

# Inherent Choice in the Search Space of Constraint Satisfaction Problem Instances

George Boukeas, Panagiotis Stamatopoulos,  
Constantinos Halatsis, and Vassilis Zissimopoulos

Department of Informatics and Telecommunications, University of Athens  
Panepistimiopolis, Athens 15784, Greece  
{boukeas,takis,halatsis,vassilis}@di.uoa.gr

**Abstract.** Constructive methods obtain solutions to constraint satisfaction problem instances by iteratively extending consistent partial assignments. In this research, we study the *solution paths* in the search space of constructive methods and examine their distribution among the assignments of the search space. By properly employing the *entropy* of this distribution, we derive measures of the average amount of *choice* available within the search space for constructing a solution. The derived quantities directly reflect both the number and the distribution of solutions, an “open question” in the phase transition literature. We show that *constrainedness*, an acknowledged predictor of computational cost, is an aggregate measure of choice deficit. This establishes a connection between an algorithm-independent property of the search space, such as the inherent choice available for constructing a solution, and the algorithm-dependent amount of resources required to actually construct a solution.

**Keywords:** search, constraint satisfaction, mathematical foundations

## 1 Introduction

A constraint satisfaction problem consists of a set of variables and a set of constraints. A variable which has been given a value is said to be *instantiated* and a set of  $i$  instantiated variables is an *assignment*  $\alpha_i$  of size  $i$ . Assignments  $\alpha_i$  which satisfy all problem constraints are called *consistent* and assignments  $\alpha_n$  in which all  $n$  problem variables are instantiated are called *complete*. Given an instance of a constraint satisfaction problem, the goal is to obtain a consistent complete assignment (a solution) or to prove that none exists. In order to accomplish this goal, *constructive* methods iteratively extend consistent partial assignments, whereas methods based on *repair* iteratively transform inconsistent complete assignments. The search spaces explored by methods belonging to these categories overlap but do not coincide. The focus here is on constructive methods but the fact remains that any particular search space offers a set of alternative ways for obtaining a solution, a set of *solution paths*.

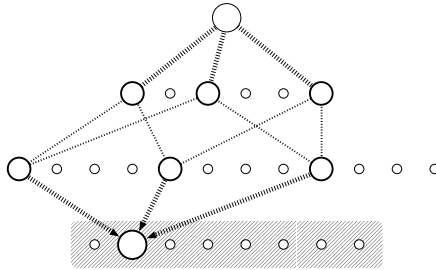
This research is prompted by the observation that both the number and distribution of solution paths play a significant role. A search space with scarce

solution paths offers very little choice for constructing a solution: a correct decision must be made at almost every choice point due to the lack of alternatives. On the other extreme, a search space with an abundance of solution paths offers ample choice for constructing a solution: decisions made at choice points are almost inconsequential due to the multitude of alternatives. The aim of this research is to quantify and investigate the amount of choice inherently available within the search space of a problem instance for constructing a solution.

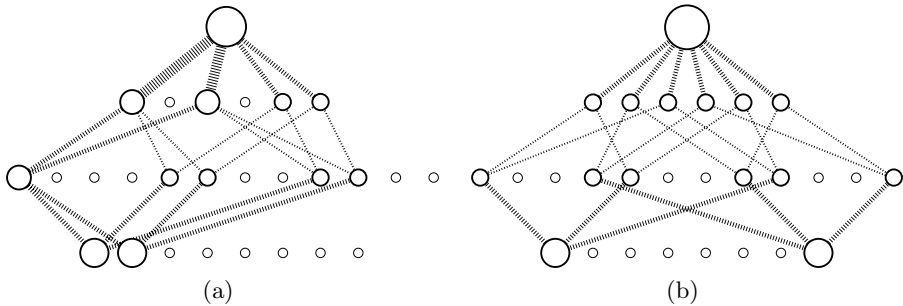
In the context of constructive methods, Sect. 2 describes the composition and size of the search space, along with the notion of paths for constructing assignments. Our view of the search space is similar to the *deep structure* of [1] and allows for an abstraction away from problem-specific and algorithm-dependent features. The structure of the search space, that is the distribution of solution paths therein, is discussed in Sect. 3. By properly employing the *entropy* of this distribution, we derive measures of the average amount of choice inherently available at each level of the search space for constructing a solution, as explained in Sects. 3 and 4. Aggregate quantities over the entire search space are discussed in Sect. 5. Recall that entropy is a measure of information and uncertainty but it is also a measure of choice [2]. Relevant information-theoretic approaches have been employed in order to characterize system structure in [3]. The introduced measures directly reflect not only the number but also the distribution of solutions to a problem instance. In fact, Sect. 6 explains how the average amount of choice available on each level of the search space is identical only among instances with isomorphic solution sets. In Sect. 7, the nature of constrainedness [4–6] as an aggregate measure of choice deficit is elucidated, which serves to explain the successful application of constrainedness as a generic predictor of computational cost and yields a number of contributions. Throughout this paper, all proofs have necessarily been omitted due to space restrictions.

