SAT, local search dynamics and density of states

Mériéma Bélaidouni and Jin-Kao Hao

LERIA, Université d'Angers
2 bd Lavoisier, 49045 Angers Cedex 01, France

Abstract. This paper presents an analysis of the search space of the well known NP-complete SAT problem. The analysis is based on a measure called "density of states" (d.o.s). We show experimentally that the distribution of assignments can be approximated by a normal law. This distribution allows us to get some insights about the behavior of local search algorithms.

1 Introduction

In the last decade, many studies intended to understand the dynamics of local search methods in order to contribute to the development of new effective methods for hard problems like the problem of satisfiability (SAT) [7]. Several authors look at the problem structure for an explanation of the behavior of local search methods. Thus studies on the number of solutions [5], the backbone fragility [20], the number and the arrangement of local optima [9, 23], provide a more complete picture of the structure of SAT instances.

However, these studies concern only the computing time for a (local) search algorithm to find an optimal solution, which is only one aspect of local search dynamics. Now, there are many other interesting aspects concerning the dynamics of local search. One of them is the question of the quality of solutions found by a local search algorithm. This important feature was the main object of the autocorrelation measure [11, 22]. Many authors verified the effect of autocorrelation on heuristics performance. Kaufmann [11] and Weinberger [22] show that a downhill algorithm produces solutions of a better quality on NK landscapes if the autocorrelation is increased. Similar conclusions was obtained by Manderick [13] with genetic algorithms applied to NK and TSP landscapes, also by Angel and Zissimopoulos [2] with Simulated Annealing applied to the Graph Bipartioning Problem (GBP).

The work presented in this paper concerns also the quality of solutions and its relation with properties of problems. We are interested in the study of the *long tail* phenomenon observed for several heuristic search algorithms [18]. More particularly, we will investigate an interesting and intriguing effect about this phenomenon. Indeed, our experiments show that the costs generated by a local search algorithm for a given problem instance stagnate invariably within a particular cost interval, independently of the starting point. This behavior cannot be explained using autocorrelation but it can be explained using a new measure of problem structure called *density of states* (d.o.s).

The density of states counts the number of configurations for each cost value of an optimization problem instance. The definition can be extended, to decision problems. Applied to SAT problem for example, d.o.s gives the number of truth assignments for each number of unsatisfied clauses. This measure contains not only the number of solutions [5], but also the number of assignments that satisfy all clauses except one, then the number of assignments that satisfy all clauses except two and so on,...until the number of assignments that satisfy no clauses. This measure is studied in biology [8, 16], physics [12] and optimization [4, 16]. Its approximation can be carried out analytically on some problems [11], by enumeration on small instances [6, 12] or by approximation on large instances [4, 12, 16].

This paper undertakes the study of the 3-SAT problem using d.o.s. It shows through benchmark instances that d.o.s follows, in good approximation, a normal law. This information (given by d.o.s) turns out to be very informative to explain and predict the dynamics of local search methods. In particular, it allows us to give some explanations to the intriguing behavior with the long tail, that we observe in this paper on a Metropolis algorithm.

The article is organized as follows: Section 2 presents the experiment concerning the long tail with a Metropolis algorithm. Section 3 defines the density of states (d.o.s). Section 4 measures d.o.s for various SAT instances and shows its relation with Metropolis dynamics. Section 5 concludes and gives some further directions.

2 Metropolis and the long tail

This section presents an intriguing experiment with Metropolis. The aim of this experiment is to answer the following general questions: Where does a local search begin? where does it end? and does it follow a particular direction (do costs go up or down)? First, let us present the Metropolis algorithm used in the experiment.

2.1 Metropolis

Metropolis [15] method is at the heart of Simulated Annealing [1] which is a method widely used in combinatorial optimisation. For a minimization problem, Metropolis at temperature T (noted MTR(T)) is defined as follows: the process starts with a random configuration s of the search space S; then the process moves from the configuration s to a neighboring configuration s', with a probability p defined by:

$$p = \begin{cases} 1 & if \triangle f < 0 \\ e^{-\triangle f / T} & if \triangle f \geq 0 \end{cases} \tag{1}$$

where $\triangle f = f(s') - f(s)$, and T is a given temperature.

