The Backdoor Key: A Path to Understanding Problem Hardness

Yongshao Ruan, Henry Kautz
Department of Computer Science
University of Washington
Seattle, WA 98195
ruan@cs.washington.edu
kautz@cs.washington.edu

Eric Horvitz
Microsoft Research
Redmond, WA 98052
horvitz@microsoft.com

Abstract

We introduce our work on the *backdoor key*, a concept that shows promise for characterizing problem hardness in backtracking search algorithms. The general notion of *backdoors* was recently introduced to explain the source of heavy-tailed behaviors in backtracking algorithms (Williams, Gomes, & Selman 2003a; 2003b). We describe empirical studies that show that the *key faction,i.e.*, the ratio of the key size to the corresponding backdoor size, is a good predictor of problem hardness of ensembles and individual instances within an ensemble for structure domains with large key fraction.

Introduction

Propositional reasoning (SAT) is the best-known NP-complete problem in computer science. Even though a class of SAT problems may be intractable in the worst case, most of its instances may still be polynomial-time solvable in practice and their hardness may vary significantly. It has proven extremely difficult in practice to define realistic problem ensembles where instances do not vary widely in hardness (Kautz *et al.* 2001). Different instances have widely varying hardness even in the well-known random 3-SAT problem ensembles with fixed clause-to-variable ratios (Kirkpatrick & Selman 1994). In such cases, it is crucial to understand the hidden structures that can be used to identify problem hardness and as points of attack for search heuristics.

There has been considerable research interest in identifying such structures for many years. One example of such hidden structure is the *backdoor* set (Williams, Gomes, & Selman 2003a). Given a SAT formula, a set of variables forms a *backdoor* for a problem instance if there is a truth assignment to these variables such that the simplified formula can be solved in polynomial time via use of the propagation and simplification mechanism of the SAT solver under consideration.

The original motivation of the backdoor concept was to explain the heavy-tailed behavior of backtracking search algorithms (Williams, Gomes, & Selman 2003b). Williams *et al.* went on to describe a variety of simple randomized “guessing” algorithms for solving problems with small backdoor sets, and empirically showed than many interesting benchmark SAT problems have surprisingly small backdoors (on the order of a few dozen variables). These algorithms are best-case exponential in the size of the backdoor, and thus are practical for problems with backdoors up to about size 20. For problems with larger backdoors — which do indeed appear to be the vast majority of satisfiability problems of interest — the best complete algorithms remain ones based on Davis-Putnam-Loveland-Logemann style backtracking search. Therefore it is important to investigate the precise connection between the backdoor properties of an problem instance or ensemble and problem hardness as measured by solution time using backtracking search.

Intuitively, the difficulty of identifying a backdoor set and setting a correct assignment to the backdoor variables by a backtracking solver is positively correlated to problem hardness because once the backdoor set is assigned correctly, the remaining problem becomes trivial. However, as we shall describe below, problem hardness is not a simple function of backdoor size. In particular, the backdoor size does not capture the *dependencies* among backdoor variables. We introduce the concepts of *backdoor key* variables to capture such dependencies. A *backdoor key variable* is a backdoor variable whose value is logically determined by settings of other backdoor variables. We define the *key fraction* of a problem as the ratio of the key size to the corresponding backdoor size, and investigate its relation with instance hardness and ensemble hardness. *Instance hardness* is the hardness of an individual instance, which we take as the median search cost of a set of runs of a search algorithm applied to the given instance. *Ensemble hardness* is the median hardness of an ensemble of instances. Empirical results demonstrate that key fractions are good predictors for both ensemble hardness and instance hardness for structure domains with large key fractions.

Previous Work on Hidden Structure

We first summarize prior work on identifying hidden structures that are linked to problem hardness. Next, we describe two structure benchmark domains used in our empirical studies.

