Efficient Initial Solution to Extremal Optimization Algorithm for Weighted MAXSAT Problem

Mohamed El-bachir Menai [1] and Mohamed Batouche [2]

[1] Computer Science Department, Tebessa University,
12000 Tebessa, Algeria
bmenai@wissal.dz
[2] LIRE Laboratory, Computer Science Department,
Mentouri University, Constantine, Algeria
batouche@wissal.dz

Abstract. Stochastic local search algorithms are proved to be one of the most effective approach for computing approximate solutions of hard combinatorial problems. Most of them are based on a typical randomness related to some uniform distributions for generating initial solutions. Particularly, Extremal Optimization is a recent meta-heuristic proposed for finding high quality solutions to hard optimization problems. In this paper, we introduce an algorithm based on another distribution, known as the Bose-Einstein distribution in quantum physics, which provides a new stochastic initialization scheme to an Extremal Optimization procedure. The resulting algorithm is proposed for the approximated solution to an instance of the weighted maximum satisfiability problem (MAX SAT). We examine its effectiveness by computational experiments on a large set of test instances and compare it with other existing meta-heuristic methods. Our results are remarkable and show that this approach is appropriate for this class of problems.

1 Introduction

The satisfiability problem (SAT) in propositional logic is known to be *NP*-complete (*NP* denotes the set of all decision problems solvable by a non deterministic polynomial time algorithm) [7] and requires algorithms of exponential time complexity to solve in the worst case : Given a propositional formula, decide whether it has a model. Many problems in artificial intelligence, mathematical logic, computer aided design and databases can be formulated as SAT. The maximum satisfiability problem (MAXSAT) is an extension of SAT and consists of satisfying the maximum number of clauses of the propositional formula. It is known to be *NP*-hard (optimization problem that has a related *NP*-complete decision version problem) even when each clause contains exactly two literals (MAX2SAT). Since finding an exact solution to this problem requires exponential time, approximation algorithms to find near optimal solutions in polynomial time, appear to be viable. Developing efficient algorithms and heuristics for MAXSAT can lead to general approaches for solving combinatorial optimization problems.

P.W.H. Chung, C.J. Hinde, M. Ali (Eds.): IEA/AIE 2003, LNAI 2718, pp. 592-603, 2003.

The present work is encouraged by the recent remarkable results obtained by Boettcher and Percus [5], [6] with their new meta-heuristic method called *Extremal Optimization* (*EO*) on graph partitioning problem. Furthermore, its application to handle unweighted MAXSAT problem instances [17] showed that this method improves significantly previous results obtained with *Simulated Annealing* [14] and *Tabu Search* [10], [11] methods on a bed test of random unweighted MAX3SAT and MAX4SAT instances. In addition, several MAXSAT studies have shown that providing interesting start-up assignments to a local search algorithm can improve its performance.

In the GRASP procedure [19] each iteration consists of a construction phase which provides a start-up assignments to a local search phase. The GRASP solution for MAXSAT [19] is generally significantly better than that obtained from a random starting point.

Boettcher and Percus [6] have enhanced the convergence of EO for the partitioning of geometric graphs by using a clustering algorithm which separates initially the graph into domains. They indicate that EO can perform extremely better when using a *clever* start-up routine.

Szedmak [25] has applied the Bose-Einstein distribution, well known in quantum physics, rather than the uniform one to generate initial solutions for a hill climbing heuristic. The experimental results obtained on unweighted MAXSAT problem instances from the DIMACS repository, prove its efficiency according to a heuristic introduced by Johnson [13].

In this paper, we introduce an algorithm based on the Bose-Einstein distribution which provides a new stochastic initialization scheme to an Extremal Optimization procedure and compare it to the more frequently used methods like WSAT, Simulated Annealing and Tabu Search on an appropriate test set of MAXSAT instances. Finally, we present experimental results which demonstrate the superiority of this new approach with respect to alternative methods on the same benchmark.