## 2 The Search Space of Constructive Methods

The search space of a problem instance consists of the complete set of states that may be explored in the attempt to obtain a solution to the problem instance. In the case of constructive methods, obtaining a solution is apparently synonymous to *constructing* a solution and the set of states constituting the search space is the set of assignments  $\alpha_i$ , with size  $i$  ranging from 0 (empty assignment) to  $n$  (complete assignments). This is because the process of constructing a solution is initiated at assignment  $\alpha_0$ , where no variables are instantiated, encounters a particular assignment  $\alpha_i$  after the performance of  $i$  successive variable instantiations and eventually leads to a complete assignment  $\alpha_n$  belonging to the solution set  $\mathcal{S}$ . The notion of search space *paths* is pertinent to this constructive process: a path is an ordered set of instantiations. There are  $n!$  paths towards any complete assignment  $\alpha_n$ , corresponding to the  $n!$  distinct ways of ordering the instantiations of problem variables. Each path is essentially a different way of constructing a complete assignment and therefore all paths are disjoint, although they may overlap. Figure 1 illustrates the search space for a small problem in-



**Fig. 1.** The search space of a constructive method for instances with  $n = 3$  binary variables. Circles at level  $i$  denote assignments of size  $i$  and lines between them denote variable instantiations. The  $n!$  alternative paths available for constructing a particular complete assignment are also depicted. The complete assignments in the shaded rectangle form the search space of a repair method.



**Fig. 2.** The search space and solution paths for instances with  $n = 3$  binary variables and  $|\mathcal{S}| = 2$  solutions, with the distance between the solutions being (a) minimal: one instantiation and (b) maximal:  $n$  instantiations. Circle area and line thickness reflect the distribution of solution paths among assignments and instantiations.

stance, as well as the alternative paths available for constructing a particular complete assignment. Note how the assignments in the search space are partitioned into disjoint levels according to size: there are  $C(n, i) \cdot 2^i$  assignments of size  $i$ , corresponding to the  $C(n, i) = n!/i!(n - i)!$  different ways of selecting  $i$  variables out of  $n$  and the  $2^i$  different instantiations of these  $i$  binary variables.

### 3 Choice in the Search Space

A search space inherently offers *choice* to any constructive algorithm in the sense that it contains a set of alternative paths available for constructing a solution. In the search space of a problem instance with  $n$  variables, there are  $n!$  paths towards any complete assignment and a total of  $n!|\mathcal{S}|$  solution paths. Figure 2 contains the search spaces of two problem instances with  $n = 3$  binary variables and  $|\mathcal{S}| = 2$  solutions, as well as all solution paths contained therein. Evidently, the solution paths need not be evenly distributed throughout the search space. The fraction  $\wp(\alpha_i)$  of solution paths intersecting assignment  $\alpha_i$  is:

$$\wp(\alpha_i) = \frac{i!(n-i)!}{n!} \cdot \frac{\rho_{\mathcal{S}}(\alpha_i)}{|\mathcal{S}|} = \frac{\rho_{\mathcal{S}}(\alpha_i)}{C(n,i) \cdot |\mathcal{S}|}, \tag{1}$$

