A probabilistic analysis of randomly generated binary constraint satisfaction problems

Martin Dyer* Alan Frieze† Michael Molloy‡

April 3, 2001

1 Introduction

The *Constraint Satisfaction Problem* (CSP) is a fundamental problem in Artificial Intelligence, with applications ranging from scene labeling to scheduling and knowledge representation. See for example Dechter [3], Mackworth [7] and Waltz [10]. An instance of the CSP comprises a set of n *variables*, each taking a value in some given domain, and a set of *constraint relations*, each of which determines the permitted joint values of a given subset of the variables. The problem is either to determine any set of values for the variables which respects all the constraint relations, or prove that none exists.

In recent years, there has been a strong interest in studying the relationship between the input parameters that define an instance of CSP (e.g. number of variables, domain sizes, tightness of constraints) and certain solution characteristics, such as the likelihood that the instance has a solution or the difficulty with which a solution may be discovered. An extensive account of relevant results, both experimental and theoretical, can be found in Hogg, Hubermann and Williams [5]. (More recently, see Smith [9] which contains some experimental work and theoretical discussion related to the results presented here.)

One of the most commonly used practices for conducting experiments with CSP is to generate a large set of random instances, all with the same defining parameters, and then for each instance in the set to use heuristics for deciding if a solution exists. (Note that, in the worst case, CSP is generally NP-complete). The proportion of random instances that have a solution is used as an indication of the likelihood that an instance will be soluble, and the average time taken per instance (by some standard algorithm) gives some measure of the hardness of such instances. A characteristic of many of these experiments is that the

*School of Computer Studies, University of Leeds, Leeds LS2 9JT, UK. Supported by EC Working Group RAND2.

†Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213, USA. Supported by grants from NSF and EPSRC.

‡Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3G4, Canada. Supported by NSERC and a Sloan Research Fellowship.

fraction of assignments of values that are permissible for each constraint is kept *constant* as the number of variables increases.

In this paper we consider only *binary* CSPs (BCSPs). These can be succinctly described in the following way: A graph $G = (V, E)$ is given, where $V = \{x_1, x_2, \ldots, x_n\}$ denotes the set of variables of the problem, and E the set of binary relations of the instance. We assume, without loss of generality, that each variable can take values in the same set $[m] = \{1, 2, \ldots, m\}$. For each edge $e = \{x_i, x_j\} \in E$, the relation can then be represented by an $m \times m$ 0-1 matrix M_e, where 0 indicates that the pair of values is forbidden and 1 that it is allowed. A solution to the associated BCSP is an assignment $f : V \to [m]$ of values to the variables, such that $M_e(f(x_i), f(x_j)) = 1$ for all $e = \{x_i, x_j\} \in E$. The aim of this paper is to conduct a probabilistic analysis of some aspects of the following simple random model of BCSP:

Model: The underlying graph G is G_{n,p_1} for some $0 < p_1 = p_1(n) < 1$. (This means that, with $V = \{x_1, x_2, \ldots, x_n\}$, we let each of the $\binom{n}{2}$ possible edges occur independently in E with probability p_1.) For each edge e of G there is a random $m \times m$ constraint matrix M_e where $M_e(i, j) = 0$ or 1 independently with probability $(1 - p_2)$ or p_2, repectively, for some $0 < p_2(n) < 1$.

We will discuss the efficacy of various simple but standard approaches to solving BCSP's. We first consider the likely efficacy of *backtrack free search.* Suppose the vertices of G are ordered $v_1, v_2, \ldots, v_n$. The *width* (Freuder [4]) $w = w(v_1, v_2, \ldots, v_n)$ of this order is given by

$$w = \max_{i \in [n]} \{|\{j : j < i \text{ and } v_j \text{ is adjacent to } v_i\}|\}.$$

For any graph G there is a minimum width $w^*(G)$ which is obtained as follows: Let v_n^* be a vertex of minimum degree in G and in general let v_i^* be a vertex of minimum degree in the subgraph of G induced by $V \setminus \{v_{i+1}^*, \ldots, v_n^*\}$.

For each integer $k \geq 3$ there is a constant c_k such that if $c_k < c < c_{k+1}$ then $w^*(G_{n,p_1}) = k$ **whp**[1] (see Pittel, Spencer and Wormald [8]). In fact, $c_k = \min_{\lambda > 0} \lambda / \mathbf{Pr}(\mathrm{Po}(\lambda) \geq k - 1)$, where $Po(\lambda)$ denotes a Poisson random variable with mean λ. Thus for example $c_3 \approx 3.35$ and $c_k = k + \sqrt{k \log k} + O(\log k)$ for large values of k, i.e. the optimal width becomes asymptotic to the average degree. It is apparently believed that, in practice, ordering by decreasing degree gives a reasonable approximation to the width ordering, but we observe the following.