2 Local Search for MAXSAT

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of Boolean variables. The set of literals over X is $L = \{x, \bar{x} | x \in X\}$. A clause C on X is a disjunction of literals. A clause form or CNF is a conjunction of clauses. An assignment of Boolean variables is a substitution of these variables by a vector $v \in \{0,1\}^n$. A clause is satisfied by an assignment if the value of the clause equals 1, otherwise the clause is unsatisfied. A weighted formula is a pair $WF = \{CF, W\}$ where $CF = (C_i)_{i \le m}$ is a clause form and $W = (w_i)_{i \le m} \in \mathrm{N}^m$ is an integer vector; for each $i \le m$, w_i is the weight of the clause C_i. An assignment $v \in \{0,1\}^n$ determines a weight value at the weighted formula WF as :

$$wc(CF, W, v) = \sum_{c_i(v)=1} w_i \tag{1}$$

The MAXSAT problem asks to determine an assignment $v_0 \in \{0,1\}^n$ that maximizes the sum of the weights of satisfied clauses :

$$wc(CF, W, v_0) = Max\left\{ wc(CF, w, v) \;\middle|\; v \in \{0,1\}^n \right\} \tag{2}$$

MAX*k*SAT is the subset of MAXSAT instances in which each clause has exactly *k* literals. If each weight is equal to one, we call the resulting problem unweighted MAXSAT, otherwise we speak of weighted MAXSAT or simply MAXSAT.

While finding an exact solution to MAXSAT problem is *NP*-hard, local search has become an important general purpose method for solving satisfiability problems : It starts with a random initial solution and tries to improve it by moving to neighbouring solutions. It can be trapped in local poor minima or *plateaux*, it requires therefore a strategy to escape from these local minima and to guide the search toward good solutions. Different strategies can be cited : Simulated Annealing [14], [24], Tabu Search [10], [11], Ant Colony Optimization [8] and Reactive Search [3]. Although, the list of competitive heuristics is not exhaustive.

In the literature, effective local search algorithms to solve MAXSAT have been proposed. The most popular one is for sure GSAT (for G*reedy* SAT*isfiability*) [21], [22] defined initially for SAT and applied next to MAXSAT. It works as follows : It starts with an initial random assignment and, at each iteration, flips the variable which decreases the number of the most unsatisfied clauses. It is not a pure hill climbing algorithm as it accepts also moves which either produce the same objective function value or increase it. The process is repeated until a maximum number of non-improving moves is reached. Different *noise* strategies to escape from basins of attraction are added to GSAT. Its variants like WSAT [23], Novelty and R-Novelty [16] were generalized to handle weighted MAXSAT problems.

In the Tabu Search method [10], [11], a list of forbidden moves *tabu list* is used to avoid the search process revisiting the previously found solutions. MAXSAT problem was one of its first applications [12], but GSAT seemed to outperform its best results. Different history-based heuristics have been proposed to intensify and diversify the search into previously unexplored regions of the search space with collecting information from the previous phase. HSAT [9] introduces a tie-breaking rule into GSAT so that if more moves produce the same best results, the preferred move is the one that has been applied for the longest period. HSAT can be considered as a version of Tabu Search [15] and the results obtained on some MAXSAT benchmark tasks present a better performance with respect to WSAT.

Simulated Annealing (SA) is a method inspired by natural systems [14], [24]. It emulates the behaviour of frustrated physical systems in thermal equilibrium : A state of minimum energy may be attained by cooling the system slowly according to a temperature schedule. SA local search algorithm moves through the space configurations according to the Metropolis algorithm [18] driving the system to equilibrium dynamics. Selman *et al* [23] affirm that they were unable to find a cooling schedule that outperformed GSAT.

3 Extremal Optimization Method

Extremal Optimization (EO) is a recently introduced meta-heuristic [4] for hard optimization problems. It was inspired by the Bak-Sneppen model [2] which was proposed to describe the self-organization phenomenon in the biological evolution of species. In this model, species x_i have an associated value $\lambda_i \in [0,1]$ called *fitness* and a selection process against the extremely bad ones is applied. At each iteration, the specie having the smallest fitness value is selected for a random update which impacts obviously the fitness of interconnected species. After a sufficient number of steps, the system reaches a highly correlated state known as *self-organized criticality* [1] in which all species have reached a fitness of optimal adaptation.

