Predicting Phase Transitions of Binary CSPs with Local Graph Topology*

Alvin C. M. Kwan, Edward P. K. Tsang, James E. Borrett
{alvin, edward, jborrett}@essex.ac.uk

Department of Computer Science
University of Essex
Wivenhoe Park
Colchester CO4 3SQ
Essex
United Kingdom

Abstract

The study of phase transitions in constraint satisfaction problems has received a lot of attention recently. Theoretical and empirical studies, largely based on randomly generated binary constraint satisfaction problems, have been done to predict where phase transitions occur. Problems near phase transition, which are typically hard to solve, are often used for studying effectiveness of algorithms. Smith conjectures that a phase transition occurs when the expected number of solutions for the problem is one. However, her model is unable to make accurate prediction for problems with sparse constraint graphs.

Dent and Mercer have recently proposed a model which takes local graph topology into consideration for generating hard problems. Their model allows them to generate problems with number of solutions closer to one, which are believed to be found typically near the phase transition region. Their results give evidence that the local graph information of a problem affects where phase transition happens. The question is how constraint graph information could be incorporated into the phase transition predictor to improve the accuracy of its prediction. We provide a possible answer to this question in this paper. A new phase transition predictor which takes constraint graph information into consideration is proposed. This allows the new predictor to include degree variation in constraint graph. Empirical results show that the new predictor is able to produce considerably better prediction than Smith's predictor for sparsely constrained problems.

* The research reported in this paper was supported by the EPSRC research grant ref. GR/J/42878.

1 INTRODUCTION

A constraint satisfaction problem (CSP)[1] is defined by a set of variables, each of them associated with a set of values that it can be assigned to, and a set of constraints over those variables. The aim is to find an assignment of values to variables which violate no constraints. CSPs are plentiful in real life. Examples include job-shop scheduling, map colouring and timetabling problems. Because of the significance, a wide range of research has been carried out on the topic.

Recently, the study of the phenomenon of phase transitions in CSPs has attracted a lot of attention from the research community. In brief, a phase transition happens, as an order parameter is varied, such that problems before the transition are typically under-constrained, and thus relatively easy to solve, and problems after the transition are typically over-constrained, and so easy to prove insoluble, or vice versa [Cheeseman *et. al.* 91]. The study of phase transitions is important for several reasons. First, the 'distance' from phase transition could hint at which particular algorithm-heuristic combination to use to tackle a CSP. For example, results in comparing a number of complete search and variable ordering heuristic combinations are reported in [Tsang *et.al.* 95]. In that study, forward checking with conflict directed backjumping using the fail-first heuristic has been found to be more efficient for tackling problems near phase transition whereas backmarking with conflict directed backjumping using the minimal width ordering heuristic is more efficient for problems which are far away from phase transition. Second, there is no point in attempting to find a solution for an over-constrained problem, again indicated by its 'distance' from the phase transition, unless some of its constraints are relaxed. Third, end-users might prefer to change the requirement specification of a CSP, *e.g.*, relax certain constraints, in order to shift the problem from phase transition region, at which search cost is maximal on average [Williams & Hogg 94], for the sake of achieving an approximate solution in a shorter time. Fourth, problems generated near phase transition region can be used as a tool for studying the behaviour of algorithms.

Both theoretical and empirical studies show that phase transitions occur when problems have a single solution [Williams & Hogg 94] [Smith 94]. