This paper appears in the *Proceedings of the Workshop "Studying and Solving Really Hard Problems" of the First International Conference on Principles and Practice of Constraint Programming (CP'95)*, Cassis, France, 1995.

TWSAT: A New Local Search Algorithm for SAT Performance and Analysis *

Mazure Bertrand
CRIL
Université d'Artois
rue de l'Université SP 16
F-62307 Lens Cedex
France
mazure@lens.lifl.fr

Saïs Lakhdar
CRIL - IUT de Lens
Université d'Artois
rue de l'Université SP 16
F-62307 Lens Cedex
France
sais@lens.lifl.fr

Grégoire Eric
CRIL
Université d'Artois
rue de l'Université SP 16
F-62307 Lens Cedex
France
gregoire@lens.lifl.fr

Abstract

In this paper, is presented a new local search algorithm for SAT and compared with Selman et al.'s GSAT algorithms. This new algorithm, called TWSAT, departs from GSAT by making a *systematic* use of a taboo list that forbids recurrent local reparations. TWSAT achieves significant performance improvements in the resolution of many problems, in particular hard random K-SAT instances. Moreover, we find peculiar experimental properties are obtained about the optimal length of the taboo list, which may lead to a better understanding of the nature of really hard random problems.

Keywords

SAT, NP-completeness, local search methods

* This work has been supported by the Ganymède II project of the Contrat de Plan Etat/Nord-Pas-de-Calais, by the PRC-GDR Intelligence Artifielle, by the MESR (Ministère de l'Enseignement Supérieur et de la Recherche and by the IUT de Lens.

1 Introduction

SAT, i.e. checking the satisfiability of a boolean formula in conjunctive normal form, is a canonical NP-complete problem. Moreover, it is a really fundamental problem in mathematical logic, automated reasoning, artificial intelligence and various computer science domains like VLSI design.

Recently, there has been a renewal of interest in understanding the nature of the difficulty of SAT (see e.g. [2, 6]). At the same time, several authors have proposed new -but amazingly simple and efficient- algorithms allowing for a real breakthrough in the class of computer-solvable SAT instances (see e.g. [12, 13, 4]).

More precisely, a class of very hard SAT instances has been characterized (see e.g. [2, 5, 11]). This class appears to be the set of random generated K-SAT instances whose probability of being satisfiable is close to 0.5 (i.e. the critical point of "phase transition"). These problems are most often beyond the reach of the most efficient techniques derived from conventional algorithms, like Davis and Putnam's one [3]. In order to address them, two families of algorithms have been designed recently. The first one is made of logically complete techniques that aim at proving the inconsistency of SAT instances (see e.g. [5]). The second one is formed of uncomplete techniques based on local reparations that attempt to find a model for SAT instances. Most notably, Selman et al.'s have proposed a very simple local search algorithm that appears to be surprisingly good in solving hard large satisfiable problems [12, 13, 8].

In this paper, is presented a new local search algorithm for SAT and compared with Selman et al.'s GSAT algorithms. This new algorithm, called TWSAT (Taboo Walk Strategy for SAT), departs from GSAT by making a *systematic* use of a taboo list that forbids recurrent local reparations. It is shown that TWSAT achieves significant performance improvements in the resolution of many problems, in particular random K-SAT instances chosen at the critical point of the phase transition. Moreover, peculiar properties are obtained about the optimal length of the (experimentally obtained) taboo list, which may lead to a better understanding of the nature of really hard random problems.

In the next section, we recall the canonical K-SAT fixed-clause-length random generation model and focus on the critical point of phase transition. Then,

