An efficient local search method for random 3-satisfiability

Sakari Seitz & Pekka Orponen
Laboratory for Theoretical Computer Science
Helsinki University of Technology*

Abstract

We report on some exceptionally good results in the solution of randomly generated 3-satisfiability instances using the "record-to-record travel (RRT)" local search method. When this simple, but less-studied algorithm is applied to random instances from the problem's satisfiable phase, it seems to find satisfying truth assignments almost always in linear time, with the coefficient of linearity depending on the ratio α of clauses to variables in the generated instances. RRT has a parameter for tuning "greediness". By lessening greediness, the linear time phase can be extended up to very close to the satisfiability threshold α_c. Such linear time complexity is typical for random-walk based local search methods for small values of α. Previously, however, it has been suspected that these methods necessarily lose their time linearity far below the satisfiability threshold. The only previously introduced algorithm reported to have nearly linear time complexity also close to the satisfiability threshold is the survey propagation (SP) algorithm. However, SP is not a local search method and is more complicated to implement than RRT.

1 Introduction

The K-satisfiability problem (K-SAT) is a fundamental problem in the theory of computation [10]. An instance of the problem is a Boolean formula consisting of M clauses, each of which is a set of K literals, i.e. Boolean variables or their negations. The goal is to determine whether the formula has a satisfying truth assignment, i.e. an assignment of truth values 'true' and 'false' to its variables so that each clause contains at least one literal that evaluates to 'true'. For $K \geqslant 3$, the K-SAT problem is a representative member of the class of NP-complete problems [10]. It is strongly conjectured, although not yet proved, that the solution of any NP-complete problem requires a computation time that grows superpolynomially with respect to input size, at least in the worst case. On the other hand, solving randomly generated instances of NP-complete problems is often possible in polynomial time with high probability, in particular if the instances have some special structure (see e.g. [32] and references therein).

The K-SAT problem is also closely related to the spin glass models [17] studied in the statistical physics of disordered systems, and its behavior on random instances has recently been extensively analyzed from this point of view. Random instances of K-SAT are easily generated: each of the M clauses is a set (of size K) selected at random from the set of N variables and their negations. The ratio of clauses to variables $\alpha = M/N$ is an important parameter, because it reflects the constrainedness of the problem. It has been (nearly) proved [9] that for each value of $K \geqslant 2$ there exists a critical value α_c that separates the satisfiable (Sat) "phase" of the problem from the

*Research partially supported by grant 81120 from the Academy of Finland. Address: P. O. Box 5400, FIN-02015 HUT, Finland. E-mail: `Firstname.Lastname@hut.fi`.

unsatisfiable (Unsat) "phase" [20, 15, 21]. More precisely, this means that if the ratio α is held constant as $N \to \infty$, then almost all instances are satisfiable if $\alpha < \alpha_c$, and almost all instances are unsatisfiable if $\alpha > \alpha_c$. The exact value of the satisfiability threshold α_c has not yet been rigorously determined for any $K \geqslant 3$, but many estimates for $K = 3$ have been derived. Currently the best guesses are around $\alpha_c \approx 4.26$ [20, 6, 4], whereas the known rigorous bounds are at $\alpha_c > 3.42$ [12] and $\alpha_c < 4.506$ [7]. Interestingly, it also seems that instances requiring longest solution times using almost any algorithm are clustered near the Sat/Unsat transition [20, 15, 21].

2 The RRT Algorithm

Because of some background considerations from combinatorial landscape theory [25], we decided to investigate the behavior of the *record-to-record travel* (RRT) algorithm [8, 1] on the random 3-SAT problem. There are only a few papers addressing the properties and effectiveness of RRT, although it was proposed a decade ago and some excellent results have been obtained with it [8, 23]. We are not aware of any previous studies applying RRT to the K-SAT problem.

Given an objective function to be optimized (say, maximized) over a large configuration space, RRT starts from a random configuration. Then randomly chosen small changes are repeatedly made to the current configuration, accepting however only such changes that lead to a configuration whose objective value is not much worse than the best value found so far (the "record"). In its simplest form RRT has only one tunable parameter, the maximum allowed *deviation* d that determines how much worse values than the current record are accepted.