Hidden Structures for Problem Hardness

An important step in understanding problem hardness is to relate hardness with observations of phase transitions in problem distributions as the underlying parameters are var-

Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

ied. Many NP-complete problems like satisfiability and constraint satisfaction display phase transition behavior in solubility and the easy-hard-easy pattern of search cost associating with increasing constrainedness. Problems that are under-constrained tend to have many solutions and thus generally easy to solve due to the high probability of guessing a solution. On the other hand, problems that are over-constrained tend to have no solution. These problems are also generally easy to solve because there are many constraints and thus many possible solutions can be pruned efficiently. Hard problems are usually located at a critical-constrained point, where roughly half the instances are satisfiable and half the instances are unsatisfiable. One well-know example is the random 3-SAT problem with varying clause-to-variable ratio: instances with the ratio equals 4.25 are the most difficult ones on average (Mitchell, Selman, & Levesque 1992). Other examples include the average graph connectivity for graph coloring problem (Cheeseman, Kanefsky, & Taylor 1991); constraint tightness (Smith 1994; Prosser 1996) for constraint satisfaction problems. A generalized the notion of constrainedness for an ensemble of instances was introduced by Gent *et al.* (Gent *et al.* 1996).

However, information about the number of solutions does not tell the whole story. Mammem *et al.* (Mammen & Hogg 1997) showed that the easy-hard-easy pattern still appears for some search methods even the number of solution is held constant. In these cases, the pattern appears to be due to changes in the size of the minimal unsolvable subproblems, rather than changing number of solutions.

Another important line of research has been focused on the relationship between hardness and *backbone* variables. For satisfiable SAT instances, the backbone is the set of literals which are logically entailed by the clauses of the instance, that is, backbone variables take on the same values in all solutions.

Achlioptas *et al.* (Achlioptas *et al.* 2000) demonstrated that there is a phase transition phenomenon in the *backbone fraction*–the ratio of the size of backbone to the total number of variables–with changes in the number of unset variables or "holes" created in solutions to Quasigroup with Holes (QWH) instances. They also show that the phase transition coincides with the hardness peak in local search. However, the hardness peak of backtrack algorithms does not coincide with the phase transition of backbone fraction seen in local search. The relationships between the hardness peak and backbone fraction is even less clear for optimization and approximation problems where backbone size has been found to be negatively or positively correlated with hardness depending on the domain at hand (Slaney & Walsh 2001).

In other work, Singer *et al.* (Singer, Gent, & Smaill 2000) found that the number of solutions is only relevant for small-backbone random 3-SAT instances. They introduced a measure of the *backbone fragility* of an instance, which indicates how persistent the backbone is as clauses are removed. The backbone fragility of an instance is positively correlated with the problem hardness for local search.

Benchmark Domains and Solvers

We shall first investigate hardness and backdoor keys within the benchmark domain of a version of the Quasigroup Completion Problem (QCP) (Gomes & Selman 1997). The basic QCP problem is to complete a partially-filled Latin square, where the "order" of the instance is the length of a side of the square. We used a version called *Quasigroup with Holes* (QWH), where problem instances are generated by erasing values from a solved Latin square (Achlioptas *et al.* 2000; Kautz *et al.* 2001). Note that QWH problems are satisfiable by definition as they are built from previously solved problems.

We also will study problem hardness for a second problem domain, the morphed Graph Coloring Problem introduced by Gent *et al.* (Gent *et al.* 1999). The Graph Coloring Problem (GCP) is a well-known combinatorial problem from graph theory. Given a graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and E the set of edges connecting the vertices, we seek to find a coloring $C : V \rightarrow N$, such that connected vertices always have different colors. The challenge is to decide whether a coloring of the given graph exists for a particular number of colors. A p-morph of two graphs $A = (V, E_1)$ and $B = (V, E_2)$ is a graph $C = (V, E)$ where E contains all the edges common to A and B, a fraction p of the edges from $E_1 - E_2$ (the remaining edges of A), and a fraction $1 - p$ of the edges from $E_2 - E_1$. The test sets considered here are obtained by morphing regular ring lattices, where the vertices are ordered cyclically and each vertex is connected to its k closest in this ordering, and random graphs from the well-known class G_{nm}. The morphing ratio p controls the amount of structure in the problem instances. The test sets used in our studies are from (sat).

