Landscapes of the Maximal Constraint Satisfaction Problem

Meriema Belaidouni[1,2] and Jin-Kao Hao[2]

[1] LGI2P, EMA-EERIE, Parc Scientifique Georges Besse, F-30000 Nîmes
[2] LERIA, Université d'Angers, 2 bd Lavoisier F-49045 Angers Cedex 01
{meriema@eerie.fr, Jin-Kao.Hao@univ-angers.fr}

Abstract. Landscape is an important notion to study the difficulty of a combinatorial problem and the behavior of heuristics . In this paper, two new measures for landscape analysis are introduced. These measures are based on Hamming distance of iso-cost levels. Sampling techniques based on neighborhood search are defined in order to carry out approximation of these measures. These measures and techniques are used to analyze and characterize the properties of random landscapes of the Maximal Constraint Satisfaction Problem.

1 Introduction

Large combinatorial problems are often hard to solve since such problems may have a huge search space. To tackle a large combinatorial problem, heuristics such as neighborhood search methods constitute one of the most powerful approaches. Though heuristics have proven to be very successful, there are few studies allowing to explain such a performance. The notion of landscape is among the rare existing concepts which help to understand the behavior of heuristics and to characterize the difficulty of a combinatorial problem.

The landscape concept has been first introduced by Wright in 1932 [17]. Since then, the term has been re-used by several researchers, with sometimes different meanings [5, 11]. Numerous measures have been proposed to analyze landscapes and to understand their difficulty for heuristics. In what follows we introduce the landscape notion and its measures. To be general, we first define search space.

1. **Search Space**: Given a combinatorial problem P, a search space associated to a mathematical formulation of P is defined by a couple (S, f) where S is a finite set of configurations and f a cost function which associates a real number to each configuration of S. Only a small number of measures are available for this structure. The minimum and the maximum costs are the two most common.
2. **Search Landscape**: Given a search space (S, f), a *search landscape* is defined by a triplet (S, v, f) where v is a neighborhood function which verifies $v : S \rightarrow 2^S - \{\emptyset\}$. This landscape, also called *energy landscape* in [5], can be considered as a "neutral" one since no search process is involved. This landscape can be conveniently viewed as a weighted[3] graph $G = (S, v, f)$.

[3] Note that the weights are defined on the nodes, not on the edges.

Search landscape has been the subject of some studies. Several measures have been defined: the number and distribution of *local minima*[4] [13, 10], and *autocorrelation* which quantifies the ruggedness of a landscape, *i.e.* the variation of the cost values between the neighbors in the graph [12, 16].

3. **Process Landscape**: Given a search landscape (S, v, f), a *process landscape* is defined by a quadruplet (S, v, f, ϕ) where ϕ is a search process. The process landscape represents a particular view of the neutral landscape (S, v, f) seen by a search process. This notion of landscape was first introduced in [11] for the purpose of studying the genetic operators of a genetic algorithm. *Autocorrelation of a random walk* [14] and *fitness distance correlation* (FDC) are well known measures for process landscape [11].

Though these measures produce interesting and useful information about a landscape, it seems that they give only a partial picture of the landscape. In particular, these measures fail to answer such important questions as: 1) How many configurations in the search space are there for a given cost? 2) Are the configurations in the search space scattered or gathered in some specific areas? 3) What is the "accessibility" of a configuration from another one?

Thanks to a measure called "density of states" recently proposed in [1], the first question is now answered. This measure, belonging to search space level, gives the number of configurations per cost value. In a previous study [4], we applied this state density measure to study the search space of random instances of the Maximal Constraint Satisfaction Problem (MAX-CSP). The estimation of state density reveals that the configurations of a given random MAX-CSP instance follow a gaussian distribution. This distribution constitutes an explanation of landscape difficulty for heuristics. Indeed, configurations are concentrated in certain cost zones attracting the heuristic and hindering its evolution towards less dense cost zones. Another important point is that this measure allowed us to introduce a stratified model for the search space (see Fig. 2 of Section 6). In this model, each cost value c is associated to the set of configurations having the cost c, which we call an iso-level. Simple distances based on cost values are also defined for the model.