A general modification of EO [4], [5] noted τ-EO, consists to rank all variables from rank $k = 1$ for the worst fitness to rank $k = n$ for the best fitness λ_k. For a given value of τ, a power-law probability distribution over the rank order *k* is considered :

$$P(k) \propto k^{-\tau}, \; (1 \le k \le n) \tag{3}$$

At each update, select a rank *k* according to $P(k)$ and update the state of the variable x_k. The worst variable (with rank 1) will be chosen most frequently, while the best ones (with higher ranks) will sometimes be updated. In this way, a bias against worst variables is maintained and no rank gets completely excluded. The search process performance depends on the value of the parameter τ. For $\tau = 0$, the algorithm becomes a random walk through the search space. While for too large values of τ, only a small number of variables with bad fitness would be chosen at each iteration and, in this way, the process tends to a deterministic local search. Boettcher and Percus [6] have established a relation between τ, run time *t* and *n* the number of variables of the system to estimate the optimal value of τ. Let $t = An$ where *A* is a constant $(1 << A << n)$, then :

$$\tau \sim 1 + \frac{\ln(A/\ln(n))}{\ln(n)}, (n \to \infty) \tag{4}$$

At this optimal value, the best fitness variables are not completely excluded from the selection process and hence, more space configurations can be reached so that greatest performance can be obtained.

4 Bose-Einstein Extremal Optimization Algorithm (BE-EO)

4.1 Bose-Einstein Distribution

Commonly used methods for MAXSAT problems generate initial assignments randomly over the set of variables that appear in clauses. These variables get their values separately with the same uniform distribution. So, the number of variables with value 1 and those with value 0 in all generated assignments are around the same average. If the optimal assignment contains a very different number from this average, the generated initial assignments will be far away from the optimal one and thus, it will be hard to find this optimum. It turns out that a probability distribution which guarantees that an arbitrary proportion of 1s and 0s will appear in an initial assignment set, can improve the performance of the search algorithm. Szedmak [25] proved that only the Bose-Einstein distribution satisfies the previous condition. He demonstrated its effectiveness by showing that the Hamming distance between the optimal and the initial assignments set, is reduced when the initial assignments are generated by this distribution rather than the uniform one. For the binary case, the Bose-Einstein distribution can be outlined as follows [25] : Let $V = \{1,...,n\}$ be a base set for a given *n*, $X = \{x_1, x_2,...,x_n\}$ be a set of Boolean variables and p_X be the probability distribution of *X* in the space $\{0,1\}^n$. For a subset *S* of *V*, let $X[S] = \sum_{i \in S} x_i$ be the number of the variables of *X* equal to 1 in *S*. We are looking for a distribution such that $X[S]$ is uniformly distributed on $\{0,...,|S|\}$ for all subsets *S* of *V* :

$$\forall k \in \{0,...,|S|\}, \forall S \subseteq V, P(X|S| = k) = \frac{1}{|S|+1} \qquad \textbf{(5)}$$

The Bose-Einstein distribution satisfies the Eqn.5 and it is defined by [20] :

$$\forall X \in \{0,1\}^n, p_X = \frac{1}{(n+1)\binom{n}{X[V]}} \qquad \textbf{(6)}$$

Its conditional probability is given by :

$$p(x_j = 1) = \frac{X[S]+1}{(j-1)+2} \text{ where } S = \{1,...,(j-1)\} \qquad \textbf{(7)}$$

4.2 BE-EO Algorithm

Let us consider a MAXSAT problem instance of *n* Boolean variables and *m* weighted clauses $WF = \{CF, W\}$ where $CF = (C_i)_{i \le m}$ and $W = (w_i)_{i \le m} \in \mathrm{N}^m$. Let *v* be the current assignment. For each variable x_i, the fitness λ_i is defined as the fraction of satisfied clauses weights sum in which that variable appears by the total weights sum :

$$\lambda_i = \frac{\sum_{x_i \in C_j \text{ and } C_j(v)=1} w_j}{\sum_{k=1}^{m} w_k} \qquad \textbf{(8)}$$

