Tabu Search for SAT *

Bertrand Mazure **Lakhdar Saïs** **Éric Grégoire**

CRIL – Université d'Artois
rue de l'Université SP 16
F-62307 Lens Cedex France
{mazure,sais,gregoire}@cril.univ-artois.fr

Abstract

In this paper, tabu search for SAT is investigated from an experimental point of view. To this end, TSAT, a basic tabu search algorithm for SAT, is introduced and compared with Selman et al. Random Walk Strategy GSAT procedure, in short RWS-GSAT. TSAT does not involve the additional stochastic process of RWS-GSAT. This should facilitate the understanding of why simple local search methods for SAT work. It is shown that the length of the tabu list plays a critical role in the performance of the algorithm. Moreover, surprising properties about the (experimental) optimal length of the tabu list are exhibited, raising interesting issues about the nature of hard random SAT problems.

Introduction

SAT, i.e. checking the satisfiability of a boolean formula in conjunctive normal form, is a canonical NP-complete problem (Cook 1971). Moreover, it is a 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. (Chvátal & Szemerédi 1988; Dubois & Carlier 1991; Mitchell, Selman, & Levesque 1992)). At the same time, several authors have proposed new –but amazingly simple and efficient– algorithms allowing for a breakthrough in the class of computer-solvable SAT instances (see e.g. (Selman, Levesque, & Mitchell 1992; Gu 1992; Selman, Kautz, & Cohen 1993; DIMACS 1993)).

More precisely, a class of very hard SAT instances has been characterized. This class is made of random generated K-SAT instances whose probability of being satisfiable is close to 0.5 and are located at the critical point of a phase transition. These problems are most often beyond the reach of the most efficient techniques derived from conventional algorithms, like Davis and Putnam's procedure (Davis & Putnam 1960). 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. (Dubois *et al.* 1996; Crawford & Auton 1993)). The second one is formed of incomplete techniques based on local reparations that attempt to find a model for SAT instances. Several authors have proposed very simple local search algorithms that prove surprisingly good in solving hard large satisfiable problems (Selman, Levesque, & Mitchell 1992; Selman, Kautz, & Cohen 1993; Gent & Walsh 1993).

In this paper, tabu search for SAT is investigated from an experimental point of view. To this end, TSAT, a basic tabu search algorithm for SAT, is introduced and compared with Selman et al. Random Walk Strategy GSAT, in short RWS-GSAT (Selman, Kautz, & Cohen 1993). TSAT proves extremely competitive in the resolution of many problems, in particular hard random K-SAT instances at the critical point of the phase transition. Actually, the FTP available GSAT already contained a basic tabu option but has not been described in the literature to our best knowledge. Moreover, no investigation of the fine-tuning of its essential parameters had ever been conducted.

In the next section, the canonical K-SAT fixed-clause-length random generation model is reviewed, with emphasis on the phase transition. Then, Selman et al. RWS-GSAT algorithm is presented. TSAT is then motivated and presented. Extensive experimentations are conducted about the optimal lengths of the tabu list with respect to several criteria, leading to surprising findings. A comparison between the performance of RWS-GSAT and TSAT is then conducted. These results were first presented for a very restricted audience in 1995 (Mazure, Saïs, & Grégoire 1995). As a conclusion, some promising ideas for further research are given.