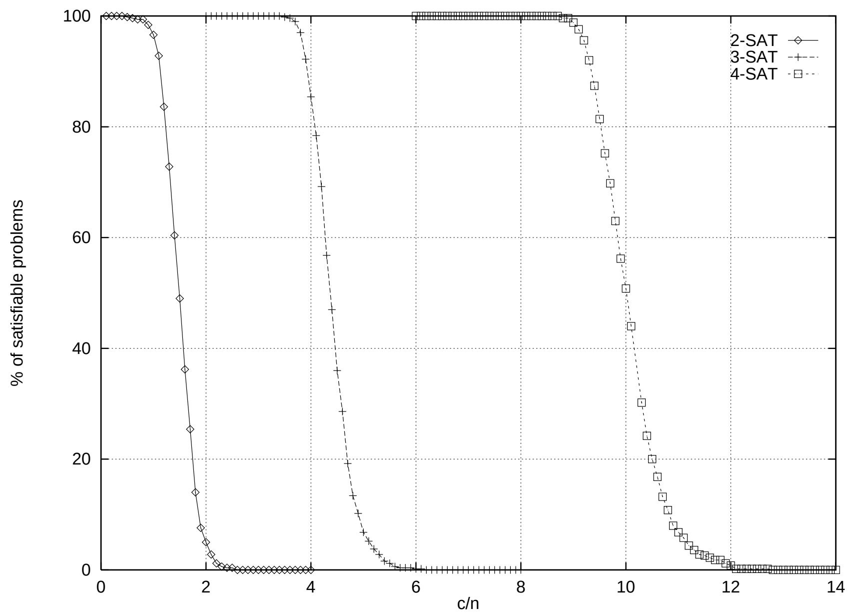

Figure 1: Phase transition phenomena

we present Selman et al.'s GSAT algorithm. Based on the systematic use of a taboo list, TWSAT is then motivated and presented. We focus on the optimal lengths of the taboo list with respect to several criteria and obtain some what surprising findings. A comparison between the performance of GSAT (and, more precisely, the Random-Walk Strategy version [13]) and TWSAT is then conducted. Before we conclude, we give some promising ideas for further research.

2 Hard random SAT instances

SAT consists in checking the satisfiability of a boolean formula in conjunctive normal form (CNF). Let us recall here that any propositional formula can be translated thanks to a linear time algorithm in CNF, equivalent with respect to SAT. A CNF formula is a set (interpreted as a conjunction) of clauses, where a clause is a disjunction of literals. A literal is a positive or negated propositional variable.

An interpretation of a boolean formula is an assignment of truth values to

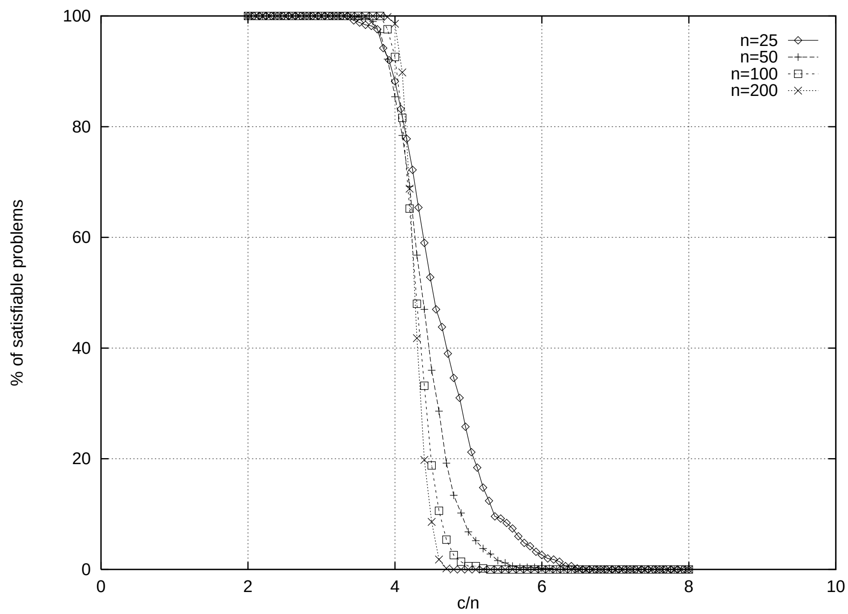

Figure 2: 3-SAT

its variables. A model is an interpretation that satisfies the formula.

SAT is one of the NP-complete problems of reference although theoretical and experimental results show good average-case performance for several classes of SAT instances (see e.g. [7]).

However, it has been observed that hard random instances of SAT are located at a "phase transition". Let us illustrate this "phase transition" in the usual K-SAT fixed-clause-length model (see e.g. [2, 1, 5, 11]), a random generation model where the number of literals per clause is a given value K and the sign of each literal is also randomly generated (with a probability 0.5).

In Figure 1 and 2, we see the phase transition observed by many authors. The probability of satisfiability decreases abruptly from 1 to converge towards 0 in a transition phase as the c/n ratio increases (where c represents the number of clauses and n the number of variables). The location of the transition phase depends on the length of clauses and on the number of variables. In Figure 1, we see how the curves moves to the right as this length increases. In Figure 2, we see how the curves straightens when the number of variables increases. It has been shown experimentally that instances at these transition phases where

Procedure *GSAT*
Input : *a set of clauses S, MAX-FLIPS, and MAX-TRIES*
Output : *a satisfying truth assignment of S, if found*
Begin
  for *i* := *1* **to** *MAX-TRIES*
    I := *a randomly generated truth assignment*
    for *j* := *1* **to** *MAX-FLIPS*
      if *I satisfies S* **then return** *I*
      x := *a propositional variable such that a change in its truth assignment gives the largest increase in the number of clauses*[1] *of S that are satisfied by I*
      I := *I with the truth assignment of x reversed*
    end for
  end for
  return *"no satisfying assignment found"*
End

Figure 3: GSAT Algorithm: basic version

the probability of being satisfiable is 0.5 are really hard problems (actually, it has been proved by [2] that these problems are exponential for resolution).