The cost function $CS(v)$ is the total cost contributions of each variable x_i. To maximize the sum of the weights of satisfied clauses, we have to minimize $CS(v)$. Hence,

$$CS(v) = -\sum_{i=1}^{n} \lambda_i \qquad (9)$$

The procedure BE-EO for MAXSAT is briefly summarized as follows :

Procedure BE-EO_MAXSAT
Input : *WeightedCl*, *Tho*, *Sample*, *MaxSteps*.
Output : *Vbest*, *CS*(*Vbest*), *UnsatCl*, *WeightUnsat*.
Vbest := Random_BE_Assignment (variables that appear in *WeightedCl*).
TotalWeight := sum of *WeightedCl* weights.
for l := 1 to *Sample* do
    V:= Random_BE_Assignment (variables that appear in *WeightedCl*).
    UnsatCl := set of clauses not satisfied by *V*.
    WeightUnsat := sum of *UnsatCl* weights.
    for k := 1 to *MaxSteps* do
        If *V* satisfies *WeightedCl* then return (*V*, *TotalWeight*).
        Evaluate λ_i for each variable x_i in *WeightedCl* w.r.t. Eqn.8.
        Rank all variables x_i w.r.t. λ_i from the worst to the best.
        Select a rank *j* with probability $P(j) \propto j^{-Tho}$.
        Vc := *V* in which the value of x_j is flipped.
        if *CS*(*Vc*) < *CS*(*Vbest*) then *Vbest* : = *Vc*.
        V := *Vc*.
        Update (*UnsatCl*, *WeightUnsat*).
    endfor
endfor

Given an initial Bose-Einstein assignment *V* generated randomly by the function Random_BE_Assignment. A fixed number of tries *MaxSteps* is executed. Each step in the search process corresponds to flipping the value assigned to a variable according to EO strategy. The best current assignment *Vbest* is related to the current maximum total weights of satisfied clauses. The unsatisfied clauses set *UnsatCl* and their weights sum *WeightUnsat* are then updated. This process is repeated as necessary up to *Sample* times, the size of Bose-Einstein sample of initial assignments.

5 Experimental Results

We now report on our experiment results obtained with a version of the procedures EO and BE-EO. The programs are coded in C language and their behaviours were investigated through experiments performed on a PC Pentium III (256 MB memory, 450 MHz) and conducted both on random weighted and unweighted MAXSAT instances.

Fig. 1. Effect of parameter τ of EO and BE-EO for random unweighted MAX3SAT instances of (*left graph*) n=100, m=500, 700; (*right graph*) n=300, m=1500, 2000

5.1 Problem Instances

The test suite was dedicated to several hard instances : (1) random unweighted MAX3SAT, MAX4SAT, MAXSAT instances, (2) random weighted MAXSAT instances. For random unweighted MAX3SAT and MAX4SAT, the instances considered are respectively (100, 500), (100, 700), (300, 1500), (300, 2000), (500, 5000) and (100, 700), (300, 1500), (300, 3000). For each couple (n, m) of n variables and m clauses, 10 instances were generated. The generator of these random instances is available at the web site `http://www.cs.cornell.edu/home/selman/sat/sat-package.tar`. In addition, we tested 10 random unweighted MAXSAT instances of n=1000 and m=11050, which are available at `http://www-or.amp.i.Kyoto-u.ac.jp/~yagiura/sat/`. Results were also discussed on 10 random weighted MAXSAT instances of n=1000 and m=11050 where clause weights are integers uniformly distributed between 1 and 1000, and available at the same web site. Yagiura and Ibaraki [26] observed that no satisfying assignments usually exist for such instances. We finally tested 17 random instances (identified by `jnh1-jnh19`) of n=100 and m=850 where clause weights are chosen randomly from [1, 1000]. These instances have been obtained from `http://www.research.att.com/~mgcr/data/maxsat.tar.gz`. We ran our experiments on those instances so that our results could be readily compared with those of alternative methods.