Finally, we consider domains of logistics planning (Kautz & Selman 1996) and circuit synthesis problems (Kamath *et al.* 1993). All instances of these domains are satisfiable and encoded in propositional encodings. The solver we used was Satz-Rand (Gomes, Selman, & Kautz 1998), a randomized version of the Satz system (Li & Anbulagan 1997) with powerful variable selection heuristics, and zChaff (version Z2001.2.17) (Zhang *et al.* 2001).

Backdoors

The *backdoor* (Williams, Gomes, & Selman 2003a; 2003b) of a problem aims to capture structural properties in a problem that underlie heavy-tailed behaviors in backtracking algorithms. A set of variables forms a backdoor for a problem instance if there is a truth assignment for these variables such that the simplified formula can be solved in polynomial time by the propagation and simplification mechanism of the SAT solver under consideration. That is, after setting the backdoor variables correctly, the simplified formula falls in a polynomially solvable class.

Given a Boolean formula $\mathcal{F}$, let V be the set of variables in $\mathcal{F}$. Let $\mathcal{A}_B : B \subseteq V \rightarrow \{True, False\}$ be a partial truth assignment and $\mathcal{F}[\mathcal{A}_B]$ denote the simplified formula obtained from the formula $\mathcal{F}$ by setting the partial truth assignment $\mathcal{A}_B$.

A set of backdoor variables is defined with respect to a particular search algorithm; once the backdoor variables are assigned certain values, the problem becomes polynomial time solvable by the algorithm. Such algorithms are called *sub-solvers* (Williams, Gomes, & Selman 2003a). A sub-solver $\mathcal{S}$ always runs in polynomial time. Given a formula $\mathcal{F}$ as the input, $\mathcal{S}$ either rejects $\mathcal{F}$, or determines $\mathcal{F}$ correctly as

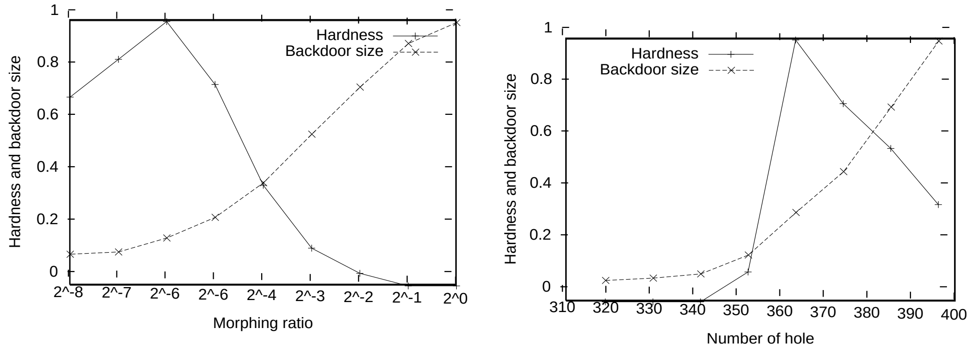

Figure 1: Normalized backdoor sizes and ensemble hardness. Left: Morphed GCP and its x-axis is the morphed ratio. Right: QWH of order 33 and its x-axis is the number of holes. For hardness, each data point represents the median value of the run-time of Satz-rand on 100 instances. For backdoor size, data points represent the mean value of 100 instances