Remark 1 *If one simply orders vertices in decreasing order of degree then* **whp** *one obtains a width which asymptotically equal to* $\sqrt{\frac{\log n}{\log \log n}}$*, assuming that* np_1 *is bounded as* $n \to \infty$*.*

Thus, asymptotically, this ordering will be arbitrarily bad compared with the minimum width ordering.

The following simple backtrack free algorithm has been discussed in the literature: Place the vertices of G into an optimal order $v_1^*, v_2^*, \ldots, v_n^*$ giving a width w^*. Starting with $i = 1$

[1]with high probability i.e. with probability tending to 1 as $n \to \infty$

iteratively assign a value to vertex i which is consistent with values already assigned to $v_1, v_2, \ldots, v_{i-1}$. We establish a sharp threshold for the likely success of this algorithm.

Theorem 1 *Suppose $k \geq 3$ is a constant integer and $c_k < c < c_{k+1}$. Let $\eta = (1 - p_2^k)^{-1}$ and $\epsilon > 0$ be a positive constant. Then*

- *If $m \geq (1+\epsilon) \log_\eta n$ then this algorithm succeeds* **whp**.
- *If $m \leq (1-\epsilon) \log_\eta n$ then this algorithm fails* **whp**.

Now consider the associated notion of strong k-consistency. A constraint satisfaction problem is *k-consistent* if for all sets of vertices $v, w_1, w_2, \ldots, w_k$ and all consistent assignments $a_1, a_2, \ldots, a_k$ of values to $w_1, w_2, \ldots, w_k$ there is at least one assignment value x for v which makes all the pairs v, w_i have consistent assignments. The problem is *strongly k-consistent* if it is i-consistent for $0 \leq i \leq k$. We establish the following sharp threshold for k-consistency. It is identical to that of Theorem 1, except that the constraints on k, p_1 are weaker and there is a constraint on p_2.

Theorem 2 *Let $d = np_1$ and let $\epsilon > 0$ be a positive constant. Suppose*

$$
\begin{aligned}
1 &\leq k = o(\log n / \log\log n) && (1)\\
|k \log d| &= o(\log n) && (2)\\
p_2 &\geq n^{-\epsilon/(2k^2)}. && (3)
\end{aligned}
$$

Let $\eta = (1 - p_2^k)^{-1}$. Then

- *If $m \geq (1+\epsilon) \log_\eta n$ then the problem is strongly k-consistent,* **whp**.
- *If $m \leq (1-\epsilon) \log_\eta n$ then the problem is not k-consistent,* **whp**.

We now consider the likely efficacy of a tree search algorithm. By computing the expected number of satisfying assignments we see that if

$$np_1(1 - p_2) \geq (2+\epsilon) \log m \tag{4}$$

then the problem is inconsistent **whp**.

One basic strategy for solving a CSP is a tree search algorithm in which one moves forward down the tree by selecting a vertex and making an assignment and which backtracks when it finds a set of vertices for which, given the current assignments, has no mutually consistent assignments left. Because of time constraints one can only check small sets of vertices for inconsistency, up to size K say. Our aim is to find values for the parameters m, n, p_1, p_2, K such that *any* such algorithm is likely to take a long time to finish. In Theorem 8 we give some rather complicated conditions. The following theorem gives some simpler but weaker conditions.

Theorem 3 *Assume (4) holds and that* $1 \leq d = np_1$. *Suppose that*

$$K \leq \max\left\{3, \frac{\log n}{2+\log d}\right\}, \; p_2^D \geq \frac{4}{5}, \; m/(\log n)^4 \to \infty, \; D = o(m/\log(m+n)).$$

Then **whp** *any tree search algorithm of the type described above must explore at least* m^D *nodes.*

For example, if

$$m = d = (\log n)^5 \text{ and } D = (\log n)^3 \text{ and } p_2 = 1 - \frac{1}{5D}$$

then the conditions of the theorem hold for any constant K and so **whp** any proof of inconsistency by this method will take super-polynomial time.

If a problem is inconsistent then one hopes that one can prove this by looking at small sets of vertices and showing that they themselves form an inconsistent subproblem. We give sufficient conditions for which this fails to happen **whp**.

Theorem 4 *Assume for convenience that* p_2 *is bounded below by a constant independent of* n *and that* $a, b > 0$ *are constants. Let* $\eta = (1-p_2^k)^{-1}$ *and suppose that*

C1 $k = b \log\log n$.

C2 $a \geq 2(1+\epsilon)(1-p_2)^{-1}(1+b\log p_2^{-1})$.