where  $\rho_{\mathcal{S}}(\alpha_i)$  is the number of complete assignments which are solutions of  $\mathcal{S}$  and are reachable from  $\alpha_i$ . This holds because there are  $i!$  ways of constructing  $\alpha_i$  and  $(n-i)!$  ways of constructing any of the  $\rho_{\mathcal{S}}(\alpha_i)$  solutions from  $\alpha_i$ . By definition,  $\rho_{\mathcal{S}}(\alpha_i)$  and  $\wp(\alpha_i)$  directly reflect the relative distances between the solutions in  $\mathcal{S}$ , with increased values indicating solution clustering under  $\alpha_i$ . The  $\wp(\alpha_i)$  fraction can be interpreted as the probability that an arbitrarily selected solution is constructed through assignment  $\alpha_i$ . Provided that a solution exists,  $\sum_{\alpha_i} \wp(\alpha_i) = 1$  holds for every level  $i$ . Essentially, the  $\wp(\alpha_i)$  fractions describe the distribution of solution paths among the assignments of level  $i$ . The entropy of this distribution, called *solution path diversity*  $\mathcal{D}_i$ , is a concise quantitative measure of the average amount of choice inherently available in the search space for constructing a solution through the assignments of level  $i$ :

$$\mathcal{D}_i = - \sum_{\alpha_i} \wp(\alpha_i) \log \wp(\alpha_i). \tag{2}$$

Assuming the log is base 2, its units are *bits*, with the intuitive interpretation that  $x$  bits of choice correspond to  $2^x$  alternatives. A small set of clustered solutions induces a search space with scarce solution paths, concentrated around a small number of assignments. In this case, there is little choice regarding how a solution may be constructed and  $\mathcal{D}_i$  is relatively low. On the other hand, when solutions and solution paths are plentiful and evenly distributed, there is ample choice regarding how a solution may be constructed and  $\mathcal{D}_i$  is relatively high. However, bear in mind that solution path diversity  $\mathcal{D}_i$  is a *relevant* quantity: its value becomes meaningful only when compared to the *general path diversity*  $\mathcal{G}_i$ , a measure of the average amount of choice available for constructing any complete assignment (not necessarily a solution), through the assignments of level  $i$ :

$$\mathcal{G}_i = \log [C(n,i) \cdot 2^i] = i + \log C(n,i). \tag{3}$$

The general path diversity reflects the size of the search space (the total number of paths) and determines the maximal attainable value for the solution path diversity. The maximum is observed in the case where all complete assignments are solutions and all paths are thus solution paths. In every other case, there exists a *choice deficit*  $\mathcal{H}_i$  in the average amount of choice available for constructing a solution through the assignments of level  $i$ :

$$\mathcal{H}_i = \mathcal{G}_i - \mathcal{D}_i. \tag{4}$$

The definition of the choice deficit  $\mathcal{H}_i$  can be interpreted as “the difference between the greatest amount of choice that can possibly be available at level  $i$  and the amount of choice actually available at level  $i$ ”. The deficit rises as the solution path diversity  $\mathcal{D}_i$  drops. It is maximal when there are no solutions and minimal when there are no infeasible assignments. It also rises along with the size of the search space, which manifests itself through the general path diversity

$\mathcal{G}_i$ . The choice deficit  $\mathcal{H}_i$  is a monotonic increasing function with respect to the search level  $i$ , due to the fact that solution paths tend to spread and become scarce at deeper levels of the search space.

## 4 Conditional Measures of Choice

A conditional version of solution path diversity  $\mathcal{D}_i$  can be derived using the distribution of solution paths among the *instantiations* leading to level  $i$ . The conditional entropy of this distribution is called *conditional solution path diversity*  $\Delta\mathcal{D}_i$  and is a more refined measure (in *bits per level* units) of the average amount of choice inherently available for constructing a solution through the assignments of level  $i$ , having reached level  $i - 1$ . It is a monotonic decreasing function with respect to  $i$ . The direct definition of  $\Delta\mathcal{D}_i$  is straightforward but it is more convenient to express it in terms of  $\mathcal{D}_i$ , since it can be proved that:

$$\Delta\mathcal{D}_i = \mathcal{D}_i - \mathcal{D}_{i-1} + \log i . \quad (5)$$

The  $\log i$  term appears because the search space implicitly contains all possible variable orderings, offering alternative paths for constructing an assignment.