2.2 Experiments with the long tail

The following experiments shed light on a behavior never presented before about dynamics of local search. Let $\mathcal{F}$ be a SAT formula, S the set of truth assignments, f the cost function that associates to each truth assignment $s \in S$ the number of unsatisfied clauses in the formula $\mathcal{F}$ and v the neighborhood relation used in this paper. Two configurations are neighbors according to v if they differ by the value of one variable of an unsatisfied clause. The experiments consist in running **once** Metropolis during **N moves** from two very different initial assignments:

- MTR(T)-I: from a random assignment,
- MTR(T)-II: from a satisfying (optimal) assignment.

Let us consider the instance uf250-01 from SATLIB library [10]. This instance is satisfiable. A solution can be obtained for a reasonable amount of time using Metropolis at different temperatures. Temperature $T = 25$ is one of these successful temperatures.

Our first experiment consists in running Metropolis at temperature $T = 30$ with a random initial assignment during $N = 20.000$ moves. Figure 1 (left) gives the evolution, through time, of the costs generated by MTR(30)-I. We notice that MTR(30)-I starts its evolution with a cost value around $f \approx 120$. After a while, all generated cost values oscillate in the cost interval $[2, 12]$. This last phase generates the so-called (long) tail.

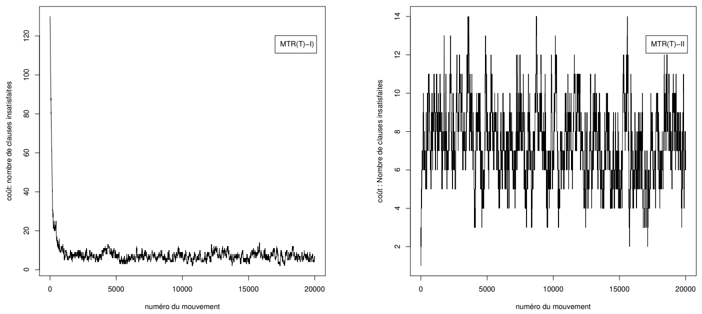

Fig. 1. Evolution of cost values through time for MTR(30)-I (left, from a random assignment) and MTR(30)-II (right, from a satisfying assignment).

Our second experiment consists in running Metropolis at temperature $T = 30$ during $N = 20.000$, but this time, we start the search with a satisfying assignment (for example the solution found with MTR(25)). Metropolis is designed such that it does not stop when it encounters a satisfying solution. It stops only when the running time (in terms of number of moves) is over. Figure 1 (right) gives the evolution, through time, of costs generated by MTR(30)-II. We observe that MTR(30)-II which is initialized at an optimal cost $f = 0$ looses this attribute through time and degrades continually its good starting assignment to finally oscillate in the **same** cost interval as the first experiment: $[2, 12]$.

These experiments are intriguing because they show that the costs generated by Metropolis are strongly attracted by the assignments of some particular cost interval (the cost interval $[2, 12]$ for the instance used in the above experiment) independently of its starting assignment.

Now, let us try to understand why MTR(T) stagnates around a particular cost interval independently of the starting assignment. We believe that this particular behavior find its source (origin) in the density of states of the instance. Therefore, we propose in this paper to examine the relation between the d.o.s and the above long tail phenomenon.

2.3 Process cost density

To establish the link between the d.o.s and Metropolis behavior, we need an intermediate notion which is the *process cost density* (p.c.d). The notion of p.c.d is not new, it is known as the equilibrium density for Metropolis [1]. The process cost density is simply the distribution of costs generated by a search strategy 'after' an infinite running time. Of course, this density is not tractable but it can be approximated in some cases [17]. In what follows, we present an algorithm to approximate the p.c.d of a search strategy ϕ (T) for SAT:

process cost density approximation

Data: $\mathcal{F}$:SAT formula, p: noise, N: Maxflip,
$\phi(T = t)$: search strategy with vector parameters T at t
begin
  run *once* ϕ with vector of parameters T at t on $\mathcal{F}$ during N flips ;
  collect the set of generated costs ;
  count the number $N(f)$ of assignments having the cost value f;
  approximate the process cost density by the frequency distribution $\frac{N(f)}{N}$
end;
Result: p.c.d

Applying this algorithm to the costs generated in the above experiments, we obtain the p.c.d of MTR(30)-I and the p.c.d MTR(30)-II.