C3 $p_1 = \frac{a \log\log n}{n}$.

C4 $m = (1+\epsilon)\log_\eta n$.

Then **whp** *the problem is inconsistent and every subproblem induced by a set of at most* γn, $\gamma = b/(3a)$ *vertices is consistent.*

2 Some probabilistic inequalities

In Theorems 5, 6 below we will have a random variable $Z = Z(Y_1, Y_2, \ldots, Y_N)$ where $Y_i \in \Omega_i$ are independent so that Z is defined on $\Omega = \Omega_1 \times \cdots \Omega_N$.

Assumption 1

Suppose that $Y, Y' \in \Omega$ and there exists i such that $Y_j = Y'_j$ for $j \neq i$. Our assumption is that in such a case we have $|Z(Y) - Z(Y')| \leq a$.

Theorem 5 *(***Azuma-Hoeffding Inequality***)*

$$\mathbf{Pr}(|Z - \mathbf{E}(Z)| \geq t) \leq 2e^{-2t^2/(Na^2)}. \tag{5}$$

Note that this is trivially applicable if $Z = Y_1 + \cdots + Y_N$, where the Y_i are independent and $|Y_i| \leq a$ for $1 \leq i \leq N$.

Assumption 2

Suppose that in addition for any ξ, if $Z(Y) \geq \xi$ then there exist $c(\xi)$ indices $j_1, j_2, \ldots, j_{c(\xi)}$ such that if $Y'_{j_t} = Y_{j_t}$ for $t = 1, 2, \ldots, c(\xi)$ then $Z(Y') \geq \xi$ also.

Let $Med = Med(Z)$ denote a *median* of Z i.e. $\mathbf{Pr}(Z \geq Med) \geq \frac{1}{2}, \mathbf{Pr}(Z \leq Med) \geq \frac{1}{2}$.

Theorem 6 *(***Talagrand's Inequality***)*

$$\mathbf{Pr}(|Z - M| \geq tc(Med)^{1/2}) \leq 2e^{-t^2/(4a^2)}. \tag{6}$$

The setting for the next inequality is different. Let Ω be a set and $A_1, A_2, \ldots, A_N$ be subsets of Ω. Let X be a random subset of Ω where $x \in \Omega$ is independently placed into x with probability p. Let Z denote the (random) number of sets A_i which X contains i.e. $Z = |\{i \in [N] : \; X \supseteq A_i\}$. We give an upper bound for the probability that $Z = 0$. Let

$$\begin{aligned} \Delta &= \sum_{i \neq j:\, A_i \cap A_j \neq \emptyset} \mathbf{Pr}(A_i \cup A_j \subseteq X) \\ &= \sum_{i=1}^{N} \mathbf{Pr}(A_i \subseteq X) \sum_{j:\, A_i \cap A_j \neq \emptyset} \mathbf{Pr}(A_j \setminus A_i \subseteq X) \end{aligned} \tag{7}$$

Theorem 7 *(Janson's Inequality)*

$$\mathbf{Pr}(Z = 0) \leq \exp\left\{-\frac{\mathbf{E}(Z)^2}{\Delta}\right\}. \tag{8}$$

Proofs of these inequalities can be found for example in Janson, Łuczak and Ruciński [6].

3 Proof of Theorem 1

Consider an ordering $v_1, v_2, ..., v_n$ of the vertices which is formed by repeatedly choosing a vertex of minimum degree in the subgraph induced by the vertices not yet listed, and adding it to the end of the list. Let $V_t = \{j : \; v_j \text{ has } t \text{ neighbours among } v_1, v_2, \ldots, v_{j-1}\}$ for $1 \leq t \leq k$. Since $c_k < c < c_{k+1}$ then **whp** $V_1, V_2, ..., V_k$ partitions V, by the definition of c_k.

Lemma 1 *There exists $\alpha_k > 0$ (independent of n) such that* **whp** $|V_k| \geq \alpha_k n$.

Proof Recall that we order the vertices by repeatedly choosing a vertex of minimum degree in the subgraph induced by the vertices which are not yet listed, and we add that vertex to the list. Since $c_k < c < c_{k+1}$, the graph a.s. has a k-core, H. Our procedure will eventually reach H as the subgraph induced by the unlisted vertices, and then it will, for the first time, choose a vertex of degree k. We denote this time step by t.