The *conditional general path diversity*  $\Delta\mathcal{G}_i$  at level  $i$ , is a measure of the average amount of choice available for constructing a complete assignment through the assignments of level  $i$ , having reached level  $i - 1$ , and is a monotonic decreasing function with respect to  $i$ :

$$\Delta\mathcal{G}_i = \mathcal{G}_i - \mathcal{G}_{i-1} + \log i = 1 + \log(n - i + 1) . \quad (6)$$

This finally leads to the definition of the *conditional choice deficit*  $\Delta\mathcal{H}_i$ , a monotonic increasing function with respect to  $i$ :

$$\Delta\mathcal{H}_i = \Delta\mathcal{G}_i - \Delta\mathcal{D}_i = \mathcal{H}_i - \mathcal{H}_{i-1} . \quad (7)$$

It is interesting to read (5), (6) and (7) as: “choice (or deficit) at level  $i$  minus choice (or deficit) removed having reached level  $i - 1$ ,” which offers an intuitive explanation as to why these conditional measures are refined enough to measure choice (or deficit) at a single level.

## 5 The Sum Property

Conditional measures of choice pertain exclusively to a single level of the search space. It is therefore acceptable to sum such conditional measures over all levels, obtaining aggregate quantities. Summing (5) over all levels yields the *total solution path diversity*  $\mathbf{D}^n$ , which is the total amount of choice available for constructing a solution. Not surprisingly,  $\mathbf{D}^n$  depends upon  $n!|\mathcal{S}|$ , the number of solution paths:

$$\mathbf{D}^n(\mathcal{S}) = \sum_{i=1}^n \Delta\mathcal{D}_i = \log(n!|\mathcal{S}|) . \quad (8)$$

**Table 1.** Conditional and aggregate measures of choice and deficit for the two instances of Fig. 2. The total amounts of choice  $\mathbf{D}^n$  and deficit  $\mathbf{H}^n$ , which are identical for both instances, are expended in a different manner among the individual search space levels, due to the difference between the two instances in the distribution of solutions.

$i$	$\Delta\mathcal{G}_i$	$\Delta\mathcal{D}_i$		$\Delta\mathcal{H}_i$	
		2(a)	2(b)	2(a)	2(b)
1	2.58	1.92	2.58	0.67	0.00
2	2.00	1.33	1.00	0.67	1.00
3	1.00	0.33	0.00	0.67	1.00
sum	5.58	3.58	3.58	2.00	2.00

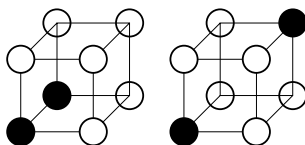
Note that  $\mathbf{D}^n$  exhibits the *sum property*: it remains invariant among problem instances with the same number of variables  $n$  and the same number of solutions  $|\mathcal{S}|$ , even though it is an aggregate quantity comprising individual, per-level amounts of choice  $\Delta\mathcal{D}_i$  which need not be identical among all such instances. In the same manner, summing (7) over all levels yields the *total choice deficit*  $\mathbf{H}^n$ , for which the sum property also holds:

$$\mathbf{H}^n(\mathcal{S}) = \sum_{i=1}^n \Delta\mathcal{H}_i = \sum_{i=1}^n \Delta\mathcal{G}_i - \sum_{i=1}^n \Delta\mathcal{D}_i = n - \log |\mathcal{S}|. \tag{9}$$

To provide a concrete example, conditional and aggregate measures of choice and deficit for the instances in Fig. 2 have been computed and included in Table 1. Interpreting the numbers in Table 1 using Fig. 2 is instructive. Notice how there is no choice deficit  $\Delta\mathcal{H}_1$  for instance 2(b) since there are no instantiations at that level not included in a solution path. Note also how, for the same instance, there is no choice  $\Delta\mathcal{D}_3$  since, having reached an assignment at the second level, there is only one instantiation in each case leading to a solution. Finally, notice how one bit of choice for  $\Delta\mathcal{D}_2$  corresponds to exactly two alternative instantiations extending each assignment at the first level. This is half of the instantiations generally available for extending each assignment at this level and this is why the choice deficit  $\Delta\mathcal{H}_2$  is equal to  $\Delta\mathcal{D}_2$  and half of  $\Delta\mathcal{G}_2$ .