Figure 2 (left) displays the p.c.d of MTR(30)-I, whereas Figure 2 (right) displays the p.c.d of MTR(30)-II. The curves obtained have a bell shape. They are centered at $f \approx 7$ and cover the cost interval [2,12]. Our tests show that both curves belong to the same bell shaped distribution.

Consequently, it appears that MTR(30)-I and MTR(30)-II leave their initial cost (high and low) to coincide with a bell shaped distribution. This behavior has been observed for MTR(T) at all temperatures.

2.4 Some facts about p.c.d

In the last section, we have established the link between Metropolis behavior and p.c.d. Now, let us make several remarks concerning p.c.d. First, as p.c.d concerns the probability of occurrence of each cost, it does not depend on their order of occurrence in the run. Therefore, p.c.d is not dependent on a particular run but is common to all runs. Our experiments on SAT instances confirm this

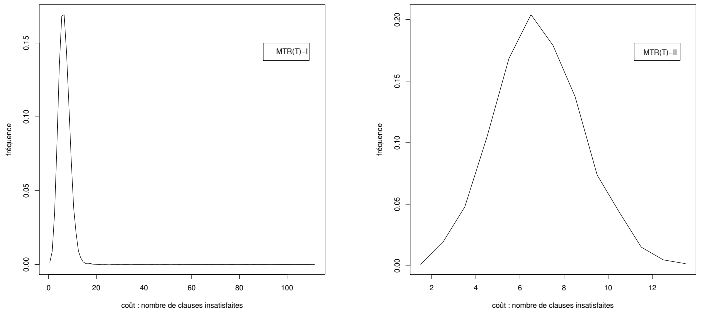

Fig. 2. Process cost density for MTR(30)-I left and MTR(30)-II right.

remark, in very good approximations. Second, the approximation of p.c.d needs a sufficiently long run. If running time is too short the p.c.d, approximation is not sufficiently accurate. The p.c.d approximation is improved by increasing the sample size N.

Up to now, we have linked the long tail behavior of Metropolis with the process cost density and analyzed some aspects of this density. The next step is to understand the relation between p.c.d and d.o.s. Next sections are dedicated to this issue.

3 Approximating Density of States

Process cost density of a search strategy depends on three factors 1) the strategy and its parameters, 2) the given problem instance and 3) the neighborhood relation used by the local search strategy.

In this section, we are concerned by the approximation of d.o.s for different classes of SAT instances. The aim is to 1) answer the question 'Where does the local search begin?' and 2) show how p.c.d is correlated to d.o.s for the Satisfiability problem?

Let $\mathcal{F}$ be a Satisfiability formula, S the set of assignments and f the cost function which associates to each formula $\mathcal{F}$ and truth assignment $s \in \mathcal{S}$ the number of unsatisfied clauses. Assuming the independence of clauses satisfaction. The random variable nuc "number of unsatisfied clauses" follows a binomial law

$$\begin{array}{l} nuc \sim B(nc, p) \\ P[nuc = k] = C_{nc}^{k}.q^{k}.(1-q)^{nc-k} \end{array} \tag{2}$$

where nc is the number of clauses, q the probability for the unsatisfiability of one clause. Thus,

$$E(nuc) = nc.q \tag{3}$$

and,

$$\sigma(nuc) = \sqrt{nc.q.(1-q)} \tag{4}$$

where E(.) and σ (.) are the mean and standard deviation. For k-SAT instances, we have $q = \frac{1}{2^k}$. For instances with variable lengths of clauses, q is averaged over all clause lengths k_1,...,k_l.

$$q = \frac{\sum_{i=1}^{i=l} n_i.q_i}{\sum_{i=1}^{i=l} n_i} \tag{5}$$

and

$$q_i = \frac{1}{2^{k_i}} \tag{6}$$

where n_i is the number of clauses of length k_i.

The binomial deduction is inspired by a study on the random MAX-CSP [21]. In the case of MAX-CSP, the study showed a perfect concordance between theory and practice [3] .

In general, the assumption of satisfaction independence for the clauses is not reliable. This question is discussed in details in the next section.

4 Experiments

In this section, we first aim to approximate d.o.s on different kinds of SAT instances, to determine the law of distribution. Secondly, we show how d.o.s. influences 1) the cost of the initial truth assignment used by the local search method and 2) its process cost density.