In this paper, we try to answer the second and the third questions. Based on the above iso-cost level model, two new measures are proposed for landscapes: Hamming Distance In a Level (HDIL) and Hamming Distance Between Levels (HDBL). HDIL measures the similarity (or diversity) of configurations within an iso-cost level. This distance translates in some sense the idea of the "width" of the landscape. Similarly, the distance between iso-cost levels (HDBL) measures the accessibility of configurations and reflects in some sense the "length" of the landscape. These two measures together give another picture of a landscape.

At this stage, we notice that, there are some similarities between our work and the recent study concerning neutral networks reported in [2], though the two studies have quite different objectives and estimation techniques. Indeed, the model of neutral networks is close to the iso-cost level model. Measures such

[4] $s \in S$ is a local minimum with respect to the neighborhood v if $\forall\ s' \in v(s)$, $f(s) \leq f(s')$.

as neutral dimension and percolation index share some common ideas with our measures HDIL and HDBL.

Like in [4], we use the very general Maximal Constraint Satisfaction Problem as a test problem for our experimentation. The measures HDIL and HDBL are applied to landscapes corresponding to random MAX-CSP instances. Experimental results show that the configurations of a given iso-cost level are separated by a large distance. The results show also that the distance between iso-cost levels change depending on whether high cost or low cost levels are considered.

This paper is organized as follows: Section 2 defines our new measures. Section 3 introduces sampling techniques. After a recall of the MAX-CSP in Section 4, experimental results on random instances are given and discussed in Section 5. Finally, conclusions and future work are presented in Section 6.

2 Measures and Approximations

In this section, we will introduce two new measures for studying landscapes. We define *Hamming distance in a level* (HDIL) which measures the average variation of distances for configurations having the same cost. Then we define *Hamming distance between levels* (HDBL) which measures the average distances to go from a configuration to another one having a different cost. These two measures translate respectively the width and the length of a landscape.

2.1 Generalities

Given a search space (S, f), the neighborhood relation N_1: $S \rightarrow 2^S - \{\emptyset\}$ and the Hamming Distance are defined as follows:

- **Neighborhood** N_1
  N_1 is defined as follows: two configurations $s(s_1 ..., s_n)$ and $s'(s'_1 ..., s'_n)$ involving n components (variables) are neighboring if they differ by a single value of a variable; more precisely, if and only if $\exists!(i,j)/s_i \neq s'_j$.
- **Hamming Distance**
  Hamming distance associated to this neighborhood is:

$$d_H(s, s') = d_H(s', s) = \sum_{i=1}^{n} \delta(s_i, s'_i) \qquad (1)$$

  where $\delta(x, y) = \begin{cases} 1\ if & x \neq y \\ 0\ otherwise \end{cases}$
  The Hamming distance $d_H(s, s')$ between two configurations s and s' counts the number of components having different values between s and s'.
- **iso-cost level**
  $C \subset S$ is an iso-cost level of cost $c \Leftrightarrow \forall s \in (C), f(s) = c$.

2.2 Hamming Distance In an iso-cost Level - HDIL

The distance in a set A can be defined by the average distance between the elements of A.

$$D(A) = \frac{1}{|A^2|} \sum_{(s,s') \in C^2} d(s, s') \tag{2}$$

In particular, the *Hamming Distance In an iso-cost Level* C is defined by the distance in a set with $A = C$ and $d = d_H$. It corresponds to a measure of landscape width. This distance, denoted by $D(C)$ represents the diversity (or similarity if one replaces δ by $\delta' = 1 - \delta$) of configurations of C. The HDIL measure differs from autocorrelation by the fact that it measures the variation of distance for a fixed cost while the autocorrelation measures the cost variation for a fixed distance. This measure has two advantages : it concerns all costs areas and it is not dependent on instance's size.