When RRT is applied to search for satisfying truth assignments of a K-SAT formula, a configuration corresponds to a truth assignment for all the variables in the formula, and the objective function to be maximized is the number of clauses satisfied by the given configuration. A solution is a truth assignment that makes *all* clauses true. A small change is a flip of the truth value of one variable from true to false or vice versa. A flip is accepted, if it leads to a configuration that has at most d fewer satisfied clauses than the current best configuration. It is essential to use a version of RRT where the variable to be flipped is always chosen among those occurring in the clauses that are currently unsatisfied (specifically, one of the unsatisfied clauses is chosen randomly, and from that clause one of the K variables is chosen at random). We call this method *focused RRT* (FRRT). Similar focusing is used in *random walk* algorithms [24, 27] and in the *WalkSAT* algorithms [28, 16].

As a time measure we use the number of flips (rejected flips included) performed in an RRT run. In the case of random K-SAT instances the time needed to make a single flip does not in principle depend much on the number of variables in a formula, although in practice the use of cached memory and other systems software issues cause a number of unpredictable scale effects. Also it should be noted that accepted and rejected flips can require different average CPU times, the difference depending on the implementation of the algorithm, and the time requirement per flip of course increases with increasing α. In the *WalkSAT* algorithms, to make a flip often requires checking the influence of *every* variable in one clause, but this is still counted traditionally as one flip [28, 16].

Stochastic local search methods [1] such as RRT cannot discover that a K-SAT instance is in fact unsatisfiable. Complete (systematic) methods, i.e. algorithms that can determine for any instance whether it is satisfiable or not, are usually less efficient than stochastic local search methods at finding a solution for satisfiable instances, when instances are randomly generated [11].

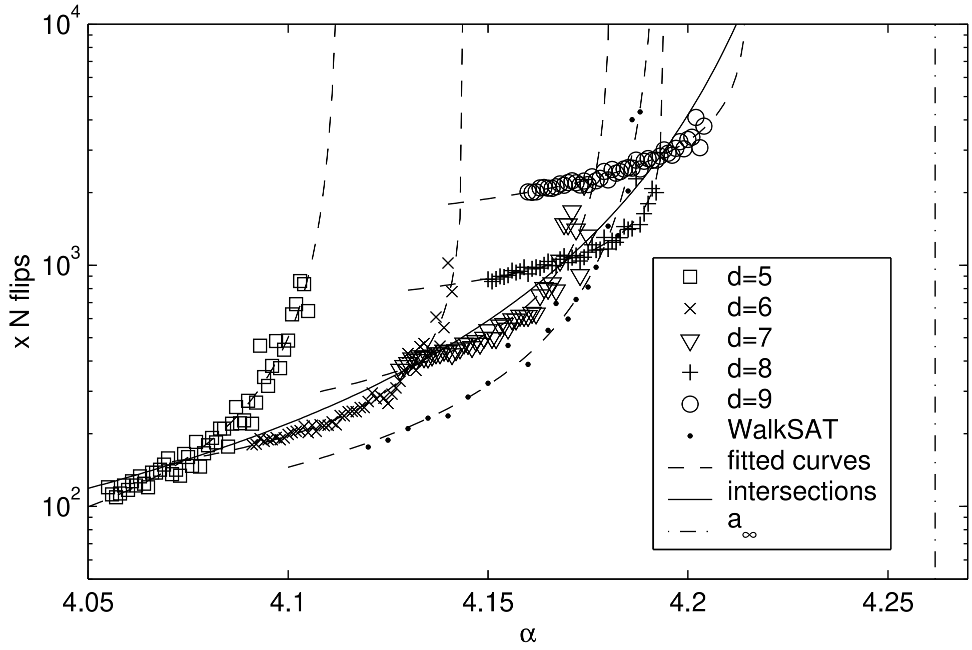

Figure 1: Each point represents the number of flips required by an FRRT run to find a satisfying solution to one randomly generated 3-SAT instance. For each value of d, instances were generated and solved in increasing order of α until the first run which did not reach a solution in a reasonable time. For each d, the logarithm of the average number of flips per variable for different α is fitted to a power law $w_d(a_d - \alpha)^{-z_d}$. From fhe fitted curve one can obtain an estimate for a_d. $N = 10^6$ was used here to approximate the limit case $N \to \infty$. (Estimates: $a_5 \approx 4.11$, $a_6 \approx 4.14$, $a_7 \approx 4.18$, $a_8 \approx 4.19$, $a_9 \approx 4.21$.) The crossing points of these fitted curves were then fitted to a similar power law, which gives an estimate $a_\infty \approx 4.26$. For comparison we have included a few experiments using *WalkSAT-35* [26] with "best" as the chosen heuristics and noise parameter set to 55%. (Note that a flip with WalkSAT is more costly than a flip with FRRT.)