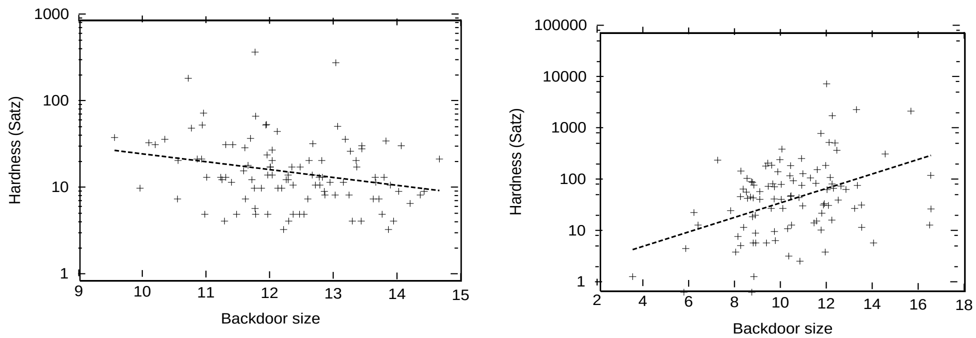

Figure 2: Backdoor sizes and instance hardness. The x-axis represents the backdoor size and the y-axis is the instance hardness. Left: Morphed GCP with morphed ratio = 2^{-6}. Right: QWH of order 33 and 363 holes. For each data point (x, y), y represents the median search cost of 100 runs of Satz-rand and x represents the mean size of 100 backdoor sets.

unsatisfiable or satisfiable, returning a solution if satisfiable.

Given a sub-solver $\mathcal{S}$, a formula $\mathcal{F}$ and its variables set V, we can definite formally the notion of backdoor.

Definition 1 (Backdoor) *A nonempty subset B of the variable set V is a backdoor for $\mathcal{F}$ w.r.t $\mathcal{S}$ if for some partial truth assignment $\mathcal{A}_B$, namely, a backdoor truth assignment, $\mathcal{S}$ returns a satisfying assignment of $\mathcal{F}[\mathcal{A}_B]$.*

Note that, by definition, backdoor sets exist only for satisfiable instances. A corresponding notion of *strong backdoors* can be defined for unsatisfiable instances (Williams, Gomes, & Selman 2003a). We will not discuss strong backdoors.

Note that any superset of a backdoor set is also a backdoor set. We are interested in backdoor sets with no unnecessarily variables, which we refer to as *minimal backdoor* sets.

Definition 2 (Minimal Backdoor) *A nonempty backdoor set B for $\mathcal{F}$ w.r.t $\mathcal{S}$ is minimal if for any $v \in B$ such that $B - v$ is not a backdoor set for $\mathcal{F}$ w.r.t $\mathcal{S}$.*

For the rest of this paper, unless otherwise specified, we assume that a backdoor set is minimal.

Backdoor Size and Hardness

We performed empirical studies to test the relation between backdoor sizes and hardness on the morphed GCP, QWH, circuit synthesis, and logistics planning domains. We modified the SAT solver Satz-rand to keep track of the set of variables selected for branching and their truth assignments in a solution. We also used Satz as a sub-solver by turning off its variable branching function, *i.e.*, allowing only unit propagation and simplification. By the definition of backdoor, the set of branching variables in a solution contains a backdoor set. Unnecessary variables were randomly removed one by one from the set of branching variables until the remaining set of variables is a minimal backdoor set. For each ensemble, we collected data of 100 instances; for each instance, we generated 100 backdoor sets.

Recall that a backdoor is a set of variables that, if set correctly, renders the remaining problem trivial to solve. Thus, the difficulty of identifying and assigning backdoor variables correctly is proportional to the difficulty of solving a problem. It is reasonable to assume that the backdoor of a problem reflects a core structural property that is closely linked to the problem hardness of the problem. However, we have found that the straightforward way of using backdoor size as a predictor does not correlate well with ensemble or instance hardness.

Figure 1 shows that there is no significant correlation between backdoor size and ensemble hardness for either morphed GCP or QWH domains. For morphed GCP, the backdoor size increases with increasing morphing ratio p, or the declining amount of structure in the problem instances. This is consistent with prior observations that structured domains tend to have smaller backdoor sizes, while random domains tend to have larger backdoor sizes (Williams, Gomes, & Selman 2003a).

We have also noted that ensemble hardness for morphed GCP does not increase monotonically with increasing backdoor size. Instead, the ensemble with the largest backdoor size is the easiest to solve and the peak of hardness is located at some intermediate point between the totally random and totally structured ensembles. The same observation applies to QWH problems: Backdoor size increases monotonically with the increasing number of holes while the peak of hardness is located at some intermediate point.