## 6 The Distribution of Solutions

The search spaces in Fig. 2 both correspond to instances with  $n = 3$  variables and  $|\mathcal{S}| = 2$  solutions. The number of solution paths is identical for the two instances and so is the aggregate amount of choice  $\mathbf{D}^n$  available for constructing a solution. It is only the relative distance between the solutions in  $\mathcal{S}$  which differentiates the two instances and yet the distribution of solution paths in the search space is notably dissimilar. The values of  $\mathcal{D}_i$  for these instances are also dissimilar since  $\mathcal{D}_i$  is, by definition, sensitive to the relative distances between the solutions and the distribution of solution paths (this carries on to  $\Delta\mathcal{D}_i$ ,  $\mathcal{H}_i$  and  $\Delta\mathcal{H}_i$  as well). It turns out that the aggregate amount of choice  $\mathbf{D}^n$  and the way this



**Fig. 3.** The distribution of solutions for the two example instances of Fig. 2, with the distance between the solutions being (a) minimal: one instantiation and (b) maximal:  $n$  instantiations.

is distributed among the search space levels through the  $\Delta\mathcal{D}_i$ 's coincides only among *isomorphic* instances. In order to obtain a definition of isomorphism, map the  $2^n$  complete assignments of a  $n$ -variable problem instance to the  $2^n$  vertices of a  $n$ -hypercube. To map the solution set  $\mathcal{S}$ , let the vertices which correspond to solutions be colored black and the rest of the vertices be colored white. Such a colored hypercube reflects the *distribution of solutions*, the relative distances between them. For example, Fig. 3 contains the distribution of solutions for the two example instances of Fig. 2, with the distance between the solutions being (a) minimal: one instantiation and (b) maximal:  $n$  instantiations. Two problem instances are *isomorphic* iff the corresponding distributions of solutions (colored hypercubes) are isomorphic. It is straightforward to show that if two instances  $S$  and  $T$  are isomorphic, then for every level  $i$  of the search space it holds that  $\Delta\mathcal{D}_i^S = \Delta\mathcal{D}_i^T$ . In addition, the inverse has also been invariably observed (no formal proof is currently available): given two instances  $S$  and  $T$ , if for every level  $i$  of the search space it holds that  $\Delta\mathcal{D}_i^S = \Delta\mathcal{D}_i^T$ , then  $S$  and  $T$  are isomorphic. It is not uncommon for entropy to be an invariant of isomorphic structures [7]. The point is that solution path diversity, the measure of the average amount of choice available in a level of the search space for constructing a solution, is *necessary and sufficient for discriminating between non-isomorphic instances*. In contrast to cruder aggregate measures which only reflect the number of solutions, it is a refined indicator of their distribution. This is a significant development in itself and will also prove useful in the following section.

## 7 Constrainedness

The *constrainedness*  $\kappa$  of an ensemble of problem instances is defined as:

$$\kappa \equiv 1 - \frac{\log\langle|\mathcal{S}|\rangle}{n}, \quad (10)$$

where  $\langle|\mathcal{S}|\rangle$  is the average number of solutions over all instances of the ensemble and  $n$  is the number of variables in each instance [4]. Ensemble constrainedness has been successfully employed in a variety of problem classes as a predictor of computational cost. Its wide applicability is due to the fact that the parameters it involves, that is number of variables  $n$  and average number of solutions  $\langle|\mathcal{S}|\rangle$ , are problem-independent. However, there exists no formal justification as to

why constrainedness is defined the way it is and therefore no formal explanation for its successful application. The interpretation of constrainedness as “average information content per variable” in [8] offers no enlightenment regarding the connection between constrainedness and computational cost. Nevertheless, using (9) and the constrainedness  $\kappa_p$  of a single instance  $p$ , an interpretation in terms of the total choice deficit  $\mathbf{H}^n$  is provided.

$$\kappa_p = 1 - \frac{\log |\mathcal{S}_p|}{n} = \frac{n - \log |\mathcal{S}_p|}{n} = \frac{\mathbf{H}^n(\mathcal{S}_p)}{n}. \quad (11)$$

Evidently, instance constrainedness  $\kappa_p$  is the average choice deficit per search space level or, in other words, the deficit of choice available in the search space for constructing a solution, averaged over all levels of the search space. When it comes to ensemble constrainedness  $\kappa$ , choice deficit is also averaged over all instances in the ensemble. This result reveals that constrainedness is a reliable and widely applicable predictor of computational cost because it reflects an intrinsic property of search spaces: deficit in the amount of choice available for constructing a solution. This verifies the claim in [4] about constrainedness being “a fundamental property of problem ensembles”. Note that the amount of choice offered in the search space of an instance is independent of the manner in which a particular algorithm may make use of such choice. Choice deficit characterizes the search space of an instance and is insufficient, by itself, for determining the exact value of computational cost. However, there is logic to *comparing* the choice deficit among instances: the lower the amount of choice available for constructing a solution path, the higher the amount of resources an algorithm is expected to expend in order to obtain such a path and vice versa.