3 Selman et al.'s local search algorithm

Let us now briefly recall Selman et al.'s GSAT algorithm [12, 13]. This algorithm performs a greedy local search for a satisfying assignment of a set of propositional clauses. The algorithm starts with a randomly generated truth assignment. It then changes ("flips") the assignment of the variable that leads to the largest increase in the total number of satisfied clauses. Such flips are repeated until either a model is found or a preset maximum number of flips (MAX-FLIPS) is reached. This process is repeated as needed up to a maximum of MAX-TRIES times (see Figure 3).

[1] this number can be negative

In the sequel, we shall consider a more recent version of GSAT [13], i.e. the Random Walk Strategy, which outperforms basic GSAT procedures. This variant of GSAT selects the variable to be flipped in the following way : it either picks with probability p a variable occurring in some unsatisfied clause or follows, with probability 1-p, the standard GSAT scheme, i.e. makes the best possible local move.

Clearly, this very simple algorithm is logically uncomplete and belongs to the local search procedures family. However, it is surprisingly efficient in demonstrating that CNF formulas are satisfiable, in particular K-SAT instances at the transition phase.

4 TWSAT: an improved local search algorithm

Let us note that Random Walk Strategy introduces an additional level of randomness in basic GSAT and thus makes an analytical study of GSAT more difficult to conduct. Another randomness property of GSAT lies in the selection of the variable to be flipped. Indeed, as Selman et al.'s stress it: "Another feature of GSAT is that the variable whose assignment is to be changed is chosen *at random* from those that would give an equally good improvement. Such non-determinism makes it very unlikely that the algorithm makes the same sequence of changes over and over [12]".

Moving further towards the goal of avoiding recurrent flips, we propose a new version of GSAT, called TWSAT. TWSAT departs from basic GSAT by making a systematic use of a taboo list of variables in order to avoid recurrent flips and thus escape from local minima. This should also allow a better and more uniform coverage of the search space. More precisely, TWSAT keeps a fixed length -chronologically-ordered FIFO- list of flipped variables and prevents any of the variables in the list from being flipped again during a given amount of time.

Let us stress that this use of a taboo list is systematic during the search process, i.e. the taboo list is updated each time a flip is made. This differs from the usual ways of dealing with taboo lists that restrict their use to situations where local minima are reached (see e.g. [10]).

The efficiency of most local search procedures depends heavily on a good

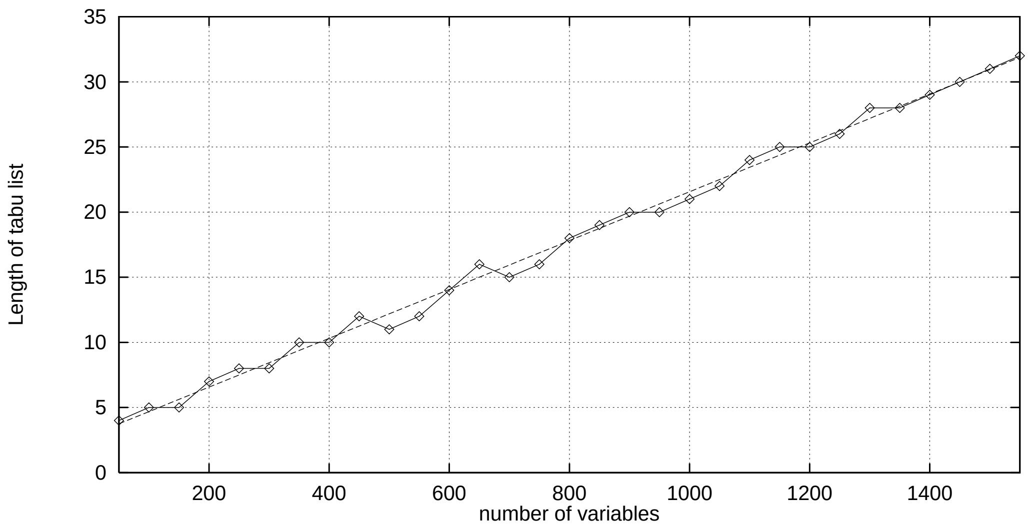

Figure 4: Optimal length of taboo list for 3-SAT problems (c/n=4.3)

setting of their parameters. For instance, Selman et al.'s [13] suggest specific values for GSAT parameters like MAX-TRIES, MAX-FLIPS and, very importantly, the probability p presented above.