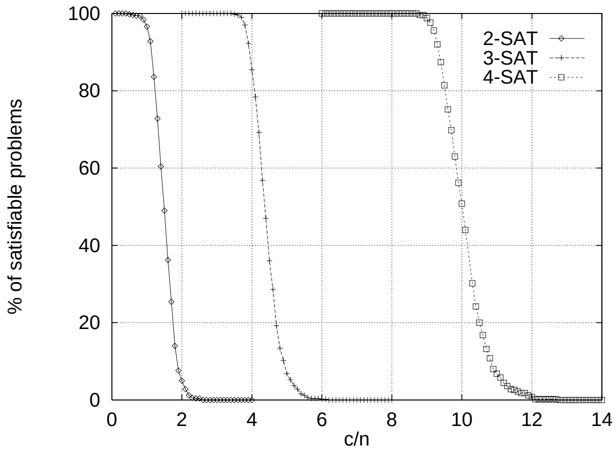

Figure 1: phase transition phenomena

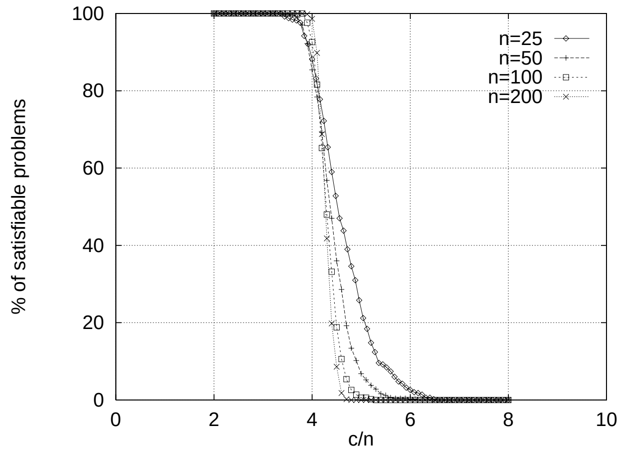

Figure 2: 3-SAT

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 into a CNF that is equivalent for SAT, thanks to a linear time algorithm (see e.g. (Siegel 1987)). 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 a negated propositional variable. An interpretation of a boolean formula is an assignment of truth values to its variables. A model of a formula is an interpretation that satisfies the formula.

Although SAT is NP-complete, theoretical and experimental results show good average-case performance for several classes of SAT instances (see e.g. (Franco & Paull 1983)).

However, hard random instances of SAT have been observed at a phase transition. Let us illustrate this phenomenon in the usual K-SAT fixed-clause-length model (see e.g. (Chvátal & Szemerédi 1988; Cheeseman, Kanefsky, & Taylor 1991; Crawford & Auton 1993; Dubois & Carlier 1991; Mitchell, Selman, & Levesque 1992)), 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 0.5 probability).

In Figures 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 phase transition as the c/n ratio increases (where c represents the number of clauses and n the number of variables). The location of the phase transition depends on the length of clauses and on the number of variables. In Figure 1, we see how the curves move to the right as this length increases. In Figure 2, we see how the curves straighten when the number of variables increases. It has been shown experimentally that instances at these phase transitions where the probability of being satisfiable is 0.5 are really hard problems. Actually, it has been proved that these problems are exponential for resolution (Chvátal & Szemerédi 1988).

Random Walk Strategy GSAT

Let us now briefly review Selman et al. GSAT algorithm (Selman, Levesque, & Mitchell 1992). 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.

In the sequel, we shall consider a more recent and efficient version of GSAT, i.e. Random Walk Strategy GSAT (in short RWS-GSAT)(Selman, Kautz, & Cohen 1993). 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 incomplete 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 phase transition.

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

Figure 3: GSAT algorithm: basic version

Restricting randomness

Let us stress that the 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. 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 nondeterminism makes it very unlikely that the algorithm makes the same sequence of changes over and over (Selman, Levesque, & Mitchell 1992)”.

Moving further towards the goal of avoiding recurrent flips, we investigate the use of tabu search (Glover 1989; 1990) for SAT by experimenting with an algorithm of our own, called TSAT. TSAT makes a systematic use (no aspiration criteria) of a tabu list of variables in order to avoid recurrent flips and thus escape from local minima. This technique was also expected to allow for a better and more uniform coverage of the search space. Let us stress that this use of a tabu list is systematic during the search process in the sense that the tabu list is updated each time a flip is made. TSAT 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. Accordingly, the tabu list contains *variables* and, for efficiency reasons, does not keep track of forbidden interpretations, explicitly.

The efficiency of most local search procedures depends heavily on a good setting of their parameters. For instance, Selman et al. suggest specific values

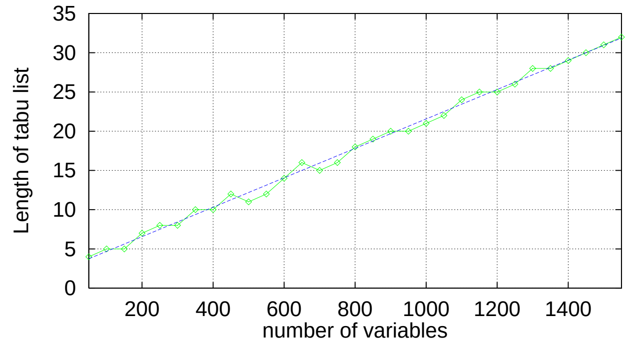

Figure 4: Optimal length of tabu lists for 3-SAT at the critical point of phase transition ($c/n = 4.25$)

for GSAT parameters like MAX-TRIES, MAX-FLIPS and, very importantly, the probability p presented above (Selman, Kautz, & Cohen 1993).

In the next section, the main parameter of TSAT is fine-tuned for random K-SAT problems, namely the length of the tabu list, using the main parameters of the problem: i.e. the number of variables and the length and number of clauses.