5.2 Effect of parameter τ

Our first set of experiments involves the numerical investigation of the optimal value for τ and its impact on the performance of BE-EO. The procedure BE-EO_MAXSAT is run 10 times for each instance where *MaxSteps* is fixed to 5n and *Sample* to 100n.

Fig. 2. Effect of parameter τ of EO and BE-EO for random unweighted MAX3SAT instances of n=500, m=5000 and random unweighted MAXSAT instances of n=1000, m=11050

In EO algorithm, *MaxSteps* is fixed to 1000n and it is run 10 times for each instance. We note that all algorithms have been run with additional iterations, but they have not produced significantly better results. Figures 1-3 illustrate the average error in % of a solution from the upper bound $\sum_{i=1}^{m} w_i$ while varying τ between 0.5 and 1.7. Let $wc(CF, W, v_0)$ be given by Eqn.2, then :

$$error(\%) = \left(1 - \frac{wc(CF, W, v_0)}{\sum_{i=1}^{m} w_i}\right) \times 100 \qquad \textbf{(10)}$$

Results of both procedures show that the optimal values for τ are similar for all random instances ranging from 1.3 to 1.5. Comparable values of parameter τ have been obtained on instances of graph partitioning problem [6]. We can notice that for n=100, the optimal value of τ is, in average, from 1.3 to 1.4 while for the other values (n=300, 500, 1000) it is between 1.4 and 1.5.

5.3 Comparisons

For comparison purposes we examine the performance of BE-EO with that of EO, GSAT, WSAT, SA and SAMD. We refer to results obtained in a previous comparative study executed on SA, SAMD and alternative methods for unweighted MAX3SAT and MAX4SAT instances [12], [23]. GSAT and WSAT codes are taken from the web site `http://www.cs.cornell.edu/home/selman/sat/sat-package.tar`. We experimented with GSAT and WSAT for unweighted instances where the default parameters were used as suggested in the authors implementation (p=0.5, MAXFLIPS=10000) [23]. In the weighted MAXSAT case, we compare our

Fig. 3. . Effect of parameter τ of EO and BE-EO for (*left graph*) random unweighted MAX4SAT instances of n=100, 300, m=700, 1500, 3000; (*right graph*) random weighted MAXSAT instances of n=1000, m=11050 and n=100, m=850

Tables 1, 2 and 3 compare the average error in % (Eqn. 10) of BE-EO with those of considered methods on instances of respectively unweighted MAX3SAT, unweighted MAX4SAT, unweighted MAXSAT of n=1000 and m=11050, and weighted MAXSAT of n=100 and m=850. The data for the first two lines of Tables 1 and 2 are derived from those obtained by Hansen and Jaumard [12] and Selman *et al.* [23], and converted from average number of unsatisfied clauses to average error (%). The first two lines of Table 3 are from the results of Yagiura and Ibaraki [26], they represent both average error (%) and the related total number of iterations on a workstation Sun Ultra 2 Model 2300 (1 GB memory, 300 MHz).

Table 1. Average error for random unweighted MAX3SAT instances

Variables (n)	100	100	300	300	500
Clauses (m)	500	700	1500	2000	5000
SA	1.6400	2.5857	2.0000	2.9000	4.5280
SAMD	1.0200	2.1000	1.0200	1.9500	3.6560
GSAT	0.5560	1.9143	0.5507	1.5970	3.2788
WSAT (p=0.5)	0.5520	1.9143	0.5413	1.6140	3.3400
EO	1.2800	2.1857	0.8067	2.3100	3.5020
τ -EO	0.8200 (τ =1.4)	1.8857 (τ =1.4)	0.6133 (τ =1.5)	1.9100 (τ =1.5)	3.3620 (τ =1.5)
BE-EO	0.6520 (τ =1.4)	1.8810 (τ =1.4)	0.5467 (τ =1.5)	1.5750 (τ =1.5)	3.2024 (τ =1.5)