In the next section, we show how to fine-tune the essential parameter of TWSAT, i.e. the length of the taboo list, with respect to properties of the problem like the number of clauses, the number of variables and the length of the clauses.

5 Experimental fine-tuning of TWSAT: peculiar results

In order to find optimal lengths of the taboo lists, we have conducted the following extensive experimentations. First, we focused on 3-SAT. According to the standard fixed-length-clause model, we randomly generated 500 instances at the phase transition (i.e. when c/n = 4.3) for every number of variables ranging from 50 to 1000 (by steps of 50). The number of instances has been limited to 100 for every number of variables between 1000 and 1500 (also by steps of 50). For each such instance, we have run TWSAT with a length of the taboo list varying from 1 to 50. In Figure 4, we show the (experimentally obtained) optimal length of the taboo list with respect to the number of variables. In the considered range of number of variables, this curve appears to be linear in the number of variables.

Experimental result : $\boxed{l = 0.01875 \times n + 2.8125}$

where l is the optimal length of taboo list
and n is the number of variables.

Moreover, we have also noted :

- that a slight departure from the optimal length leads to a corresponding graceful degradation of the performance of TWSAT.
- that these lengths remain optimal for random-generated instances outside the transition phase.

We have reconducted the above tests for 4-SAT (100 instances for each number of variables ranging from 50 to 600 (by steps of 50)) and have obtained the same values for the optimal lengths.

6 TWSAT vs. GSAT

We have made extensive experimental comparisons between (Random Walk) GSAT and TWSAT for 3-SAT instances at the phase transition. Both algorithms have been implemented in a common platform written in C under Linux 1.1.59 for PC, available from the authors[2].

Both algorithms are best compared with respect to the percentage of solved problems and with respect to the number of performed flips (let us stress that our taboo list is implemented as a circular list whose FIFO access is made in constant time). The percentage of solved problems is actually relative to the 50 % of tested instances that are expected to be satisfiable since they are selected at the transition phase. We have also indicated the (average cumulated flips for solved instances)/(percentage of solved problems) ratio in order to conduct

[2]platform available by ftp anonymous: ftp.lifl.fr, directory pub/projects/SAT

problems		Nb.	GSAT			
n	c	inst.	time (sc.)	flips	solved	ratio
100	430	500	.18	2803	88%	31.85
200	860	500	1.99	18626	73%	255.85
400	1700	500	15.03	204670	100%	2046.70
600	2550	500	19.59	250464	62%	4013.85
800	3400	500	140.61	1809986	67%	26854.39
1000	4250	500	369.88	4633763	57%	81009.84
2000	8240	50	3147.26	26542387	16%	1658899.19

problems		Nb.	TWSAT			
n	c	inst.	time (sc.)	flips	solved	ratio
100	430	500	.11	1633	93%	17.60
200	860	500	.73	9678	74%	130.78
400	1700	500	11.51	145710	100%	1457.10
600	2550	500	13.92	167236	65%	2580.80
800	3400	500	99.45	1143444	71%	16150.34
1000	4250	500	292.10	3232463	62%	51802.29
2000	8240	50	3269.15	29415465	40%	735386.63

Table 1: GSAT vs. TWSAT

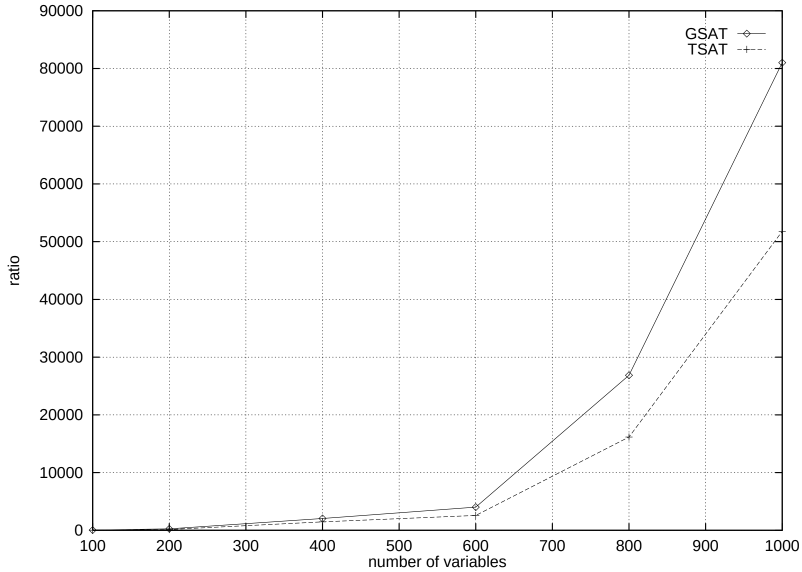

Figure 5: Results for 3-SAT instances obtained by GSAT and TWSAT

a fair comparison when one of the algorithms solves more instances than the other one. Parameters for GSAT and TWSAT are those proposed by Selman et al.'s in [13].