2.3 Hamming Distance Between iso-cost Levels - HDBL

The Hamming distance between two sets A and A' can be defined by the average distance between the elements of A and A'.

$$D(A, A') = \frac{1}{|A \times A'|} \sum_{(s,s') \in A \times A'} d(s, s') \tag{3}$$

We define the *Hamming Distance Between iso-cost Levels* C and C' by the Hamming distance between sets with $A = C$, $A' = C'$ and $d = d_H$. This distance is denoted by $D(C, C')$ hereafter. This measure could give very large distances and has little meaning if it is used to study the behavior of search algorithms. The "vertical" Hamming Distance between two levels, defined in the following formula, seems to be more relevant.

$$D_{CC'} = \frac{1}{|C|} \sum_{s \in C} d(s, s') \tag{4}$$

where $s' \in C'$ is a configuration reachable from s by a search process ϕ. Thus, this distance concerns configurations 'linked' by a search process. The measure $D_{CC'}$ informs us about the distance between two successive levels C and $C + \delta C$. Using this measure, it is now possible to know the potential ancestor of a configuration. Note that the measure $D_{CC'}$ is different from FDC since $D_{CC'}$ concerns distance variations while FDC concerns the relationship between distance variation and cost variation. Moreover, in $D_{CC'}$, the distance is measured with respect to a set of configurations, whereas FDC has a single reference configuration.

2.4 Estimators

Because of the large number of configurations of a given cost value c, it is difficult to calculate exactly the distances $D(C)$, $D(C,C')$ and $D_{CC'}$. We can nevertheless approximate these values by using sufficiently large and representative subsets $E \subset C$ and $E' \subset C'$. The corresponding estimators $\hat{D}(C)$, $\hat{D}(C,C')$ and $\hat{D}_{CC'}$ are defined as follows:

$$\hat{D}(C) = \frac{1}{|E^2|} \sum_{(s,s') \in E^2} d_H(s,s') \tag{5}$$

$$\hat{D}(C,C') = \frac{1}{|E \times E'|} \sum_{(s,s') \in E \times E'} d_H(s,s') \tag{6}$$

$$\hat{D}_{CC'} = \frac{1}{|E|} \sum_{s \in E, \psi(s) \in E'} d(s, \psi(s)) \tag{7}$$

where $\psi(s)$ is the first configuration of E' encountered by the search process ψ.

3 Sample Construction

Sample construction is a decisive stage for approximation validity. As our measures are related to the level C of a cost c, we need a sampling process which is able to reach configurations of the cost c. Moreover, the sampling process must be able to reach sufficiently diversified configurations in order to constitute a representative sample E of C. Theoretically, random search may be an interesting sampling process. However, random search is not viable in practice since it is unable to reach configurations in low cost areas (for minimization). In this work, we adopt neighborhood search methods including Metropolis, Simulated Annealing and Tabu Search to reach low cost areas. Let us notice that these methods have been used successfully to resolve the MAX-CSP problem [7, 9]. Two sampling techniques are proposed, the first technique may use any "neighborhood heuristic", while the second technique is based on Metropolis algorithm.

3.1 Sampling with Re-run

Sampling with re-run consists to build a representative sample by keeping only one configuration per run (execution) of any neighborhood heuristic. Thus, to build a sample E, at least $|E|$ runs are needed. Indeed, we notice that a single run of any neighborhood search process does not gather a sufficient number of configurations of a given cost. In order to increase this number we run the heuristic (Tabu, Simulated Annealing and Metropolis) as many times as necessary until a sample of sufficient size is obtained.

This simple technique does not require any tuning effort (except to reach the wanted cost value). Moreover, as configurations of the sample come from different runs, this method insures the configurations independency. However, this technique may be very time consuming.

3.2 Sampling without Re-run

Sampling without re-run consists to build a sample of a given cost c in only one run (execution) of a given "neighborhood heuristic". This task is not easy because most of time, neighborhood search methods generate only few configurations for a given cost value in one run.

However, there exists a neighborhood heuristic for which a special phenomenon occurs. As shown in [4], Metropolis applied on MAX-CSP random instances stagnates around an average cost value which depends only on the temperature used. Fig. 1 shows the phenomenon with different temperatures $T = \{35, 15, 5, 0.5\}$. For example, for $T = 0.5$, 0.075% of the encountered configurations in one run have a cost of 32. Therefore, in order to give a sample of size K with a cost of 32, it is sufficient to run the Metropolis process with $T = 0.5$ for N iterations ($K = 0.075\% \times N$).