We additionally discovered that there is poor correlation between backdoor sizes and instance hardness within an ensemble. In Figure 2, the graph on the left captures results obtained with morphed GCP, showing a weak negative correlation while the right figure for QWH shows a weak positive correlation. Empirical studies using zChaff demonstrated similar results, *i.e.*, no significant correlation between backdoor sizes and ensemble or instance hardness, which are not shown here due to limitation of space.

We believe that the reason that the size of a backdoor set alone is not enough to predict hardness is that the backdoor size does not capture the *dependencies* among backdoor variables, *i.e.*, the "dependent" variables whose truth values are completely determined by other variables in the backdoor set. For an example, in the case of morphed GCP, although the backdoor sizes of the totally random ensemble are larger, there are few dependent variables. Therefore, the probability of setting some backdoor variable to the "wrong" value is smaller, even though more backdoor variables must be set. On the other hand, for instances at the hardness peak, the dependencies among backdoor variables are higher and there are more dependent variables. In these cases, it is likely that a backdoor variable will be set to a value that makes the remaining problem unsatisfiable.

This observation suggests that problem hardness is the result of the interaction between backdoor size and dependencies among backdoor variables, which we will address in the next section.

Backdoor Keys

In the last section, we showed that problem hardness is not a simple function of backdoor size, and proposed the need for considering the dependencies among backdoor variables. To capture such dependencies, we introduce the notion of the backdoor key and examine its relationship with problem hardness.

Defi nitions

Given a sub-solver $\mathcal{S}$, let B be a backdoor of a formula $\mathcal{F}$ with respect to $\mathcal{S}$, and $\mathcal{A}_B$ be a backdoor truth assignment, *i.e.*, a value setting of B such that $\mathcal{S}$ returns a satisfiable assignment of $\mathcal{F}$. We use $B - v$ as a simple denotation of $B - \{v\}$, for any $v \in V$. Before our exposition of backdoor keys, we need to define the notion of a dependent variable.

Definition 3 (Dependent Variable) *A variable $v \in V$ is a dependent variable of formula $\mathcal{F}$ with respect to a partial truth assignment $\mathcal{A}_B$ if $\mathcal{F}[\mathcal{A}_B]$ determines v,* i.e., *there is a unique value assignment x of v such that $\mathcal{F}[\mathcal{A}_B \cup \{v/x\}]$ is satisfiable.*

To capture the dependencies among backdoor variables, we introduce the definition of a *backdoor key*, which we define formally as follows:

Definition 4 (Backdoor Key) *A backdoor variable v is in the backdoor key set of B with respect to a backdoor truth assignment $\mathcal{A}_B$ if and only if v is a dependent variable in $\mathcal{F}$ with respect to the partial truth assignment $\mathcal{A}_{B-v}$*

In distinction to variables with superficial dependencies, which can be easily detected by unit-propagation and simplification, a backdoor key set represent a deep dependency whose detection requires extra effort beyond unit-propagation and simplification. We believe that such deep dependencies provide a view onto the core structural properties that lay at the foundations of problem hardness.

We know that once the backdoor variables are set correctly, the remaining problem becomes trivial. Thus, the difficulty of solving a problem is proportional to the difficulty of identifying and assigning backdoor variables. We do not yet know how to estimate the difficulty of identifying backdoor variables; this is an open challenge. It is generally believed that modern variable selection heuristics have done reasonably well at identifying backdoor variables from the success of the state-of-the-art backtrack search algorithms.

We believe that the difficulty of *assigning* backdoor variables correctly is captured by the notion of a backdoor key, because the dependent variables are the ones which are likely to be assigned incorrectly. Thus, the size of a backdoor key is a prime candidate for predicting problem hardness. Another candidate is the relative size of a backdoor key set, which we refer to as the *backdoor key fraction*:

Definition 5 (Backdoor Key Fraction) *A backdoor key fraction is the ratio of the size of the backdoor key set to the size of the corresponding backdoor set.*

We will see that in many domains the key fraction is a precise predictor of problem hardness.