The results show that τ -EO significantly improves upon the results obtained with SA and SAMD while they trail those of GSAT and WSAT. In Table 1, we can observe that WSAT with p=0.5 gives the best results for (100, 500), (300, 1500) instances and BE-EO gives the best ones for (100, 700), (300, 2000), (500, 5000) instances. In Table 2, the best results for instance of (100, 700) are obtained with WSAT (p=0.5), while BE-EO gives the best ones for instance of (300, 3000). As can be seen in Table 3, all the best results are obtained by the procedure BE-EO which significantly outperforms GSAT, WSAT and τ -EO.

Table 2. Average error for random unweighted MAX4SAT instances

Variables (n)	100	300	300
Clauses (m)	700	1500	3000
SA	0.0429	0.0667	0.4767
SAMD	0.0143	0.0867	0.3133
GSAT	0	0	0.1687
WSAT (p=0.5)	0	0	0.1573
EO	0.0429	0.0800	0.2400
τ -EO	0.0286 (τ =1.3)	0.0667 (τ =1.5)	0.1767 (τ =1.4)
BE-EO	0.0071 (τ =1.4)	0 (τ =1.5)	0.0800 (τ =1.4)

Table 3. Average error for random unweighted and weighted MAXSAT instances

	Unweighted		Weighted		Weighted	
Variables (n)	1000		1000		100	
Clauses (m)	11050		11050		850	
	Error (%)	Iterations	Error (%)	Iterations	Error (%)	Iterations
GSAT	0.5484	5000000	0.7348	15000	0.4890	2000
WSAT	0.4525 (p=0.5)	25000000	0.2733 (p=0.2)	100000	0.0099 (p=0.5)	12264
EO	0.5311	1000000	0.3156	100000	0.0151	10000
τ -EO (τ =1.4)	0.0286	1000000	0.2544	100000	0.0054	10000
BE-EO (τ =1.4)	0.0071	100000	0.2024	100000	0.0021	10000

We can observe that the overall best performance is given by BE-EO. It is at least as good as WSAT for each type of instances and generally it provides better results. The BE-EO high quality solution can have the following explanations. Firstly, MAXSAT search spaces are known to hold too many local minima and the large fluctuations of EO allow search process to explore many of these local minima without loosing well-adapted portions of a solution. Secondly, It is difficult to construct an initial assignment near an optimal one but the Bose-Einstein distribution samples an initial assignments set more efficient than that provided with the usual uniform distribution. Hence, combining EO with this distribution may guarantee that the resulting algorithm efficiently seeks out the region of the fitness landscape containing the global optimum.

6 Conclusion

In this paper, we examined experimentally the effectiveness of sampling initial solutions set to Extremal Optimization search process with the Bose-Einstein distribution. EO is a simple and powerful method to find high quality solutions to hard optimization problems and Bose-Einstein distribution has proved to generate more efficient initial solutions for searching than uniform distribution. A new algorithm called BE-EO, has been proposed to approximate solution of weighted MAXSAT problem instances and computational tests were conducted on both random weighted and unweighted instances. The results provide experimental evidence that this approach is efficient for this class of problems and demonstrate generally its superiority with respect to Simulated Annealing, Tabu Search, GSAT, WSAT and τ-EO heuristic methods. The solution quality achieved by this procedure is due both to the flexibility of EO process to explore more space configurations and to the ability of Bose-Einstein distribution to generate efficient initial assignments so that the number of searching steps needed to reach an optimum is reduced. Furthermore, this new method requires few control parameters, is easy to implement and test, therefore providing motivation to adapt it for solving various classes of *NP*-hard optimization problems.