Experimental fine-tuning of TSAT: peculiar findings

In order to find optimal lengths of the tabu list, the following extensive experimentations have been conducted. First, the 3-SAT framework has been considered. According to the standard fixed-length-clause model, 500 instances were randomly generated at the phase transition for every number of variables ranging from 50 to 1000 (by steps of 50, with their best estimated c/n ratios). 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, TSAT has been run, varying the length of the tabu list from 1 to 50. In Figure 4, the (experimentally obtained) optimal length of the tabu list with respect to the number of variables is given. In the considered range of number of variables, this curve appears to be linear in the number of variables. Experimental result:

$$\boxed{\text{optimal length of tabu list} = 0.01875 \;\; n + 2.8125}$$

where n is the number of variables.

Moreover,

- a slight departure from the optimal length leads to a corresponding graceful degradation of the performance of TSAT. A more important distance from this optimal length leads to a dramatic performance degradation.
- these lengths remain optimal for random-generated instances outside the phase transition (the optimal

problems n	c	Nb. inst.	RWS-GSAT time (sc.)	flips	solved	ratio	TSAT time (sc.)	flips	solved	ratio
100	430	500	.18	2803	88%	31.85	.11	1633	93%	17.60
200	860	500	1.99	18626	73%	255.85	.73	9678	74%	130.78
400	1700	500	15.03	204670	100%	2046.70	11.51	145710	100%	1457.10
600	2550	500	19.59	250464	62%	4013.85	13.92	167236	65%	2580.80
800	3400	500	140.61	1809986	67%	26854.39	99.45	1143444	71%	16150.34
1000	4250	500	369.88	4633763	57%	81009.84	292.10	3232463	62%	51802.29
2000	8240	50	3147.26	26542387	16%	1658899.19	3269.15	29415465	40%	735386.63

Table 1: TSAT vs. RWS-GSAT

size depends only on the number n of involved variables according to the above equation).

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

TSAT vs. RWS-GSAT

Extensive experimental comparisons between RWS-GSAT and TSAT have been conducted for 3-SAT instances at the phase transition. Both algorithms have been implemented in a common platform written in C under Linux for Pentium PC, available from the authors. Our local implementation of Random Walk GSAT proves as efficient as Selman's one.

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 the tabu list is implemented as a circular list whose FIFO access is made in constant time. Time and number of flips in the Table 1 are average ones for solved instances. The percentage of solved problems is actually relative to the 50% (which is an approximation) of tested instances that are expected to be satisfiable since they are selected at the phase transition. The (average cumulated flips for solved instances)/(percentage of solved problems) ratio is also given in order to conduct a fair comparison when one of the algorithms solves more instances than the other one. MAX-TRIES is 5, MAX-FLIPS is n^2 and $p = 0.5$ for RWS-GSAT. These two last values are the (experimentally) best ones for GSAT with respect to random 3-SAT problems (Parkes & Walser 1996; Selman, Kautz, & Cohen 1993).

Table 1 summarizes the results and shows the good performance of TSAT. Let us stress that the given number of flips corresponds to the cumulated number of performed flips during the successive tries. However, most of the time, TSAT just required one try.

Let us note that the competitiveness of TSAT increases with the number of variables. This is illustrated in Figure 5). Results for 2000 variables are the most significant ones but could not have been inserted in the diagram because of the huge difference (see Table 1).

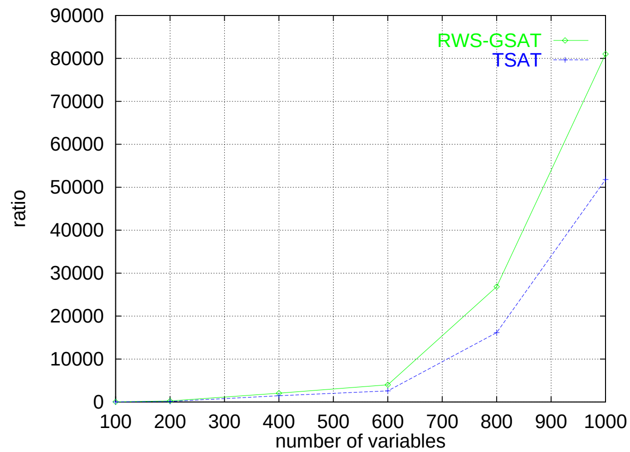

Figure 5: Results for 3-SAT instances at the phase transition obtained by RWS-GSAT and TSAT