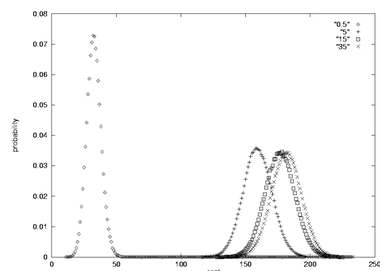

Fig. 1. Metropolis for different temperatures

The technique of sampling without re-run can be very useful: it is a fast method (only one run is sufficient). However, the sampled configurations are not completely independent.

In what follows, we define the Maximal Constraint Satisfaction Problem - a very general optimisation formalism. We then go on to apply the proposed measures to comparing binary MAX-CSP landscapes.

4 Maximal Constraint Satisfaction Problem (MAX-CSP)

4.1 Problem Definition

The MAX-CSP can be conveniently defined with the notion of *constraint network*. A constraint network CN is a triplet $< V, D, C >$ where:

- V = $\{V_1, V_2 ... V_n\}$ is a finite set of variables;
- D = $\{D_1, D_2 ... D_n\}$ is a finite collection of value domains associated to the variables;

– C = $\{C_1, C_2...C_n\}$ is a set of constraints, each being a subset of the Cartesian product of the domains of some variables specifying allowed (or forbidden) value combinations.

Given such a constraint network $CN < V, D, C >$, the MAX-CSP is to find an assignment of the values of D to the variables of V such that the number of satisfied constraints is maximized [6]. In practice, the equivalent minimization version is often used, *i.e.* one minimizes the number of violated constraints instead of maximizing satisfied constraints. The MAX-CSP can be formulated as a couple (S, f) where S is the set of all possible assignments of values of D to the variables of V and f is the number of unsatisfied constraints of C. Note that the classical Constraint Satisfaction Problem (CSP) is a special case of MAX-CSP for which the optimal cost $f^* = 0$.

5 Experimentation

5.1 Random Instances Generation

Test instances used in this work correspond to random, binary constraint networks generated according to a standard model [15]. A network class is defined by $< n, d, p_1, p_2 >$ which has n variables, d values per variable, $p_1.n.(n-1)/2$ constraints taken randomly from $n.(n-1)/2$ possible ones (p_1 is called the density), and $p_2.d^2$ forbidden pairs of values taken randomly from d^2 possible ones for each constraint (p_2 is called the tightness). For each given class $< n, d, p_1, p_2 >$, different instances can be generated using different random seeds.

A constraint network may be under-constrained, or over-constrained. A phase transition in solubility occurs in between when the network is critically constrained [15]. Under-constrained networks tend to be easily satisfiable (cost $f^* = 0$). Over-constrained networks are usually unsatisfiable (cost $f^* > 0$). Critically constrained networks may or may not be satisfiable and are usually hard to solve from a satisfaction point of view. These different regions are characterized by a factor called *constrainedness* [8]:

$$\kappa = \frac{n-1}{2} p_1 \log_d\left(\frac{1}{1-p_2}\right)$$

$\kappa = 1$ delimits under- ($\kappa < 1$) and over- ($\kappa > 1$) constrained networks. Networks with $\kappa \approx 1$ corresponds to critically-constrained ones. For the purpose of this work, we use over- ($\kappa > 1$) or critically ($\kappa \approx 1$) constrained networks.

Two distance measures are applied on the search landscape (S, v, f) of random MAX-CSP instances. HDIL and HDBL are calculated and analyzed for various cost values. The random instance $< 100.10.15.25 >$[5] is used as an example. This instance has a known optimal value of (f^*)=0 [9] and a central level of cost $C_m = 186$ [4]. The MAX-CSP landscapes are supposed to be isotropic, the calculations confirm this assumption.

[5] This instance is generated with a random seed equal to 3.