Unfortunately, both  $\mathbf{H}^n$  and  $\kappa$  suffer from the same deficiency: they are not refined enough to *distinguish between instances with non-isomorphic solution sets*. They are aggregate quantities which depend only on the number of solutions while failing to retain any information about their distribution. According to [9], it is an important “open question” to derive a “better specification of the number and location of solutions”. Having introduced the individual quantities  $\Delta\mathcal{D}_i$ ’s and  $\Delta\mathcal{H}_i$ ’s, a direct reflection of the distribution of solutions is now available. Moreover, it can be argued along the lines of [3], that a single aggregate quantity such as the sum of  $\mathcal{H}_i$ ’s can capture additional information about the distribution of solutions. In such a case, the instances offering the least amount of choice (among all instances with the same number of solutions  $|\mathcal{S}|$ ) and are expected to require increased computational effort are the ones in which the solutions are *clustered*, which is in accordance with [10].

## 8 Conclusions

The solution set  $\mathcal{S}$  of a  $n$ -variable constraint satisfaction problem instance induces a search space common to all constructive methods. This search space contains  $n!|\mathcal{S}|$  solution paths, distinct alternative instantiation orderings which lead to solutions in  $\mathcal{S}$ . The structure of the solution set  $\mathcal{S}$  directly determines the



structure of the search space, that is the distribution of solution paths among the assignments of each search space level. The entropy of this distribution yields a measure of the average amount of choice available at each search space level for constructing a solution. Aggregate measures of choice for the entire search space, as well as measures of choice deficit are also defined. Such application of entropy as a means to study the structure of problem instance search spaces is completely novel and so is the notion of choice which is inherently available within the search space for constructing solutions. All the derived quantities directly reflect not only the number but also the distribution of solutions and are, in fact, refined enough to distinguish between instances with non-isomorphic solution sets. This is a contribution to research in *phase transitions*, where structural parameters of problem instances are investigated in order to predict the expected computational cost of solving them [11] and the “better specification of the number and location of solutions” is considered to be an “open question” [9]. Another contribution of this work is the interpretation of constrainedness, an acknowledged predictor of computational cost [4–6], as an aggregate measure of choice deficit. This establishes a connection between an algorithm-independent property of the search space, such as the inherent choice available for constructing a solution, and the algorithm-dependent amount of resources required to actually construct a solution. It also underlines how the simplicity of a measure such as constrainedness incurs certain limitations and explains how these can be alleviated using the introduced measures of inherent choice.

## References

1. Williams, C., Hogg, T.: Exploiting the deep structure of constraint problems. *Artificial Intelligence* **70** (1994) 73–117
2. Shannon, C.E.: A mathematical theory of communication. *The Bell Systems Technical Journal* **27** (1948) Reprinted with corrections.
3. Crutchfield, J., Feldman, D.: Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos* **13** (2003) 25–54
4. Gent, I.P., MacIntyre, E., Prosser, P., Walsh, T.: The constrainedness of search. In: *AAAI/IAAI*. Volume 1. (1996) 246–252
5. Gent, I.P., MacIntyre, E., Prosser, P., Walsh, T.: The constrainedness of arc consistency. In: *Principles and Practice of Constraint Programming*. (1997) 327–340
6. Walsh, T.: The constrainedness knife-edge. In: *AAAI/IAAI*. (1998) 406–411
7. Ornstein, D.: Bernoulli shifts with the same entropy are isomorphic. *Adv. in Math.* **4** (1970) 337–352
8. Slaney, J.: Is there a constrainedness knife-edge? In: *Proceedings of the 14th European Conference on Artificial Intelligence*. (2000) 614–618
9. Hogg, T.: Refining the phase transitions in combinatorial search. *Artificial Intelligence* **81** (1996) 127–154
10. Parkes, A.J.: Clustering at the phase transition. In: *AAAI/IAAI*. (1997) 340–345
11. Hogg, T., Huberman, B., Williams, C.: Phase transitions and the search problem. *Artificial Intelligence* **81** (1996) 1–15