Experiments

We performed a set of empirical studies to test our hypothesis. We used the method described in previous section to collect minimal backdoor sets and then calculated the key size for each backdoor set by testing whether a backdoor variable is in the key set with respect to the truth assignment of the backdoor set.

Fig. 3 shows the relation between ensemble hardness and key fractions as well as key sizes. The graph at the left displays results for the morphed GCP domain and the graph on the right shows results for the QWH domain. For both problem domains, the peaks of key fraction coincide with those of ensemble hardness, which illustrates a strong correlation between key fraction and ensemble hardness. On the other hand, there is little or no correlation between key size and problem hardness. For morphed GCP, the peak of backdoor key size shifts to the left of the hardness peak. For QWH, the size of backdoor key increases monotonically, which does not coincide with the easy-hard-easy pattern of problem hardness. Empirical studies using zChaff demonstrated similar results.

In predicting instance hardness within an ensemble, the key fraction also shows the strongest correlation. Fig. 4 shows the correlation between key fraction and instance hardness. For both domains, the overall shapes suggest linear correlations between the key fraction and the log of instance hardness. We performed linear regression on the data and summarized the results in table 1. As a comparison, we

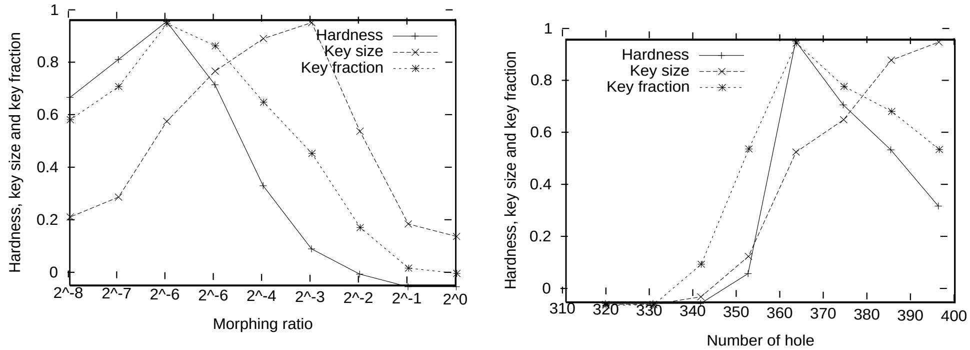

Figure 3: Normalized key sizes, key fractions and ensemble hardness. Left: Morphed GCP where the x-axis represents the morphed ratio. Right: QWH where the x-axis represents the number of unset variables or holes, removed from prior solutions. For hardness, each data point represents the median hardness of 100 instances solved by Satz-rand. For backdoor size, the data points represent the mean value of 100 instances.

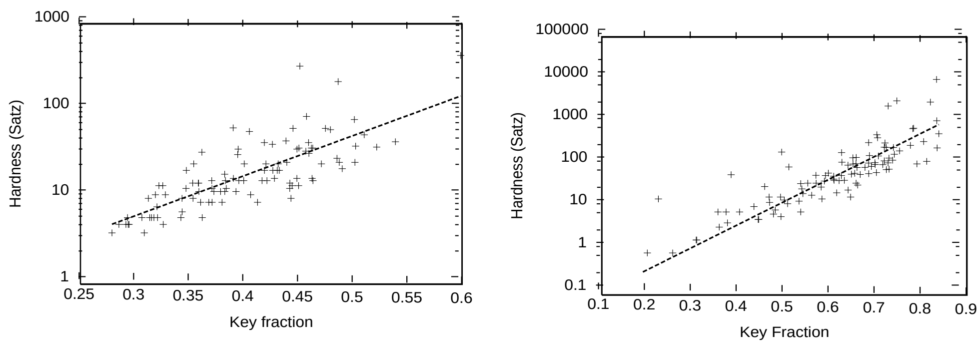

Figure 4: Key fractions and instance hardness. The x-axis is the key fraction (the ratio of key size and backdoor size) and the y-axis is the instance hardness. Left: Morphed GCP with morphed ratio = 2^{-6}. Right: QWH of order 33 and 363 holes. For each data point (x, y), y represents the median search cost of 100 runs of Satz-rand and x represents the mean key fraction of 100 backdoor sets.

also include results for backdoor size and key size. We report correlation coefficients as well as two error measures for linear regression, the root mean squared error (RMSE) and mean absolute error (MAE). The table shows that the *key fraction is the best predictor for problem hardness in all aspects*.