4.1 Approximating density of states

Experiments are realized on one instance of each family of SATLIB library [10] (see Table 1 and Table 2). Density of states is approximated by two different methods:

1. Theoretical (analytical) method: under the assumption of clause satisfaction independence, we apply the formulas (1) to (6).
2. Experimental method: we use random selection which consists of taking assignments from the assignments set S in a random and independent manner. The sample size is between 1.000 and 50.000 depending on the studied formula.

Random SAT All instances used in Table 1 are random and satisfiable formulas generated by models of Satisfiability problem. n is the number of variables and nc the number of clauses.

Instances from (a) to (h) are 3-SAT formulas. So, d.o.s estimation can be carried out by formulas (2) and (3) with $q = \frac{1}{8}$. Instances (i) and (j) are variable length formula. So, we apply formulas (2) and (3) with q calculated by formulas

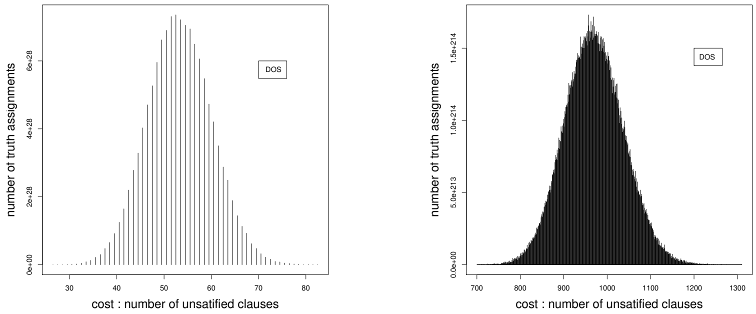

Fig. 3. (left) d.o.s as approximated by random selection for uf100-01. (right) d.o.s as approximated by random selection for *hanoi*4.

	Instances	n	nc	$B(nc,p)$		RandomSelection	
				μ	σ	μ95%Conf.	σ95%Conf.
(a)	uf100-01	100	430	53.75	6.85	$[53.66, 53.75]$	$[6.77, 6.83]$
(b)	uf 200-01	200	860	107.5	9.70	$[107.5, 107.61]$	$[9.72, 9.80]$
(c)	uf 250-01	250	1065	133.12	10.80	$[133.06, 133.20]$	$[11.16, 11.26]$
(d)	$f600$	600	2550	318.75	16.70	$[318.62, 318.83]$	$[16.75, 16.90]$
(e)	$f1000$	1000	4250	531.25	21.56	$[531.18, 531.44]$	$[21.48, 21.67]$
(f)	RTI_k3_n100_m429_1	100	429	53.62	6.85	$[53.60, 53.67]$	$[7.01, 7.07]$
(g)	$BMS\_k3\_n100\_m429\_1$	100	272	34	5.45	$[33.97, 34.03]$	$[4.56, 4.60]$
(h)	$CBS\_k3\_n100\_m429\_b50\_1$	100	429	53.62	6.85	$[53.59, 53.68]$	$[6.78, 6.84]$
(i)	$jnh201$	100	800	45.63	6.55	$[45.62, 45.88]$	$[6.43, 6.61]$
(j)	aim-200-3_-yes1-1	200	680	85.25024	8.63	$[85.23, 85.31]$	$[6.07, 6.13]$

Table 1. D.o.s of random SAT instances according to analytical approximation and random selection.

(4) and (5). We have obtained for example for uf100-01 a mean of $\mu = 53.75$ and a standard deviation of $\sigma = 6.85$.

Now, it remains to confirm the binomial distribution and its parameters by a different method. Concerning the shape, the experimental method described above leads for uf100-01 to a normal distribution as shown in Figure 3 (left). For all other instances (with $nc > 1000$) the agreement with normality has been tested successfully (using a standard normality test). This normality result agrees perfectly with analytical formula since it is possible to approximate a binomial by a normal under certain conditions (conditions satisfied in this case).

Concerning the distribution parameters, the experimental results for all families of random SAT instances of SATLIB are presented in Table 1. We observe that the mean and standard deviation approximated by the experimental method agree with the mean and standard deviation approximated by the analytical method. Thus, according to the experimental method, uf100-01 founds a mean in the confidence interval [53.66,53.75] and a standard deviation in the confidence interval $[6.77, 6.83]$, and the analytical method founds the estimated mean at 53.75 and the estimated standard deviation at 6.85. We remark that the mean belongs to the confidence interval, whereas the standard deviation is sometimes outside the confidence interval without being very far. The difference in the estimation of the standard deviation varies with the instance.