5.2 Hamming Distance In an iso-cost Level

The purpose of the following experiment is to estimate Hamming Distances in various levels of the studied instance $< 100.10.15.25 >$. Sampling with re-run with Simulated Annealing (SA) and Tabu Search (TS) is used to sample configurations of given costs. Preliminary tests were carried out to determine the number of configurations in a sample E. It has been found that 5000 configurations are sufficient to make the statistics stable. Therefore, both SA and TS are run 5000 times for each fixed cost of the target cost level. Table 1 gives the results of TS for different cost levels running from the average instance cost 186 to the cost 12. For each level, we count the number of execution (called size) that meet the considered level among the 5000 tries, and its Hamming distance. The experiments are repeated 10 times. The first thing one can notice is that some re-runs failed to reach configurations of the wanted cost. For example, for 5.000 executions, TS reaches the cost value 183 only 332 ± 19 times. However, the size increases when cost value decreases. It reaches 4932 ± 4 for cost level 12. Concerning the Hamming distances, large magnitudes (Table 1) are obtained. Similar results are observed for costs lower than 12.

In the same way, Table 2 gives the results obtained with SA. Once again, we observe large magnitudes for Hamming distances. Similar results are obtained by the sampling without re-run using Metropolis. The Hamming distances are very close to those generated by Simulated Annealing Table 2. Similar results are observed for costs lower than 12.

At this stage, some conclusions can be made. The measure of HDIL discloses some interesting aspects of the landscape for random instances. A so large width (HDIL) within a level shows that configurations are not clustered in one group (which does not mean that they do not belong to large clusters). One implication of this result is that a single configuration can not represent a whole level. Another point is that, HDIL obtained over one run are always smaller than those obtained by our sampling method therefore our results can be considered as upper bounds on the considered instance. Moreover, Hamming distances obtained by Simulated Annealing and Metropolis are smaller than those obtained by Tabu as showed in Table 1 and Table 2. Simulated Annealing and Metropolis reach more frequently the desired (low) cost levels. Therefore, Tabu explores a larger low cost area of the landscape. In addition to these considerations, one should notice that though Tabu search is faster than Metropolis and Simulated Annealing, all these calculations are very time consuming (Table 1 and Table 2).

5.3 HDIL on other instances

We have measured Hamming distance in an iso-cost level of different random MAX-CSP instances (Table 3). These instances have sizes ranging from 100 to 300 variables with κ around 1. For each instance, we give the near optimal cost value (f^*), the central cost value (C_m), the Hamming distance in a level of cost

cost	Tabu(30) 12	32	82	125	159	183
size	4932 ± 4	4766 ± 13	1902 ± 31	1219 ± 25	906 ± 19	332 ± 19
distance	0.862 ± 10^{-3}	0.882 ± 10^{-3}	0.895 ± 10^{-3}	0.898 ± 10^{-3}	0.899 ± 10^{-3}	0.899 ± 10^{-3}

Table 1. Hamming Distance In an iso-cost Level (HDIL) by sampling with re-run (Tabu)

cost	Simulated Annealing 12	32	82	125	159	183
size	4738	4260 ± 30	3465 ± 26	2948 ± 30	2519 ± 36	1303 ± 42
distance	0.8188	0.851 ± 10^{-3}	0.886 ± 10^{-3}	$0.896 \pm 10.^{-3}$	0.899 ± 10^{-3}	0.899 ± 10^{-3}

Table 2. Hamming Distance In an iso-cost Level (HDIL) by sampling with re-run (Simulated Annealing)

C_m $(D(C_m))$. We applied sampling with re-run method using Tabu search to approximate HDIL.

We observe from Table 3 that for all instances $HDIL$ is about 0.9, which corresponds to a large diversity. This consistency may have something to do with random generation. Even if this measure may be conditioned by the approximation technique used, it seems possible that the original value is large.