At any point during the ordering procedure, we denote by W_i the set of vertices of degree i in the subgraph induced by the unlisted vertices. Consider the parameter

$$F = \frac{k(k-1)|W_k|}{\sum_{i\geq 1} i|W_i|} - 1.$$

Define λ be the largest solution to $c = \lambda/\mathbf{Pr}(Po(\lambda) \geq k-1)$ (note that λ exists by the definition of c_k). It is implicit from [8] that **whp** for each $i \geq k$, the number of vertices of degree i in H is $\gamma_i n + o(n)$ where $\gamma_i = \mathbf{Pr}(Po(\lambda) = i)$. (In particular, see (4.19) of [8] and the preceding discussion to see that at any point during their stripping procedure, the proportion of remaining vertices which have degree $i \geq k$ is distributed as a Poisson with mean equal z (a parameter of the stripping process) truncated at k; see (6.29) to see that in the terminal stages of the procedure, $z = \lambda$; and see the statement of their Theorem 2 to see that the total number of vertices in the k-core is $n\mathbf{Pr}(Po(\lambda) \geq k)$. These three facts imply our claim.)

From this, we will show that since $c > c_k$ then **whp** at time t we have $F < -\epsilon$ for some $\epsilon = \epsilon(c) > 0$:

It is obvious that $\frac{k\gamma_k}{\sum_{i\geq k} i\gamma_i} = \frac{1/(k-1)!}{\sum_{i\geq k} \lambda^{i-k}/(i-1)!}$ is strictly monotone decreasing as λ increases. Therefore, it will suffice to show that if c were equal to c_k then we would have $k(k-1)\gamma_k/\sum_{i\geq k} i\gamma_i = 1$.

First recall that at $c = c_k$, λ minimizes $\lambda/(1 - e^{-\lambda}\sum_{i=0}^{k-2}\frac{\lambda^i}{i!})$. Setting the derivative of this expression equal to 0 gives:

$$\begin{aligned} 1 - e^{-\lambda}\sum_{i=0}^{k-2}\frac{\lambda^i}{i!} &= \lambda\left(1 - e^{-\lambda}\sum_{i=0}^{k-2}\frac{\lambda^i}{i!}\right)' \\ &= \lambda\left(e^{-\lambda}\sum_{i=0}^{k-2}\frac{\lambda^i}{i!} - e^{-\lambda}\sum_{i=1}^{k-2}\frac{\lambda^{i-1}}{(i-1)!}\right) \\ &= \lambda e^{-\lambda}\left(\frac{\lambda^{k-2}}{(k-2)!}\right). \end{aligned}$$

Multiplying both sides by e^{λ} yields

$$e^{\lambda} - \sum_{i=0}^{k-2}\frac{\lambda^i}{i!} = \sum_{i\geq k-1}\frac{\lambda^i}{i!} \quad = \quad \frac{\lambda^{k-1}}{(k-2)!}.$$

Then multiplying both sides by λ and shifting indices yields

$$\begin{aligned} \sum_{i\geq k}\frac{i\lambda^i}{i!} &= k(k-1)\frac{\lambda^k}{k!}, \\ \sum_{i\geq k} i\gamma_i &= k(k-1)\gamma_k \end{aligned}$$

as required.

Furthermore, at each step of our procedure at most $k+1$ vertices are either put in the list or have their degrees reduced. Therefore, it is straightforward to verify that there is some $\delta = \delta(\epsilon) > 0$ such that for the first δn steps of our procedure we will have $F < -\frac{\epsilon}{2}$.

Expose the degree sequence of H. H is uniformly random with respect to its degree sequence, see for example [8]. Thus, we can generate H according to the configuration model (Bollobás [1], Bender and Canfield [2]). We will expose the pairs of the configuration, i.e. the edges of H, as they are exposed by our procedure. Therefore, when removing a vertex of degree i, its i neighbours are chosen at random from amongst all unlisted vertices, where the probability that a vertex u is chosen is proportional to its degree.

For each $j \geq 0$ define X_j to be the value of the sum $|W_1| + ... + |W_{k-1}|$ after the jth vertex has been removed, i.e. the number of remaining vertices of degree less than k. If $X_j > 0$ then upon adding the $(j+1)$st vertex to our list, we select at most $k-1$ neighbours, reducing each of their degrees by one. The expected proportion of neighbours from W_k is $k|W_k|/\sum_{i\geq 1} i|W_i|$. If $X_j = 0$ then $X_{j+1} \leq k$. Therefore, it is straightforward to verify that for $j \leq \delta n$, X_j is statistically dominated by Y_j defined as:

- $Y_0 = 0$;
- if $Y_j > 0$ then $Y_{j+1} = Y_j - 1 + BIN(k-1, q)$, where $(k-1)q - 1 = -\frac{\epsilon}{2}$;
- if $Y_j = 0$ then $Y_{j+1} = k$.