Extension to other instances The following experiments concern satisfiable formula of SATLIB that are issued from other optimization problems (instances (k) to (q) in Table 2). Again, we apply the analytical formula (even if the independence condition is likely unsatisfied) to calculate the mean and standard deviation. Table 2 gives the mean and standard deviation estimated by the analytical method.

Moreover, we approximate mean and standard deviation by random selection. Concerning the shape, experiments show that for hanoi4 the normality remains true as shown in Figure 3 (right). The normality agreement has been tested successfully for all other instances (using a standard normality test). Concerning the distribution parameters, we observe, surprisingly, that the estimated mean found by the analytical method agrees with the confidence interval found by the experimental method, and this for all instances. However it is not the case for the standard deviation. Indeed the analytical and experimental methods found very different values.

	Instances	n	nc	$B(nc,p)$		RandomSelection	
				μ	σ	μ95%Conf.	σ95%
(k)	flat150-1	450	1680	401.2495	17.47614	$[400.54, 401.98]$	$[36.20, 37.21]$
(l)	$ais10$	181	3151	681.7787	23.114	$[681.55, 682.76]$	$[97.91, 98.77]$
(m)	$bwlarge.c$	3016	50457	10988.78	92.71	$[10987, 10991]$	$[127.61, 128.73]$
(n)	$logistics.c$	1141	10719	2012.814	40.43	$[2011, 2013]$	$[127.61, 128.73]$
(o)	ssa7552-038	1501	3575	767.60	24.551	$[767.43, 767.74]$	$[24.34, 24.55]$
(p)	par16-1-c	317	1264	173.25	12.22	$[173.20, 173.32]$	$[9.79, 9.88]$
(q)	hanoi4	718	4934	970.42	27.92	$[969.80, 970.65]$	$[68.00, 68.60]$

Table 2. D.o.s of random SAT instances according to analytical approximation and random selection.

To conclude this section we recall its main results: 1) d.o.s for Satisfiability can be approximated by a normal law; 2) the approximated mean of the analytical approximation and the experimental approximation coincide; 3) the approximated standard deviation of the analytical method and the experimental approximation do not coincide for SAT instances issued from other optimization problems.

However, as the experimental method (random selection) cannot cover all costs in a reasonable amount of time, the accuracy of the above results may be discussed. Improving the experimental estimation of d.o.s needs more elaborate sampling techniques based on Metropolis algorithm [16], for example. Notice that such a sampling technique is usually very time consuming.

Now, let us see the implications of normality of d.o.s for local search. One first point is that finding an assignment that satisfies the maximum number of clauses may be as difficult as finding an assignment that satisfies the minimum