References

1. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized Criticality : An Explanation of 1/f-noise. Physical Review Letters, V86 N23. (1987) 5211-5214
2. Bak, P., Sneppen, K.: Punctuated Equilibrium and Criticality in a Simple Model of Evolution. Physical Review letters, 59. (1993) 381-384
3. Battiti, R., Protasi, M.: Reactive Search, a History-Sensitive Heuristic for MAX-SAT. ACM Journal of Experimental Algorithmics, Vol. 2, Paper 2 (1997)
4. Boettcher, S., Percus, A.G.: Nature's Way of Optimizing. Elsevier Science, Artificial Intelligence, 119. (2000) 275-286
5. Boettcher, S., Percus, A.G.: Optimization with Extremal Dynamics. Physical Review Letters, V86 N23. (2001a) 5211-5214
6. Boettcher, S., Percus, A.G.: Extremal Optimization for Graph Partitioning. Physical Review E, V64 026114. (2001b) 1-13
7. Cook, S. A.: The Complexity of Theorem Proving Procedures. Proceedings of the 3rd Annual ACM Symposium of the Theory of Computation. (1971) 263-268
8. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System : Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, N1. (1996) 1-13
9. Gent, I.P., Walsh, T.: Towards an Understanding of Hill-Climbing Procedures for SAT. Proceedings of the 11th National Conference on Artificial Intelligence. (1993) 28-33
10. Glover, F.: Tabu Search : Part I. ORSA Journal on Computing 1 (3). (1989a) 190-206

11. Glover, F.: Tabu Search : Part II. ORSA Journal on Computing 2 (1). (1989a) +32
12. Hansen, P., Jaumard, B.: Algorithms for the Maximum Satisfiability Problems. Computing, 44. (1990) 279-303
13. Johnson, D.: Approximation Algorithms for Combinatorial Problems. Journal of Computer and System Sciences, 9. (1974) 256-278
14. Kirkpatrick, S., Gelatt, C.D., Vecchi, P.M.: Optimization by Simulated Annealing. Science, 220. (1983) 671-680
15. Mazure, B., Sais, L., Gregoire, E.: Tabu Search for SAT. Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference. (1997) 281-285
16. McAllester, D., Selman, B., Kautz, H.A.: Evidence for Invariants in Local Search. Proceedings of AAAI'92. MIT Press (1997) 321-326
17. Menai, M.B., Batouche, M.: Extremal Optimization for MAXSAT. Proceedings of the International Conference on Artificial Intelligence (IC-AI'02), Las Vegas, USA. 954-958
18. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21. (1953) 1087-1092
19. Resende, M.G.C., Pitsoulis, L.S., Pardalos, P.M.: Approximate Solution of Weighted MAX-SAT Problems using GRASP. In Satisfiability Problem : Theory and Applications, Vol. 35 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science (American Mathematical Society, 1997). (1997) 393-405
20. Ross, M. S.: Introduction to Probability Models. Academic Press, New York. (2000) 137-141
21. Selman, B., Kautz, H.A.: An Empirical Study of Greedy Local Search for Satisfiability Testing. Proceedings of the 11th National Conference on Artificial Intelligence. (1993a) 46-51
22. Selman, B., Kautz, H.A.: Domain Independent Extensions to GSAT : Solving Large Structured Satisfiability Problems. Proceedings of the 13th International Joint Conference on Artificial Intelligence. (1993b) 290-295
23. Selman, B., Kautz, H.A., Cohen B.: Noise Strategies for Improving Local Search. Proceedings of the 12th National Conference on Artificial Intelligence. (1994) 337-343
24. Spears, W. M.: Simulated Annealing for Hard Satisfiability Problems. In D.S. Johnson and M.A. Trick (eds.), Cliques, Coloring and Satisfiability : Second DIMACS Implementation Challenge, Vol. 26 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science (American Mathematical Society, 1996). (1996) 553-558
25. Szedmak, S.: How to Find More Efficient Initial Solution for Searching ? RUTCOR Research Report, 49-2001, Rutgers Center for Operations Research, Rutgers University. (2001)
26. Yagiura, M., Ibaraki, T.: Efficient 2 and 3-Flip Neighborhood Search Algorithms for the MAX SAT : Experimental Evaluation. Journal of Heuristics, 7. (2001) 423-442