(In the second point, we use the fact that $F < -\frac{\epsilon}{2}$.) The sequence $Y_0, Y_1, ...$ is a random walk with negative drift and a reflective barrier at 0, and it is easy to confirm that a.s. the number of return-to-zeroes in this sequence before step δn is at least ζn where $\zeta = \zeta(\epsilon, \delta) > 0$. Therefore, a.s. the number of return-to-zeroes of X_i is at least ζn. Each return-to-zero of X_i corresponds to another vertex being added to V_k. This proves the lemma with $\alpha_k = \zeta$. □

Now consider vertex v. Let us say that v is "bad" if when the algorithm looks for an assignment for v it cannot find anything consistent with assignments made previously. Instead of stopping at this point, let the algorithm make an arbitrary assignment. Hence if $BAD = \{\text{bad vertices}\}$ the algorithm fails iff $BAD \neq \emptyset$. If $v \in V_t$ then

$$\mathbf{Pr}(v \in BAD) = (1 - p_2^t)^m$$

since we can generate the constraint matrix for $v = v_a, v_b$, $b < a$ when we first consider v. This matrix is not examined in any previous decisions. Now

$$\begin{aligned}
\mathbf{E}(|BAD| \mid G) &= \sum_{t=0}^{k} |V_t|(1 - p_2^t)^m \\
&\leq n(1 - p_2^k)^m \\
&= n\eta^{-m} \\
&\leq n^{-\epsilon}
\end{aligned}$$

if $m \geq (1+\epsilon)\log_\eta n$.

This verifies the first part of the theorem. Now assume that $m \leq (1-\epsilon)\log_\eta n$. Then

$$\begin{aligned}\mathbf{E}(|BAD| \mid |V_k| \geq \alpha_k n) &\geq \alpha_k n \eta^{-m} \\ &\geq \alpha_k n^{\epsilon}.\end{aligned}$$

Since, given G we can write $|BAD| = \delta_1 + \delta_2 + \cdots + \delta_n$ where the δ_i are *independent* 0-1 variables, we can use (5) to show concentration round the mean. In particular $|BAD| > 0$ **whp**. □

4 Proof of Theorem 2

Let Z_t count the number of choices of vertices v, neighbours $w_1, w_2, \ldots, w_t$ of v for which there exist consistent assignments $a_1, a_2, \ldots, a_t$ for $w_1, w_2, \ldots, w_t$ such that there is no consistent choice of a value for v. The problem is strongly k-consistent iff $Z = Z_0 + Z_1 + \cdots + Z_k = 0$. Now

$$\begin{aligned}\mathbf{E}(Z) &\leq \sum_{t=0}^{k} n\binom{n-1}{t} p_1^t \mathbf{Pr}(\exists \text{choices } a_1, a_2, \ldots, a_t) \\ &\leq \sum_{t=0}^{k} n\binom{n-1}{t} p_1^t m^t (1-p_2^t)^m \\ &\leq n\sum_{t=1}^{k} (md)^t (1-p_2^t)^m \\ &\leq kn(m(d+1))^k \eta^{-m}.\end{aligned}$$

If $m \geq (1+\epsilon)\log_\eta n$ then

$$\mathbf{E}(Z) \leq kn^{-\epsilon}((d+1)(1+\epsilon)\log_\eta n)^k \leq kn^{-\epsilon/2}((d+1)(1+\epsilon)\log n)^k = o(1)$$

after using (1), (2) and (3). So in this case $Z = 0$ **whp** proving the first part of the theorem.

Suppose next that $m \leq (1-\epsilon)\log_\eta n$. Let $\hat{Z}_k$ denote the number of choices of vertices v, neighbours $w_1, w_2, \ldots, w_k$ of v such that $w_1, w_2, \ldots, w_k$ form an independent set and assigments $a_1, a_2, \ldots, a_t$ for $w_1, w_2, \ldots, w_t$ such that there is no consistent choice of a value for v. Then

$$\mathbf{E}(\hat{Z}_k) \geq n\binom{n-1}{k} p_1^k (1 - O(k^2 p_1))\eta^{-m}$$

where the $(1 - O(k^2 p_1))$ term is the probability that the chosen vertices $w_1, w_2, \ldots, w_k$ form an independent set and we only consider one assignment to $w_1, w_2, \ldots, w_k$. So, after using Stirling's inequality,

$$\begin{aligned}\mathbf{E}(\hat{Z}_k) &\geq \frac{n}{2\sqrt{k}}\left(\frac{ed}{k}\right)^k \eta^{-m}\\ &\geq \frac{n^\epsilon}{2\sqrt{k}}\left(\frac{ed}{k}\right)^k\\ &\geq n^{\epsilon-o(1)}.\end{aligned}$$