Ins− −tances	MTR(T) $T > t_\infty$		MTR(T) $T = t(t_{opt} < t < t_\infty)$		MTR(T) $T \approx t_{opt}$	
	μ95%Conf.	σ95%Conf.	μ95%Conf.	σ95%Conf.	μ95%Conf.	σ95%Conf.
(a)	[49.36, 49.44]	[6.61, 6.67]	[13.22, 13.24]	[3.34, 3.36]	[3.47]	[1.55]
(b)	[98.28, 98.40]	[9.38, 9.47]	[24.66, 24.69]	[4.65, 4.67]	[3.97]	[1.70]
(c)	[121.05, 121.18]	[10.69, 10.79]	[28.25, 28.28]	[5.10, 5.12]	[4.42, 4.43]	[1.84]
(d)	[292.12, 292.33]	[16.99, 17.13]	[72.18, 72.20]	[7.90, 7.94]	[53.83, 53.98]	[11.22, 11.32]
(e)	[486.41, 486.67]	[20.74, 20.92]	[123.02, 123.08]	[10.24, 10.29]	[20.60, 20.63]	[4.55, 4.56]
(f)	[48.83, 48.92]	[6.60, 6.66]	[11.74, 11.76]	[3.22, 3.24]	[3.24]	[1.59]
(g)	[31.96, 32.02]	[4.47, 4.50]	[11.11, 11.12]	[2.81, 2.83]	[3.46, 4.47]	[1.52, 1.53]
(h)	[49.22, 49.31]	[6.62, 6.67]	[11.88, 11.90]	[3.10, 3.12]	[3.31, 3.34]	[1.93, 1.94]
(i)	[41.73, 41.81]	[6.18, 6.23]	[10.56, 10.58]	[2.94, 2.96]	[2.76, 2.78]	[1.65, 1.67]
(j)	[81.61, 81.64]	[5.96, 5.99]	[36.01, 36.04]	[4.60, 4.62]	(12.25)∗	(3.96)∗
(k)	[301.38, 301.74]	[28.87, 29.13]	[55.42, 55.46]	[6.14, 6.17]	[11.05, 11.06]	[2.88, 2.89]
(l)	[228.15, 228.42]	[42.46, 42.65]	[10.47, 10.49]	[2.73, 2.74]	[3.60, 3.61]	[1.08, 1.09]
(m)	[4667.90, 4673.71]	[466.63, 470.74]	[137.38, 137.46]	[12.97, 13.02]	[10.40, 10.42]	[3.77, 3.78]
(n)	[1075.96, 1076.24]	[71.44, 71.63]	[96.64, 96, 69]	[8.80, 8.84]	[24.88, 21.90]	[4.46, 4.48]
(o)	[711.94, 712.03]	[23.28, 23.35]	[206.35, 206.43]	[12.97, 13.02]	[28.15, 28.17]	[5.01, 5.03]
(p)	[163.44, 163.56]	[9.71, 9.80]	[49.00, 49.04]	[6.63, 6.66]	[11.61, 11.62]	[1.89, 1.90]
(q)	[660.37, 660.94]	[45.98, 46.38]	[54.86, 54.91]	[7.82, 7.85]	[11.10, 11.11]	[2.67, 2.68]

Table 3. Approximation of p.c.d for MTR(T) at different temperatures

number of clauses since d.o.s is symmetric. Other implications are presented in what follows.

4.2 Does d.o.s explain the initial costs?

At this point, one can answer the question 'Where does local search begin?'. Indeed, as d.o.s is normal, random initial assignments have very probably a cost around the mean. For the experiment of Section 2.2 with uf250-01, the initial random cost was 120 which coincides with the mean and the standard deviation of d.o.s ($\mu = 133.12$ and $\sigma = 10.80$). This random initial assignment cost is not that bad, since it is between very good and very bad costs. Another point is that it is very unlikely to select by chance an optimal (satisfying) assignment, since the associated probability is the smallest over all costs.

4.3 Does d.o.s explain the process cost density?

After explaining the effect of d.o.s on the random initial assignment, we look at the relation between the process cost density of the local search strategy and the density of states of the given instance. We propose to link these densities by comparing their distributions and their parameters (μ and σ in our case). In this paper, we will establish the link between p.c.d of Metropolis and d.o.s.

Metropolis To establish the link between p.c.d of Metropolis and d.o.s., we will show that the mean of the p.c.d. goes from the mean of d.o.s to the best costs with the temperature decrease, and standard deviation of the p.c.d starts at the standard deviation of d.o.s and diminishes with the temperature decrease. Our experiments concern Metropolis at three different categories of temperatures. 1) t_∞ a very high temperature 2) t_{opt} a temperature that finds an optimal assignment in a reasonable amount of time and, 3) an intermediate temperature $t_{opt} < T < t_\infty$.

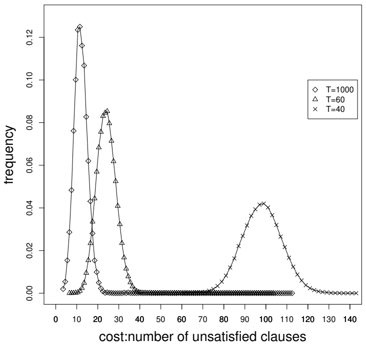

Fig. 4. MTR(T) p.c.d for different (decreasing) temperatures on formula uf200-01.

As shown by Figure 4 the process cost density of MTR(T) is a bell shaped distribution (or slightly asymmetric when p.c.d mean approaches the near optimal area). The mean becomes lower and lower as the temperature decreases (T=1000, T=60 then T=40). Thus, p.c.d is closer and closer to optimal cost areas and so it has more and more chances to reach a satisfying solution. Similarly, the standard deviation becomes smaller and smaller as temperature decreases. This implies that neighboring assignments are more and more similar, which is not in favor of the search process. The bell shaped distribution and the trends for both mean and standard deviation are confirmed for all instances as shown by Table 3.