	κ	f^*	C_m	$D(C_m)$	$D(\frac{C_m - f^*}{2})$	$D(f^*)$	D_{f^*,C_m}
(a) 100.10.15.25	0.93	0	186	0.89	0.89	0.86	52
(b) 100.10.20.25	1.24	19	248	0.90	0.89	0.85	54
(c) 100.15.10.45	1.64	11	223	0.93	0.93	0.90	55
(d) 100.15.20.30	1.3	25	297	0.93	0.93	0.90	60
(e) 100.15.30.20	1.22	18	297	0.93	0.93	0.89	56
(f) 200.20.20.15	1.08	21	597	0.94	0.94	0.93	119
(g) 200.20.18.14	0.9	0	501	0.95	0.94	0.93	112
(h) 300.20.10.18	0.99	14	807	0.95	0.94	0.93	168
(i) 300.30.16.12	0.9	4	861	0.96	0.96	0.96	172

Table 3. HDIL on other instances

5.4 Hamming Distance Between iso-cost Levels

Hamming distance between levels expresses the length of a landscape. We would like to answer questions like: can one estimate the distance between two levels? If yes, can one draw up a distance map? Or are levels at equal distances? Can one know the potential ancestors of a given configuration? Do these distances change with the search heuristic?

The goal of this experiment is to compute the Hamming distances between levels for the descent using sampling with re-run. Let us recall that, the descent process begins with a random initial configuration s_0, then at each iteration, moves to a better neighbor. The process stops when a local optimum is reached. We run 1000 executions of this strict descent algorithm and we compute distances between levels in the form of a map of distances using formula (7). We analyze the results on two levels areas. The area (A) (Table 4, left) corresponds to a high cost area (around the average cost level $C_m = 186$). The area (B) (Table 4, right) corresponds to a low cost area. The results are interpreted as follows. For $\hat{D}_{179,177}$, if the search process wants to go from a configuration of cost 179 to a configuration of cost 177, a distance of 1.3 is necessary to make this change. The results shows that for a given cost value variation, configurations are more separated in lower cost levels. For instance, by comparing the distances $\hat{D}_{179,177}$ and $\hat{D}_{39,37}$ which both have a cost variation of 2, we observe that $\hat{D}_{179,177} < \hat{D}_{39,37}$. This comparison is valid for other distances between levels.

The increase of Hamming distance between levels shows the flat structure of this landscape in low cost areas.

$\hat{D}_{C,C'}$	177	178	179	180	181	182	183
179	1.3	1	0	1	1.4	1.7	2.2
180	1.8	1.3	1	0	1	1.3	1.7
181	2.2	1.8	1.4	1	0	1	1.4

$\hat{D}_{C,C'}$	37	38	39	40	41	42	43
39	1.9	1	0	1	1.8	2.8	3.6
40	2.8	1.9	1	0	1	1.9	2.7
41	3.6	2.8	1.8	1	0	1	1.9

Table 4. Distance map between levels for high and low cost areas

5.5 HDBL on other instances

Similar calculi have been carried out for the instances (a) to (i) of Table 3. Given the fact that the computing of HDBL distance map is quite long, we limited our computing to the distance between a cost level C_m and the cost level f^*. We used the sampling technique with re-run with the following "reversed" random walk: one begins with an optimal solution and then makes a random walk at each step.

Again, the obtained HDBL results look rather similar to those for instances with the same number of variables. This consistency may have something to do with random generation. At the same time, $C_m - f^*$ is not the same for all instances. Therefore, $\frac{D_{f^*,C_m}}{C_m - f^*}$ may be used to compare instances. However, our actual sampling processes are not fast enough to have a precise approximation of this measure.

6 Conclusions and Future Work

In this paper, we have proposed two measures called *Hamming Distance In a Level - HDIL* and *Hamming Distance Between Levels - HDBL* to analyze the search landscape (S, v, f). Both distances concern links between components of configurations. However, Hamming distance in a cost level quantifies the diversity of configurations or the width of a landscape whereas Hamming distance between cost levels translates the accessibility of configurations or the length of the landscape. Hamming distances in a level and between levels can be represented as follows Fig. 2: .