We will apply Talagrand's inequality to a slight modification of this variable. Thus let $\tilde{Z}_k$ be $\hat{Z}_k$ where in the count for each v we only include $w_1, w_2, \ldots, w_k$ from the $\lambda = (d+1)(\log n)^2$ lowest indexed neighbours of v. Now, with $\Delta(G)$ denoting the maximum degree of G, we have

$$\mathbf{Pr}(\Delta(G) \geq \lambda) \leq n\binom{n}{\lambda}p_1^\lambda \leq n(nep_1/\lambda)^\lambda \leq (\log n)^{-(2-o(1))(\log n)^2} = o(n^{-(k+1)}).$$

Thus

$$\hat{Z}_k = \tilde{Z}_k \ \ \mathbf{whp}.$$

Furthermore, $\hat{Z}_k \leq n^{k+1}$ and so we see that

$$\mathbf{E}(\tilde{Z}_k) = \mathbf{E}(\hat{Z}_k) + o(1) \geq n^{\epsilon-o(1)}.$$

Now consider the probability space for our problem to be Ω^N, $N = \binom{n}{2}$ where Ω is the set of $m \times m$ 0-1 matrices i.e. one matrix M_e for each edge e of K_n. Then if $\nu = \nu(M)$ denotes the number of 1's in matrix M we let

$$\mathbf{Pr}(M_e = M) = \begin{cases} p_1 p_2^\nu (1-p_2)^{m^2-\nu} & M \neq 0 \\ (1-p_1) + p_1(1-p_2)^{m^2} & M = 0 \end{cases}$$

Now changing one matrix M_e can change $\tilde{Z}_k$ by at most $2\lambda^k = n^{o(1)}$. Furthermore if $\tilde{Z}_k \geq \xi$ then we can find at most $(k+1)\xi$ indices (edges) which force $\tilde{Z}_k \geq \xi$. Applying Talagrand's inequality we get

$$\mathbf{Pr}(|\tilde{Z}_k - Med(\tilde{Z}_k)| \geq t((k+1)Med(\tilde{Z}_k))^{1/2}) \leq \exp\{-t^2/(4\lambda^{2k})\}.$$

Putting $t = n^{\epsilon/3}$ we see that $\mathbf{E}(\tilde{Z}_k) = Med(\tilde{Z}_k) + O(n^{\epsilon/3+o(1)}(Med(\tilde{Z}_k))^{1/2})$. So we see that $Med(\tilde{Z}_k) \geq n^{\epsilon-o(1)}$ and then that $\tilde{Z}_k = \hat{Z}_k \neq 0$ **whp**. □

5 Proof of Theorem 3

We prove the following theorem which has a more complex set of conditions on the various parameters. It is simple to verify that the conditions of Theorem 3 imply these.

Theorem 8 *Assume (4) holds and that* $1 \le d = np_1$. *Suppose that*

$$K \le \max\left\{3, \frac{\log n}{2+\log d}\right\}. \tag{9}$$

$$p_1^{-1}K(de^2 2^m)^K \exp\left\{-\frac{m^2p_2^2}{K(K+1)^2}\right\} \to 0 \text{ as } n \to \infty. \tag{10}$$

$$p_1^{-1}2^m(dm)^D(1-p_2^D)^{m/2} \to 0. \tag{11}$$

Then **whp** *any tree search algorithm of the type described above must explore at least* m^D *nodes.*

First of all we verify (4). Let A denote the number of consistent assignments.

$$\mathbf{E}(A) = m^n(1-p_1(1-p_2))^{\binom{n}{2}} \le \exp\left\{-n\left(\frac{n-1}{2}p_1(1-p_2) - \log m\right)\right\}$$

and this tends to zero if (4) holds.

Consider the following conditions:

$\mathcal{C}_1$ There exists a set of vertices $S = \{v_1, v_2, \dots, v_k\}$, $k \le K$ which induce a connected subset of G and sets $B_1, B_2, \dots, B_k \subseteq [m]$, each of size $m_0 = \lceil m/2 \rceil$ such that there are no feasible assignments $a_i, i = 1, 2, \dots, k$ for $v_1, v_2, \dots, v_k$ for which $a_i \in B_i, i = 1, 2, \dots, k$.

$\mathcal{C}_2$ There exists a vertex v, neighbours $u_1, \dots, u_\ell \in V$ of v where $\ell \le D$, and assignments to $u_1, \dots, u_\ell$ such that v has fewer than m_0 choices of assignment which are consistent with those of $u_1 \dots, u_\ell$.

If neither $\mathcal{C}_1$ nor $\mathcal{C}_2$ occur and the problem is inconsistent then the algorithm must explore at least m^D nodes. At depth $\le D$ every vertex which has not been assigned a value will still have at least m_0 choices of assignment which are consistent with any given to its neighbours ($\bar{\mathcal{C}}_2$). Then each set of K vertices will have a mutually consistent set of choices ($\bar{\mathcal{C}}_1$).