An intuitive explanation for the success of the key fraction in predicting hardness is that it represents the probability that a backdoor variable is dependent. We have argued that variable selection heuristics essentially attempt to identify backdoor variables, that only dependent backdoor variables can be set to incorrect values (values that rule out all solutions), and that large backtracking trees and long runs result when backdoor variables are initially set to incorrect values. Putting these arguments together makes the prediction that *as the key fraction increases the probability that the solver has a long run should increase as well*. Our experimental results show that this is exactly the case.

The concept of backdoor keys works well for domains with relatively large key fractions, such as QWH and morphed GCP. We did the same investigation for the logistics and circuit synthesis domains but found no significant correlations. It turns out that for the two latter domains, most instances from the two domains have key sets of size zero! In other words, given any backdoor and its corresponding solution, you can flip the truth assignment of any single variable in the backdoor and still extend the backdoor to a solution.

A similar issue was highlighted in work that looked at the phenomena of backbones in graph coloring problems (Culberson & Gent 2001). By the original definition of backbone—a variable that is fixed in all solutions—the problems had no backbones. Culberson and Gent therefore adopted a new definition of backbone in terms of *pairs* of variables that must take on different values in all solutions. We are currently pursuing a similar generalized notion of backdoor key to take into consideration special problems where the key fails to be predictive of hardness. For example, one can define dependent pairs of variables with respect to a partial truth assignment, and then analogously define the set of backdoor key pairs.

Conclusion and Research Directions

We extended the notion of backdoor variables by introducing and investigating the concept of *backdoor keys*. Backdoor keys capture the link between dependencies among backdoor variables and problem complexity. To examine the relationship between the backdoor key of a problem and its hardness, we performed experiments on several structured domains. Our analysis suggests that the key fraction is a good predictor for both ensemble and instance hardness for domains with relatively large key fractions.

We are pursuing a deeper understanding of backdoors and backdoor keys, as well as their relationships to problem hardness. In parallel, we are seeking to apply the backdoor and backdoor key concepts to inform problem-solving methods. Moving beyond backdoor keys, we are seeking to

Measures	Satz-rand						zChaff					
	QWH			GCP			QWH			GCP		
	F	K	B	F	K	B	F	K	B	F	K	B
Correlation Coefficient	0.86	0.79	0.42	0.78	0.56	-0.26	0.70	0.70	0.42	0.54	0.53	0.01
RMSE	0.41	0.47	0.71	0.24	0.37	0.38	0.39	0.68	0.87	0.37	0.37	0.38
MAE	0.29	0.37	0.57	0.29	0.28	0.28	0.29	0.53	0.69	0.31	0.31	0.32

Table 1: Results of linear regression analysis for instance hardness, where F represents key fraction, K represents key size and B represents backdoor size

understand more about the hidden structure of problems. We are particularly interested in understanding structure among variables in the backdoor set for domains with small key fractions. The presence and nature of such hidden structures may be related to the difficulty of identifying backdoor variables. Finally, we interested in investigating the effect of clause learning on backdoor and key sizes. Clause learning (Marques-Silva & Sakallah 1996; Bayardo & Schrag 1997; Zhang 1997; Moskewicz *et al.* 2001; Zhang *et al.* 2001) can be seen as a mechanism that helps to identify variable dependencies. Our hypothesis is that the effectiveness of clause learning may be related to the variation in backdoor or key sizes during the learning process.