Conclusion

TSAT, a basic tabu search algorithm for SAT, has been proposed and compared with Selman et al. Random Walk Strategy GSAT procedure. TSAT makes a systematic use of a tabu list and takes out one of the randomness properties of RWS-GSAT. TSAT proves very competitive in the resolution of many problems, in particular hard random K-SAT instances. Quite surprisingly, the optimal length of the tabu lists for these random problems proves (experimentally) linear with respect to the number of variables. This linearity was quite unexpected, as well as the fact that the length does only depend on the number of variables. This finding is certainly worth further research. Intuitively, the length of the tabu list could be 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. Also, as TSAT uses a basic form of tabu search, we are currently working on more sophisticated tabu strategies (Glover 1989; 1990) and extending them to other related problems (in this respect, see also the related work by (Battiti & Protasi 1996)). In particular, we are experimenting with TSAT with respect to structured examples (as those suggested in (DIMACS 1993)), using tabu lists whose lengths are dynamically fine-tuned. Another promising path of research consists in extending TSAT into a logically complete technique that keeps the power of local search for satisfiable instances. In this respect, (Mazure, Saïs, & Grégoire 1996) is a first promising step.

Acknowledgments

This work has been supported in part by the Ganymède II project of the Contrat de Plan Etat/Nord–Pas-de-Calais, and by the IUT de Lens. We thank the reviewers for their useful comments.

References

Battiti, R., and Protasi, M. 1996. Reactive Search, a history-based heuristic for MAX-SAT. In *Proceedings of the Workshop on Satifiability Problem.* Siena, Italy.

Cheeseman, P.; Kanefsky, B.; and Taylor, W. 1991. Where the Really Hard Problems are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI'91)*, 163–169.

Chvátal, V., and Szemerédi, E. 1988. Many Hard Examples for Resolution. *Journal of the Association for Computing Machinery* 33(4):759–768.

Cook, S. 1971. The Complexity of Theorem Proving Procedures. In *Proceedings of the third Ann. Symp. on Theory of Computing, ACM*, 151–158.

Crawford, J., and Auton, L. 1993. Experimental results on the cross-over point in satisfiability problems. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, 21–27.

Davis, M., and Putnam, H. 1960. A Computing Procedure for Quantification Theory. *Journal of the Association for Computing Machinery* 7:201–215.

DIMACS. 1993. *Proceedings of the Second Challenge, organized by the Center for Discrete Mathematics and Computer Science of Rutgers University.*

Dubois, O., and Carlier, J. 1991. Probabilistic Approach to the Satisfiability Problem. *Journal of Theoretical Computer Science* 81:65–75.

Dubois, O.; André, P.; Boufkhad, Y.; and Carlier, J. 1996. SAT versus UNSAT. In Johnson, D., and Trick, M., eds., *Cliques, Coloring and Satisfiability, Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Science. American Mathematical Society.

Franco, J., and Paull, M. 1983. Probabilistic analysis of the Davis and Putnam Procedure for Solving the Satisfiability Problem. *Journal of Discrete Applied Math.* 5:77–87.

Gent, I., and Walsh, T. 1993. Towards an Understanding of Hill-climbing Procedures for SAT. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, 28–33.

Glover, F. 1989. Tabu search - Part I. *ORSA Journal of Computing* 1(3):190–206.

Glover, F. 1990. Tabu search - Part II. *ORSA Journal of Computing* 2(1):4–32.

Gu, J. 1992. Efficient Local Search for Very Large-Scale Satisfiability Problems. *SIGART Bulletin* 3(1):8–12.

Mazure, B.; Saïs, L.; and Grégoire, E. 1995. A New Local Search Algorithm for SAT: Performance and Analysis. In *Proceedings of the CP'95 Workshop on Studying and Solving Really Hard Problems*, 127–130.

Mazure, B.; Saïs, L.; and Grégoire, E. 1996. Detecting Logical Inconsistencies. In *Proceedings of the Fourth International Symposium on Artificial Intelligence and Mathematics (AI/MATH-96)*, 116–121.

Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and Easy Distributions of SAT Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, 459–465.

Parkes, A., and Walser, J. 1996. A Resampling Method to Optimize Maxflips in Local Search: Theory and Application. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, 356–361.

Selman, B.; Kautz, H.; and Cohen, B. 1993. Local Search Strategies for Satisfiability Testing. In *Proceedings of DIMACS Workshop on Maximum Clique, Graph Coloring, and Satisfiability*, 1993.

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A New Method for Solving Hard Satisfiability Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, 440–446.

Siegel, P. 1987. Représentation et utilisation des connaissances en calcul propositionnel. Thèse d'État, Groupe d'Intelligence Artificielle, Université d'Aix-Marseille II.