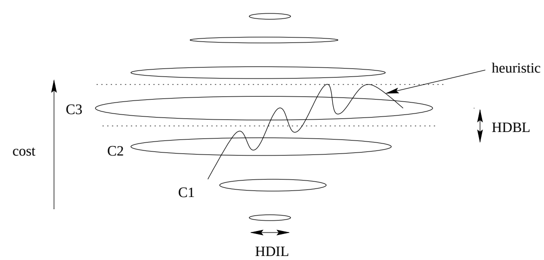

Fig. 2. Iso-cost level model for search space (S, f)

To estimate these distances, sampling techniques based on neighborhood search were developed. Notice that depending on the required precision, the approximation of the HDIL and HDBL measures may be quite time consuming.

These measures have been applied to a number of random MAX-CSP instances. Experimental results showed that within a level, the configurations have a large distance (about 0.9 for a maximum of 1). This distance remains stable for different cost levels.

Experiments with HDBL disclose that the distance between two different cost levels of high costs is smaller than the distance between two levels of low costs. Moreover, these distances change for different instances.

Such information may be used as a criterion to compare the structure of different instances. The information may also help to predict the possible adaptation of heuristics on the landscape.

Currently, we are working on two points. First, we study other search landscapes with the measures proposed in this paper. Second, we are working on measures for the *process landscape* (S, v, f, ϕ). In particular, we are studying a measure called Average Cost / Temperature function (AC/T function) [3]. This measure is based on statistical features obtained with Metropolis sampling at different temperatures. It allows one to associate a curve with each process landscape, and consequently to compare process landscapes.

References

1. Asselmeyer, E., Rosé H., Korst, J.: Smoothing representation of fitness landscape - The genotype phenotype map of evolution. Biol. Systems (1995)

2. Barnett, L.: Evolutionary dynamics on fitness landscapes with neutrality. MSc Dissertation. http://www.lionelb.cogs.susx.ac.uk (1997)
3. Belaidouni, M., Hao, J.K.: A measure of combinatorial landscape difficulty for the Metropolis algorithm. Submitted (1999)
4. Belaidouni, M., Hao, J.K.: Search space analysis of the maximal constraint satisfaction problem (in French). Proc. Of 5th French Workshop on Practical Solving of NP Complete Problems. 58-71,(1999)
5. Catoni, O.: Rough large estimates for simulated annealing: application to exponential schedules. The Annals of Probability. 20(3): 196-208, (1997)
6. Freuder, E.C., Wallace, R.J.: Partial Constraint Satisfaction. Artificial Intelligence. 58(1-3): 21-70, (1992)
7. Galinier, P., Hao, J.K.: Tabu search for maximal constraint satisfaction problems. LNCS 1330: 196-208, (1997)
8. Gent, I.P., MacIntyre, E., Prosser, P., Walsh, T.: Scaling effects in the CSP phase transition. Proc. of CP95, 70-87, (1995)
9. Hao, J.K., Pannier, J.: Simulated annealing and Tabu search for constraint solving. Artificial Intelligence and Mathematics IV. (1998)
10. Hertz, A., Jaumard, B., De Aragao, M.P.: Local optima topology for the k-coloring problem. Discrete Applied Mathematics 49: 257-280, (1994)
11. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Proc. of the 6th International Conference on Genetic Algorithms. 184-192, (1995)
12. Kauffman, S. A.: 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)
13. Kirkpatrick, S., Toulouse, G.: Configuration space analysis of travelling salesman problem. J. Physics. 46: 1277-1292, (1985)
14. Manderick, B., de Weger, B., Spiessens, P.: The genetic algorithm and the structure of the fitness landscape, PPSN III. (1994)
15. Smith, B.M.: Phase transition and the mushy region in constraint satisfaction problems. Proc of ECAI 94. 100-104, (1994)
16. Weinberger E. : Correlated and uncorrelated fitness landscapes and how to tell the difference, Biol. Cybern. vol. 63 : 325-336, (1990)
17. Wright, S.: The roles of mutation, inbreeding, and selection in evolution. Proc. of the Sixth Congress on Genetics. Vol.1: 365, (1932)

This article was processed using the LaTeX macro package with LLNCS style