3 Experiments

The results of our experiments on using FRRT to solve random 3-SAT instances for N up to one million are summarized in Figure 1. We believe that results for $K > 3$ would be qualitatively similar; at least for 4-SAT this seems to be the case, based on a few experiments. The experiments with 3-SAT lead us to exciting conclusions. (Similar observations have recently been made concerning the WalkSAT algorithm [3, 29]). It seems that when the deviation d and constrainedness α are held constant, the time needed for FRRT to almost always (a.a.) find a solution grows only linearly with respect to N if $\alpha < a_d$, where a_d is a threshold value for the given deviation d. Instances with $\alpha < a_d$ can be solved a.a. in linear time using FRRT with the given d, and instances with $\alpha > a_d$ get solved almost never in linear time.[1] More precisely, this means that when $\alpha < a_d$, the

[1] When considering relatively small problem instances, the expression "almost always" should of course be used with care. Data in Fig. 1 indicate that the most effective value of the deviation d at $\alpha = 4.2$ is probably 9. We set a limit of $5000N$ flips and tried how large a proportion of 3-SAT instances are solved, when $\alpha = 4.2$ and $N = 1000$. With $d = 9$ the proportion was about 0.02 (15 out of 1000 instances). Using $d = 5$ the proportion was much larger, about 0.5 (507 out of 1000). Generally, the optimal value of d for small instances can be considerably lower than for large instances. When $\alpha = 4.2$ and $N = 10^5$, the proportion solved within $5000N$ flips was about 0.75 (75 out of 100) using $d = 9$. When $d = 5$, none of the 100 instances were solved (there remained 96 to 165 unsatisfied clauses). When $N = 10^6$, all the ten test instances were solved using $d = 9$, with a number of flips ranging from 2919N to 3751N.

probability P of finding a satisfying assignment using $c(\alpha)N$ flips converges towards one as $N \to \infty$ if $c(\alpha)$ is big enough, but when $\alpha > a_d$, P converges towards zero no matter how big $c(\alpha)$ is. There seems to happen a phase transition in the time complexity of the algorithm at the points a_d, from a.a. linear to more than linear (maybe exponential, cf. [3, 29]). When the deviation parameter d increases, the a.a. linear time phase seems to extend further, so that $a_{d+1} > a_d$ for all d.

Specifically, there likely exists for every d a limit function $b_d(\alpha)$ defined in the range $0 \leqslant \alpha < a_d$, so that the average number of flips per variable needed to find a solution converges in probability towards $b_d(\alpha)$ as $N \to \infty$ (cf. Fig. 1). When $d \to \infty$, thresholds a_d cluster at the threshold a_∞, which may or may not be the same as the satisfiability threshold α_c. Extrapolating the results in Fig. 1 gives some support to the possibility that these two threshold values coincide, although we have not been able to find a solution with FRRT for any $N \geqslant 10^6$ instance with $\alpha \geqslant 4.23$. If indeed $a_\infty = \alpha_c$, one can find for every α below α_c a value for the deviation parameter d making almost all random 3-SAT instances with $M = \alpha N$ clauses solvable in linear time by FRRT, although the corresponding coefficient $b_d(\alpha)$ of the linear term likely tends to infinity, as α gets closer to a_c. So if $a_\infty = \alpha_c$, there is a transition from linear time complexity to unsatisfiability in the random 3-SAT problem, without any intervening phase. (As regards the unsatisfiable phase, it is known that proving an instance unsatisfiable by resolution methods almost always takes exponential time [2, 5].)