Now, let us examine the relation between p.c.d of MTR(t_∞) and d.o.s. Firstly, the p.c.d of MTR(t_∞) coincides exactly with the p.c.d of the biased random walk[1]. This is because the temperature is so high that it has no effect: the condition of selection is always satisfied.

Secondly, the p.c.d of the biased random walk is generally different from the p.c.d of the unbiased random walk[2]. This difference is discussed in [3]. Also, the p.c.d of the unbiased random walk coincides exactly with d.o.s. This point is also discussed in [3].

Consequently, the p.c.d of MTR(t_∞) is slightly different from d.o.s. The difference between p.c.d of MTR(t_∞) and d.o.s corresponds to the difference

[1] Biased random walk starts at a random initial assignment and moves from an assignment to another by flipping a variable within an unsatisfied clause.

[2] Unbiased random walk starts at a random initial assignment and moves from an assignment to another by flipping a random variable.

between biased and unbiased random walk. This bias is confirmed by experiments: for example the uf-100-01 has an estimated mean in the confidence interval [53.66,53.75] and an estimated standard deviation in the confidence interval [6.77,6.83] (see Table 1) but MTR(t_∞) has an estimated mean in the confidence interval [49.36,49.44] and an estimated standard deviation in the confidence interval [6.61,6.67] (see Table 3).

In conclusion, there exists an infinity of MRT(T) p.c.d that cover the cost area from d.o.s mean to optimal costs. All these densities are bell shaped curves (or slightly asymmetric when p.c.d mean approaches the near optimal area). The mean of MTR(T) p.c.d starts at the mean of d.o.s (or slightly below if the neighborhood is biased). Then it diminishes as temperature decreases. Similarly, the estimated standard deviation of MTR(T) p.c.d starts at the standard deviation of d.o.s (or slightly below if the neighborhood is biased). Then it diminishes as temperature decreases.

Concerning the extension of this work to the case of a varying temperature interesting observations can be made. Changing the temperature allows to jump from a curve to another, so that the obtained shape for p.c.d depends on the running time allowed for each temperature. If we consider a schedule with two temperatures T=1000 and T=40 and a sufficiently long running time for each temperature, we will obtain the two bell shaped curves of Fig. 4. Otherwise the temperature with too small running time will generate only a piece of the corresponding bell shaped curve. The previous remark could be used to tune objectively simulated annealing algorithm. For example temperature T=1000 should be changed to a smaller one, when the cost f =120 is reached because the process starts its stagnation.

To extent the previous conclusions to other local search methods, we have repeated all the experiments of this study with WSAT [19]. WSAT is one of the best methods for finding solutions to satisfiable formulas. Our experiments concern WSAT(p) with noise parameter from p=1 to p=0.5 on SATLIB instances. We have observed that WSAT(p) presents the same behavior as Metropolis. Indeed WSAT(p) reach the same particular cost interval whatever is the starting solution. Even a near-optimal one. This cost interval can be related to the p.c.d of WSAT(p). We have also observed that the p.c.d of WSAT(p) is a bell-shaped curve whose mean and standard deviation become smaller and smaller as noise decreases. The mean of p.c.d is between the mean of d.o.s and the optimal cost, whereas the standard deviation of p.c.d is between the standard deviation of d.o.s and zero. Therefore, the conclusions concerning the dynamics of Metropolis remain true for WSAT. We conjecture these conclusions would be true for evolutionary algorithms applied to SAT.

5 Conclusions and Perspectives

In this paper, we have analyzed the search space of SAT problem. We have shown on random and structured SAT formulas that the density of states approaches a normal law. This distribution sheds light on some interesting questions related to local search behavior. First, we understand why local search methods like

Metropolis and WSAT are attracted by some cost intervals independently of the cost of the initial assignment. In fact, these costs correspond to an equilibrium cost interval which is determined by the density of states, the neighborhood relation and the local search method itself and its parameters. Second, we learn that the random initial assignment will have a cost around the mean of the d.o.s.

In our future work, we want to reinforce the proof of the relation between p.c.d and d.o.s. We also want to see what happens to p.c.d of WSAT beyond the optimal noise value 0.5 (i.e. between 0.5 and 0). Indeed, in [14] the authors show that the number of resolved formulas decreases. Also, we want to understand the relations between local optima and the dynamics of local search.