The following table (Table 1) summarizes the results and shows that TWSAT is more efficient than GSAT. Let us stress that the given number of flips corresponds to the cumulated number of performed flips during the different tries. However, most of the time, TWSAT just needed one try.

Let us note that the positive difference in favour of TWSAT increases with the number of variables. This is illustrated in the following diagram (see Figure 5).

Results obtained by TWSAT for 2000 variables are the most significant ones (see the above Table 1) but could not have been inserted in the Figure 5 because of the huge difference with GSAT performance.

7 Further work

Clearly, the linearity of the curve of the (experimental) optimal lengths of the taboo list was quite unexpected, as well as the fact that this curve does only depend on the number of variables. We think that this finding should be the object of further research. We feel that the length of the taboo list is related to some extent to the height of local extrema. We are currently working on that, trying to relate this feature to the nature of really hard random SAT problems.

We are also testing TWSAT with respect to other random generation models and real-life examples (as those suggested in the DIMACS challenge [4]). For instance, we have obtained promising preliminary results for the random generator by [9], which gives rise to instances of clauses with mixed lengths.

8 Conclusion

In this paper, TWSAT, a new local search algorithm for SAT, has been proposed and compared with Selman et al.'s GSAT algorithms. TWSAT makes a *systematic* use of a taboo list and suppresses one of the randomness properties of Random Walk Strategy GSAT. TWSAT achieves significant and surprising performance improvements in the resolution of many problems, in particular hard random K-SAT instances. This improvement increases with the size of the problem. Moreover, have been obtained peculiar properties about the (experimental) optimal length of the taboo list, which may lead to a better understanding of the nature of really hard random problems.

References

[1] P. Cheeseman, B. Kanefsky, W.M. Taylor (1991). Where the Really Hard Problems are. *Proc. IJCAI-91*, pp. 163-169.

[2] V. Chvátal, E. Szemerédi (1988). Many Hard Examples for Resolution. *Journ. of the ACM*, vol. 33, no. 4, pp. 759-768.

[3] M. Davis, H. Putnam (1960). A Computing Procedure for Quantification Theory. *Journ. of the ACM*, vol. 7, pp. 201-215.

[4] DIMACS (1993). Second challenge organized by the Center for Discrete Mathematics and Computer Science of Rutgers University.

[5] O. Dubois, P. André, Y. Boufkhad, J. Carlier (1993). SAT versus UNSAT. *Journ. of the Am. Mathematical Society* (submitted). also in *Proc. of the Second DIMACS Challenge.*

[6] O. Dubois, J. Carlier (1991). Probabilistic Approach to the Satisfiability Problem. *Theoretical Computer Science*, vol. 81, pp. 65-75.

[7] J. Franco, M. Paull (1983). Probabilistic Analysis of the Davis and Putnam Procedure for Solving the Satisfiability Problem. *Discrete Applied Math.*, vol. 5, pp. 77-87.

[8] I.P. Gent, T. Walsh (1993). Towards an Understanding of Hill-climbing Procedures for SAT. *Proc. AAAI-93*, pp. 28-33.

[9] I.P. Gent, T. Walsh (1994). The SAT Phase Transition. *Proc. ECAI-94*, pp. 105-109.

[10] F. Glover (1989). Tabu search - Part I. *ORSA Journ. of Computing*, vol. 1, pp. 190-206.

[11] D. Mitchell, B. Selman, H. Levesque (1992). Hard and Easy Distributions of SAT Problems. *Proc. AAAI-92*, pp. 459-465.

[12] B. Selman, H. Levesque, D. Mitchell (1992). A New Method for Solving Hard Satisfiability Problems. *Proc. AAAI-92*, pp. 440-446.

[13] B. Selman, H.A. Kautz, B. Cohen (1993). Local Search Strategies for Satisfiability Testing. *Proc. 1993 DIMACS Workshop on Maximum Clique, Graph Coloring, and Satisfiability.*