According to two recent papers [3, 29], such time linearity is typical for random-walk based local search methods such as *WalkSAT* or FRRT, when applied to random 3-SAT instances at small α, and there is a transition from a linear to exponential time phase at some algorithm-dependent "dynamical" threshold α_{dyn}. We tried to estimate α_{dyn} for *WalkSAT-35* [26] with "best" as the chosen heuristic and noise parameter set to 55% (these settings seemed to be near optimal, in accordance with [33]). Based on experiments with million variable instances and the same estimation formula used in Fig. 1, we got an estimate $\alpha_{\text{dyn}} \approx 4.19$, which is quite near the satisfiability threshold α_c but still distinctly apart from it. WalkSAT's noise parameter can be used to tune greediness similarly as FRRT's deviation parameter. There is, however, a major difference: α_{dyn} does not grow monotonically with increasing noise. There is some optimal value for the noise level, beyond which α_{dyn} starts to decrease again.

The recently introduced survey propagation (SP) algorithm [4, 18, 19] for random K-SAT, based on methods from statistical physics and the theory of Bayesian belief networks [34], also has a time requirement that seems to grow nearly linearly with N, when α is fixed. Experiments in [4] seem to confirm this up to at least $\alpha \approx 4.22$. Also SP has a tunable greediness parameter: the proportion of variables that are fixed after each iteration. By decreasing this value towards zero, the efficiently solvable phase can be extended, perhaps even up to α_c. SP is not, however, a local search method and is more complicated to implement than FRRT. It requires more memory, but seems to run somewhat faster than FRRT, especially very near the Sat/Unsat transition (based on our few comparative experiments with $N = 10^6$ [30]). It is possible though that also FRRT can be made more efficient by adding more "intelligence" to the simplistic algorithm.

4 Discussion

Unlike many other local search methods, FRRT does not work by finding repeatedly some local optimum and then escaping to a better one. Instead there exist (nearly) all the time improving flip possibilities, i.e. ones that increase the number of satisfied clauses, right until a complete solution is found. The algorithm rather avoids getting trapped in local optima by letting the current configuration slowly "relax" into a favourable state. The same idea is behind the well-known simulated annealing (SA) method [14, 1]. SA with a suitable cooling schedule can therefore probably achieve comparable efficiency on random 3-SAT instances, if the flips are focused on the unsatisfied clauses. (The basic SA, where the variable to be flipped is picked randomly from among all the variables, is not good at finding solutions for random 3-SAT instances near the Sat/Unsat transition [28].)

This kind of behaviour also suggests that it is more important to have enough time (i.e. enough flips) for the relaxation process rather than a specific value for the deviation parameter. We have made a few experiments (data not shown) with a kind of *restrained* version of FRRT that accepts a flip leading to a configuration with more satisfied clauses than the current "record" only with a small probability p, and behaves otherwise just like the normal FRRT. This way the range of α where a solution can be found in a.a. linear time with a fixed deviation d can be extended considerably beyond the values a_d discussed above. This also indicates that the thresholds a_d are not directly related to how high barriers [13, 25] there are between local optima when $a_{d-1} < \alpha < a_d$ — although the effectiveness of FRRT suggests that typical barrier heights are at most d when $\alpha < a_d$. By restraining the acceptance of new records the search process can be slowed down (i.e. the speed of improving the number of satisfied clauses decreased) also in the beginning of the run, when it is quite fast with standard FRRT no matter how large the deviation parameter is. This may in fact turn out to be necessary for extending the linear time phase up to the satisfiability threshold α_c.

FRRT may work well also on other constraint satisfaction problems that are generated randomly with a uniform distribution, like random K-coloring [22, 31]. One needs to keep in mind, however, that "real world" problem instances do not necessarily have the same simple structure as randomly generated ones, and therefore such results are not by themselves sufficient evidence for the power of FRRT in "real" applications.

References

[1] Aarts, E. & Lenstra, J. K. *Local Search in Combinatorial Optimization.* Wiley (1997).

[2] Beame, P., Karp, R., Pitassi, T. & Saks, M. On the complexity of unsatisfiability proofs for random k-CNF formulas. *Proc. 30th ACM STOC*, 561-571 (1998).

[3] Barthel, W., Hartmann, A. K. & Weigt, M. Solving satisfiability problems by fluctuations: the dynamics of stochastic local search algorithms. Preprint arXiv:cond-mat/0301271 (2003).

[4] Braunstein, A., Mézard, M. & Zecchina, R. Survey propagation: an algorithm for satisfiability. Preprint arXiv:cs.CC/0212002 (2002).