Acknowledgments This work was partially supported by the Sino-French Joint Laboratory in Computer Science and Applied Mathematics (LIAMA) and Sino-French Advanced Research Program (PRA).

References

1. E. Aarts and J. Korst. Simulated Annealing and Boltzmann Machines. *John Wiley and Sons*, 1989.
2. E. Angel and V. Zissimopoulos. Autocorrelation Coefficient for Graph Bipartitioning Problem. *Theo. Computer Sience*, 91: 229–243, 1998.
3. M. Belaidouni and J. K. Hao. An Analysis of the Configuration Space of the Maximal Constraint Satisfaction Problem. *Proc. of PPSN VI*, LNCS 1917: 49–58, 2000.
4. J. Bresina, M. Drummond and K. Swanson. Search Space Characterization for Telescope Scheduling Application. *Proc. of AAAI*, 1994.
5. D. A. Clark, J. Frank, I.P Gent, E. MacInttyre, N. Tomov and T. Walsh. Local Search and the Number of Solutions, *Proc. of CP*, 119–133 1996.
6. J. Cupal, I. L. Hofacker and P. F. Stadler. Dynamic Programming Algorithm for the Density of States of RNA Secondar Structures. *Computer Sciences and Biology*, 184–186, 1996.
7. S. A. Cook. The Complexity of Theorem proving procedures. *Proc. of ACM Symp. Theo. Comput.*, 151–158, 1971.
8. J. Cupal, C. Flamm, A. Renner and P.F. Stadler. Density of States, Metastable States and Saddle Points Exploring the Energy Landscape of an RNA molecule. *Proc. Int. Conf. Int. Sys. Mol. Biol.*, 88–91, 1997.
9. J. Frank, P. Cheeseman, and J. Stutz. When Gravity Fails: Local Search Topology. *(Electronic) Journal of Artificial Intelligence Research.*, 7: 249–281, 1997.
10. H. H. Hoos and T. Stutzle *Proc. of SAT, IOS Press.*, 283–292 , 2000.
11. S.A. Kauffman. Adaptation on Rugged Fitness Landscapes, *Lectures in the Sciences of Complexity SFI Studies in the Sciences of Complexity, ed. D. Stein, Addison-Wesley Longman*, 527–618, 1989.
12. S. Kirkpatrick and G. Toulouse. Configuration Space Analysis of Travelling Salesman Problem. *J. Physics*, 46: 1277–1292, 1985.
13. B. Manderick, B. de Weger, and P. Spiessens. The Genetic Algorithm and the Structure of the Fitness Landscape. *Proc. of ICGA*, 143–150, 1991.
14. D. McAllester, B. Selman and H. Kautz. Evidence for Invariants in Local Search. *Proc. of AAAI*, 321–326, 1997.
15. N. Metropolis, A.W. Rosenbluth, M.N Rosenbluth and A.H. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21: 1087–1092, 1953.

16. H. Rosé, W. Ebeling and T. Asselmeyer. Density of States - a Measure of the Difficulty of Optimization Problems. *Proc. of PPSN IV*, LNCS 1141: 208–226, 1996.
17. G.R. Schreiber and O.C. Martin. Procedure for Ranking Heuristics Applied to Graph Partitionning. *Proc. of MIC*, July 1997.
18. B. Selman, H.A. Kautz. An Empirical Study of Greedy Local Search for Satisfiability Testing. *Proc. of AAAI*, 1993.
19. B. Selman, H.A. Kautz and B. Cohen. Noise Strategies for Improving Local Search. *Proc. of AAAI*, 1994.
20. J. Singer, I. P. Gent and A. Smaill. Backbone Fragility and Local Search Cost Peak. *Journal of Artificial Intelligence Research*, 235–270, May 2000.
21. G. Verfaillé. CSP Aléatoires: Estimation de l'Optimum, *Research Notes*, 1998.
22. E. Weinberger. Correlated and Uncorrelated Fitness Landscape and How to Tell the Difference. *J. Biol. Cybern.*, 63: 325–336, 1990.
23. M. Yokoo. Why Adding More Constraints Makes a Problem Easier for Hill-Climbing Algorithms: Analyzing Landscapes of CSPs. *Proc. of CP*, LNCS 1330: 357–370, 1997.