We observe first that (9) implies that **whp** every set of $k \le K$ vertices of G contains at most $k+1$ edges. Then (9) implies $d \le n^{1/K}e^{-6}$. The probability that there exists a set of $k \le K$ vertices of G containing at least $k+2$ edges is at most

$$\sum_{k=4}^{K}\binom{n}{k}\binom{\binom{k}{2}}{k+2}p_1^{k+2} \le \sum_{k=4}^{K}\left(\frac{ne}{k}\right)^k\left(\frac{ke}{2}\right)^{k+2}\left(\frac{d}{n}\right)^{k+2} \le \left(\frac{eKd}{2n}\right)^2\sum_{k=4}^{K}\left(\frac{e^2d}{2}\right)^k$$
$$\le \left(\frac{K}{n^{1-1/K}}\right)^2\sum_{k=4}^{K}\left(\frac{n^{1/K}}{2e^4}\right)^k < n\left(\frac{K}{n^{1-1/K}}\right)^2 = o(1),$$

when $K \geq 4$ (for $K < 4$ the sum is empty).

Now, for some constant $c > 0$ we have

$$\mathbf{Pr}(\mathcal{C}_1) \leq c \sum_{k=2}^{K} \binom{n}{k} k^{k+1} p_1^{k-1} 2^{km} \pi_0 + o(1)$$

where for any fixed variables $v_1, ..., v_k$ and sets $B_1, ..., B_k$ each of size at least m_0, π_0 is an upper bound on the probability that there are no feasible assignments $a_i \in B_i$ for $v_i, i = 1, 2, \ldots, k$.

Explanation: $\binom{n}{k}$ counts the choices for $v_1, v_2, \ldots, v_k$. The $o(1)$ term is the probability that there is some such subset with at least $k+2$ edges. It is well known that there are at most k^{k-2} choices for a tree on the variables, at most $O(k^{k-0.5})$ choices for a connected subgraph with k edges, and at most $O(k^{k+1})$ choices for a connected subgraph with $k+1$ edges. Thus, the probability of such a component being present in the graph is $k^{k-2}p_1^{k-1} + O(k^{k-.5}p_1^k) + O(k^{k+1}p_1^{k+1}) < ck^{k+1}p_1^{k-1}$. Then we choose $B_1, B_2, \ldots, B_k \subseteq [m]$ in $(2^m)^k$ ways and multiply by the probability, at most π_0, that there is no possible assignment.

We need to bound π_0. For this we will use Janson's Inequality. We will use the notation of Section 2. Let H be the connected subgraph induced by $v_1, ..., v_k$ with at most $k+1$ edges. To apply Janson's inequality we define Ω to be the set of triples (v_iv_j, a_i, a_j) where v_iv_j is an edge of H, $a_i \in B_i, a_j \in B_j$. The sets $A_1, A_2, \ldots, A_N$ are those subsets of Ω which satisfy (a) there is exactly one triple for each edge of H and (b) for each v_i the triples corresponding to edges with endpoint v_i have the same values a_i. Thus, each A_i represents an assignment to $v_1, ..., v_k$, and so there are exactly $N = m_0^k$ sets A_i.

For each edge $e \in H$, we choose the random matrix M_e which gives the permissible assignments to the endpoints of e. X is the random subset of Ω which contains the triples for which $v_i = a_i, v_j = a_j$ is a permissible assignment. Z counts the number of A_i which X contains, i.e. the number of permissable assignments to H. Each element of Ω appears independently with probability p_2 and so

$$\mathbf{E}(Z) = m_0^k p_2^{k+\delta}$$

where $\delta = -1$ if H is a tree, $\delta = 0$ if H has one cycle, and $\delta = 1$ otherwise. Next we see that

$$\begin{aligned}\Delta &\leq m_0^k p_2^{k+\delta} \sum_{t=1}^{k+\delta} \binom{k+\delta}{t} m_0^{k-\max\{t,2\}} p_2^{k+\delta-t} \qquad (12)\\ &\leq k(k+1)^2 m_0^{2k-2} p_2^{2k+2\delta-2}.\end{aligned}$$

Explanation: We use expression (7). $m_0^k p_2^{k+\delta}$ is $\mathbf{E}(Z) = \sum_{i=1}^{N} \mathbf{Pr}(A_i)$ and then the sum in (12) bounds $\sum_{A_j \cap A_i \neq \emptyset} \mathbf{Pr}(A_j \setminus A_i \subseteq X)$ for every i. In the sum $t = |A_j \cap A_i|$ and $t = 2$ gives the largest contribution, by far, since (10) implies $m_0 p_2 / k \to \infty$.