[5] Chvátal, V. & Szemerédi, E. Many hard examples for resolution. *J. ACM* **35**, 759-768 (1988).

[6] Crawford, J. M. & Auton, L. D. Experimental results on the crossover point in random 3SAT. *Artif. Intell.* **81**, 31-57 (1996).

[7] Dubois, O., Boufkhad, Y. & Mandler, J. Typical random 3-SAT formulae and the satisfiability threshold. *Proc. SODA'2000*, 124-126 (2000).

[8] Dueck, G. New optimization heuristics: the great deluge algorithm and the record-to-record travel. *J. Comp. Phys.* **104**, 86-92 (1993).

[9] Friedgut, E. Sharp thresholds of graph properties, and the k-SAT problem. *J. Amer. Math. Soc.* **12**, 1017-1054 (1999).

[10] Garey, M. R. & Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York (1979).

[11] Hoos, H. H. & Stützle, T. Systematic vs. local search for SAT. *Proc. KI-99, LNAI* **1701**, 289-293 (1999).

[12] Kaporis, A. C., Kirousis, L. M. & Lalas, E. G. The probabilistic analysis of a greedy satisfiability algorithm. *European Symposium on Algorithms*, 574-585 (2002).

[13] Kern, W. On the depth of combinatorial optimization problems. *Discr. Appl. Math.* **43**, 115-129 (1993).

[14] Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. Optimization by simulated annealing. *Science* **220**, 671-680 (1983).

[15] Kirkpatrick, S. & Selman, B. Critical behavior in the satisfiability of random Boolean expressions. *Science* **264**, 1297-1301 (1994).

[16] McAllester, D., Selman, B. & Kautz, H. Evidence for invariants in local search. *Proc. AAAI-97* (1997).

[17] Mézard, M., Parisi, G. & Virasoro, M. A. *Spin Glass Theory and Beyond.* World Scientific, Singapore (1987).

[18] Mézard, M., Parisi, G. & Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science* **297**, 812-815 (2002).

[19] Mézard, M. & Zecchina, R. Random K-satisfiability problem: from an analytic solution to an efficient algorithm. *Phys. Rev. E* **66**, 056126 (2002).

[20] Mitchell, D., Selman, B. & Levesque, H. Hard and easy distributions of SAT problems. *Proc. AAAI-92*, 459-465 (1992).

[21] Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B. & Troyansky, L. Determining computational complexity from characteristic 'phase transitions'. *Nature* **400**, 133-137 (1999).

[22] Mulet, R., Pagnani, A., Weigt, M. & Zecchina, R. Coloring random graphs. *Phys. Rev. Lett.* **89**, 268701 (2002).

[23] Östergård, P. R. J. Computer search for small complete caps. *J. Geom.* **69**, 172-179 (2000).

[24] Papadimitriou, C. H. On selecting a satisfying truth assignment. *Proc. FOCS-91*, 163-169 (1991).

[25] Reidys, C. M. & Stadler, P. F. Combinatorial landscapes. *SIAM Review* **44**, 3-54 (2002).

[26] Source code for WalkSAT algorithm: http://www.satlib.org/

[27] Schöning, U. A probabilistic algorithm for k-SAT and constraint satisfaction problems. *Proc. FOCS-99*, 410-414 (1999).

[28] Selman, B., Kautz, H. & Cohen, B. Local search strategies for satisfiability testing. *DIMACS Series in Discr. Math. and Theor. Comp. Sc.* **26**, 521-532 (1996).

[29] Semerjian, G. & Monasson, R. Relaxation and metastability in the random WalkSAT search procedure. Preprint (2003).

[30] Source code for survey propagation algorithm: http://www.ictp.trieste.it/~zecchina/SP/

[31] Svenson, P. & Nordahl, M. G. Relaxation in graph coloring and satisfiability problems. *Phys. Rev. E* **59**, 3983-3999 (1999).

[32] Wang, J. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman (eds.), Complexity Theory Retrospective II, 295-328. Springer-Verlag (1997).

[33] Wei, W. & Selman, B. Accelerating random walks. *Proc. CP-2002, LNAI* **2470**, 216-232 (2002).

[34] Yedidia, J. S., Freeman, W. T. & Weiss, Y. Generalized belief propagation. *Proc. NIPS 13*, 689-695, MIT Press (2001).