As seen in Figure 2, the time needed to solve the instances undergoes the usual easy-hard-easy transition.

Consider next the possible existence of supermodels at the satisfiability phase transition itself, $\alpha \approx 4.2$. We have just seen that there will almost certainly be no $(1, 1)$-supermodels at this phase transition. We also know that as we approach the transition, the number of prime implicates of the associated theory increases (Schrag & Crawford 1996), until at the transition itself, we have many instances with large numbers of unary prime implicates (UPIs) (Parkes 1997). Any model must respect these UPIs: if a variable in a UPI is changed then no repair is possible. Hence, any variables in UPIs must be excluded from the breakage set. Since flipping the value of a UPI can never be involved in a repair, these variables can also be excluded from the repair set. The simplest choice is thus to look for (S_1^a, S_2^b)-supermodels with

$$S_1 = S_2 = R = (V - \{v | v \text{ or } \neg v \text{ is a UPI}\})$$

Parkes called this set R the *residual variables* of the instance. Looking for an (R^a, R^b)-supermodel is equivalent to looking for an (a,b)-supermodel of the residual theory, which consists of the constraints remaining on the residual variables after accounting for the UPIs. Figure 3 shows that the residual theories tend to have (1,1)-supermodels in the so-called *single cluster instances*, instances with at least 80% UPIs. For $n = 50$, 80% or more of the variables appear in UPIs some 37% of the time.

The above experiments reflect two extremes. Demanding full (1,1)-supermodels forced us into a very underconstrained region. Conversely, instances from the critically constrained region having many UPIs can still have (S_1^a, S_2^b)-supermodels. In practice, it seems likely that realistic problems will require solutions with intermediate levels of robustness: not so robust as to be able to cater to any possible difficulty, but sufficiently robust as to require some sacrifice in optimality. The framework we have described allows to quantify the tradeoff between robustness and solution cost precisely.

Related Work

Since robustness has generally been viewed as a property of the solution *engine* as opposed to a property of the *solutions*, there has been little work on the development of robust solutions to AI problems. Perhaps the most relevant approach has been the attempt to use optimal Markov Decision Processes (MDPs) to find solutions that can recover from likely execution difficulties.

Unfortunately, it appears[2] that the cost of using MDPs to achieve robustness is extreme, in the sense that it is impractical with current technology to solve problems of interesting size. This is to be contrasted to our approach, where the apparent existence of a phase transition suggests that it will be practical to find near-optimal supermodels for problems of practical interest.

Of course, the supermodel approach is solving a substantially easier problem than is the MDP community. We do not (and at this point cannot) consider the differing likelihoods of various failures; a possible breakage is either in the set S_1 or it isn't. We also have no general framework for measuring the probabilistic cost of a solution; we simply require a certain degree of robustness and can then produce solutions that are optimal or nearly so given that requirement. On the other hand, our technology is capable of solving far larger problems and can be applied in any area where satisfiability techniques are applicable, as opposed to the currently restricted domains of applicability of MDPs (planning and scheduling problems, essentially).

A changing environment might also be modeled as a Dynamic Constraint Satisfaction Problem (DCSP) (Dechter & Dechter 1988); what we have called a "break" could instead be viewed as the dynamic addi-

[2] Steve Hanks, personal communication

tion of a unary constraint to the existing theory. The work in DCSPs aiming to prevent the solutions changing wildly from one CSP to the next (e. g. (Verfaillie & Schiex 1994, and others)) has similar motivations to our requirement for "small repairs", but DCSPs do not supply a way to select solutions to the existing constraints. Supermodels allow us to select the solutions themselves so as to partially guard against future changes of constraints requiring large changes to the solution. Conversely, DCSPs can handle changes that are more general than just one set of unary constraints, although we expect that the supermodel idea can be generalized in this direction.

MIXED-CSPs (Fargier, Lang, & Schiex 1996) allow variables to be controllable (e. g. our flight departure time) or uncontrollable (e. g. the weather). A system is consistent iff any allowed set of values for the uncontrollable variables can be extended to a solution by valuing the controllable variables appropriately. While this has some flavor of preserving the existence of models in the presence of other changes, MIXED-CSPs do not require that the model change be small: No attempt is made to select a model so that nearby worlds have nearby models. On a technical level, this is reflected in the MIXED-CSP consistency check being Π_2^P-complete as opposed to NP-complete for supermodels. Our observations about the phase transitions and reductions to SAT also give us significant practical advantages that are not shared by the MIXED-CSP approach.

Conclusions

This paper relies on two fundamental and linked observations. First, robustness should be a property not of the techniques used to solve a problem, but of the solutions those techniques produce. Second, the operational need for solutions that can be modified slightly to recover from small changes in the external environment subsumes the need for solutions for which the repairs can be found quickly. *Supermodels* are a generalization of the existing notion of a model of a logical theory that capture this idea of robustness and that allow us to quantify it precisely.

While the definition of a supermodel is simple, the associated mathematical structure appears to be fairly rich. There is a hierarchy of supermodels corresponding to varying degrees of robustness. Searching for a supermodel is of the same theoretical complexity as solving the original problem, and the experiments on finding supermodels bear this out, revealing a phase transition in the existence of supermodels that is associated with the usual easy-hard-easy transition in terms of computational expense.

Experimental results suggest that finding fully robust supermodels will in general involve substantial cost in terms of the quality of the overall solution. This can be dealt with by considering supermodels that are robust against a limited set of external changes, and we can quantify the expected cost of finding such supermodels as a function of the set of contingencies against which one must guard.

Acknowledgements

This work has been supported by the Air Force Research Laboratory and by the Defense Advanced Research Projects Agency under contracts F30602-95-1-0023 and F30602-97-1-0294. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, or the U.S. Government.

The authors would like to thank the members of CIRL for many invaluable discussions related to the ideas we have presented.

References

Bayardo, R. J., and Schrag, R. C. 1997. Using CSP look-back techniques to solve real-world SAT instances. In *Proc. of AAAI–97*, 203–208.

Crawford, J. M., and Auton, L. D. 1996. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence* 81:31–57.

Dechter, R., and Dechter, A. 1988. Belief maintenance in dynamic constraint networks. In *Proc. of AAAI-88*, 37–42.

Fargier, H.; Lang, J.; and Schiex, T. 1996. Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge. In *Proc. of AAAI–96*, 175–180.

Huberman, B. A., and Hogg, T. 1987. Phase transitions in artificial intelligence systems. *Artificial Intelligence* 33:155–171.

Mitchell, D.; Selman, B.; and Levesque, H. J. 1992. Hard and easy distributions of SAT problems. In *Proc. of AAAI–92*, 459–465.

Parkes, A. J. 1997. Clustering at the phase transition. In *Proc. of AAAI–97*, 340–345.

Schrag, R., and Crawford, J. M. 1996. Implicates and prime implicates in Random 3SAT. *Artificial Intelligence* 88:199–222.

Verfaillie, G., and Schiex, T. 1994. Solution reuse in Dynamic Constraint Satisfaction Problems. In *Proc. of AAAI–94*, 307–312.