Applying (8) we get

$$\pi_0 = \mathbf{Pr}(Z = 0) \leq \exp\left\{ -\frac{m_0^2 p_2^2}{k(k+1)^2} \right\},$$

and so

$$\mathbf{Pr}(\mathcal{C}_1) \leq c \sum_{k=1}^{K} p_1^{-1} k (de2^m)^k \exp\left\{ -\frac{m_0^2 p_2^2}{k(k+1)^2} \right\} \to 0$$

which follows directly from (10).

Now consider $\mathcal{C}_2$.

$$\begin{aligned}
\mathbf{Pr}(\mathcal{C}_2) &\leq n \sum_{\ell=1}^{D} \binom{n}{\ell} p_1^\ell m^\ell \binom{m}{m-m_0+1} (1-p_2^\ell)^{m-m_0+1} \\
&\leq n \sum_{\ell=1}^{D} (dm)^\ell 2^m (1-p_2^D)^{m/2} \\
&\to 0.
\end{aligned}$$

□

6 Proof of Theorem 4

First note that $m \leq (1+\epsilon)p_2^{-k} \log n$ and so

$$np_1(1-p_2) \geq 2(1+\epsilon)(1+b\log p_2^{-1}) \log\log n = 2(1+\epsilon)(\log\log n + k \log p_2^{-1}) \geq (2+\epsilon)\log m$$

and so by (4) the problem is inconsistent **whp**.

We deduce from C1 and C4 and Theorem 2 that **whp** the problem is strongly k-consistent. We prove next that **whp** every set S of $s \leq \gamma n$ vertices induces a subgraph with fewer than $ks/2$ edges. Thus if $|S| \leq \gamma n$ and $H = G[S]$ is the subgraph of G induced by S then H has no k-core and so the algorithm of Theorem 1 will find an assignment which is consistent for the sub-problem induced by H.

Now

$$\begin{aligned}
\mathbf{Pr}(\exists S, |S| \leq \gamma n \text{ containing } \geq \tfrac{k|S|}{2} \text{ edges}) &\leq \sum_{s=k+1}^{\gamma n} \binom{n}{s} \binom{\binom{s}{2}}{ks/2} p_1^{ks/2} \\
&\leq \sum_{s=k+1}^{\gamma n} \left(\frac{ne}{s} \left(\frac{sea}{bn} \right)^{k/2} \right)^s \\
&= o(1).
\end{aligned}$$

□

Remark 2 *We see from the proof that if $a < \frac{1}{3}b$ then* **whp** *there is no k-core and the problem itself is k-consistent. So presumably there is a transition from being solvable by a simple backtrack free algorithm to infeasible as a increases. Whether this transition is sharp is unclear.*

References

[1] B. Bollobás, *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European Journal on Combinatorics **1** (1980) 311–316.

[2] E. A. Bender and E. R. Canfield, *The asymptotic number of labelled graphs with given degree sequence*, Journal of Combinatorial Theory (A) **24** (1978) 296–307.

[3] R. Dechter, *Constraint networks*, in Encyclopedia of Artificial Intelligence, S. Shapiro (ed.), Wiley, New York, 2nd ed. (1992) 276–285.

[4] E. C. Freuder, *A sufficient condition for backtrack-free search*, Journal of the ACM **29** (1982) 24–32.

[5] D. G. Bobrow and M. Brady, eds., Special Volume on Frontiers in Problem Solving: Phase Transitions and Complexity, Guest Editors: T. Hogg, B. A. Hubermann, and C. P. Williams, *Artificial Intelligence* **81** (1996), nos. 1 and 2.

[6] S. Janson, T. Łuczak and A. Ruciński, *Random Graphs*, Wiley, 2000.

[7] A. K. Mackworth, *Constraint satisfaction*, in Encyclopedia of Artificial Intelligence, S. Shapiro (ed.), Wiley, New York, 2nd ed. (1992) 285-293.

[8] B. Pittel, J. Spencer and N. Wormald, *Sudden emergence of a giant k-core in a random graph*, Journal of Combinatorial Theory (B) **67** (1996) 111–151.

[9] B. M. Smith, *Constructing an asymptotic phase transition in random binary constraint satisfaction problems*, Research Report 2000.02, University of Leeds, January 2000. (To appear in Theoretical Computer Science: Special Issue on NP-Hardness and Phase Transitions.)

[10] D. Waltz, *Understanding line drawings of scenes with shadows*, in The Psychology of Computer Vision, McGraw-Hill, New York, (1975) 19-91.