References

Achlioptas, D.; Gomes, C. P.; Kautz, H. A.; and Selman, B. 2000. Generating satisfiable problem instances. In *AAAI/IAAI*, 256–261.

Bayardo, R., and Schrag, R. 1997. Using csp look-back techniques to solve real-world sat instances. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 203–208. New Providence, RI: AAAI Press.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. M. 1991. Where the Really Hard Problems Are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, IJCAI-91, Sidney, Australia*, 331–337.

Culberson, J., and Gent, I. 2001. Frozen development in graph coloring. *Theoretical Computer Science* 265(1–2):227–264.

Gent, I. P.; MacIntyre, E.; Prosser, P.; and Walsh, T. 1996. The constrainedness of search. In *AAAI/IAAI, Vol. 1*, 246–252.

Gent, I. P.; Hoos, H. H.; Prosser, P.; and Walsh, T. 1999. Morphing: Combining structure and randomness. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, 654–660.

Gomes, C. P., and Selman, B. 1997. Problem Structure in the Presence of Perturbations. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 221–227. New Providence, RI: AAAI Press.

Gomes, C. P.; Selman, B.; and Kautz, H. 1998. Boosting Combinatorial Search Through Randomization. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 431–438. New Providence, RI: AAAI Press.

Kamath, A. P.; Karmarkar, N. K.; Ramakrishnan, K. G.; and Resende, M. G. C. 1993. An interior point approach to Boolean vector function synthesis. *Proceedings of the 36th MSCAS* 185–189.

Kautz, H., and Selman, B. 1996. Pushing the envelope: planning, propositional logic, and stochastic search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1188–1194. Portland, OR: AAAI Press.

Kautz, H.; Ruan, Y.; Achlioptas, D.; Gomes, C. P.; Selman, B.; and Stickel, M. 2001. Balance and filtering in structured satisfiable problems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, 351–358.

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

Li, C. M., and Anbulagan. 1997. Heuristics based on unit propagation for satisfiability problems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 366–371. AAAI Pess.

Mammen, D. L., and Hogg, T. 1997. A new look at the easy-hard-easy pattern of combinatorial search difficulty. *Journal of Artificial Intelligence Research* 7:47–66.

Marques-Silva, J., and Sakallah, K. 1996. Grasp – a new search algorithm for satisfiability. In *Proceedings of the International Conference on Computer Aided Design*, 220–227.

Mitchell, D. G.; Selman, B.; and Levesque, H. J. 1992. Hard and easy distributions for SAT problems. In Rosenbloom, P., and Szolovits, P., eds., *Proceedings of the Tenth National Conference on Artificial Intelligence*, 459–465. Menlo Park, California: AAAI Press.

Moskewicz, M. W.; Madigan, C. F.; Zhao, Y.; Zhang, L.; and Malik, S. 2001. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*.

Prosser, P. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence* 81:81–109.

www.satlib.org.

Singer, J.; Gent, I.; and Smaill, A. 2000. Backbone Fragility and the Local Search Cost Peak. *Journal of Artificial Intelligence Research* 12:235–270.

Slaney, J. K., and Walsh, T. 2001. Backbones in optimization and approximation. In *IJCAI*, 254–259.

Smith, B. 1994. Phase transition and the mushy region in constraint satisfaction problems. In *In Proceedings of the eleventh European Conference on Artificial Intelligence*, 100–104.

Williams, R.; Gomes, C.; and Selman, B. 2003a. Backdoors to typical case complexity. In *IJCAI03*.

Williams, R.; Gomes, C.; and Selman, B. 2003b. On the connections between backdoors, restarts, and heavy-tailedness in combinatorial search. In *SAT03*.

Zhang, L.; Madigan, C. F.; Moskewicz, M. W.; and Malik, S. 2001. Efficient conflict driven learning in boolean satisfiability solver. In *ICCAD*, 279–285.

Zhang, H. 1997. SATO: an efficient propositional prover. In *Proceedings of the International Conference on Automated Deduction (CADE'97), volume 1249 of LNAI*, 272–275.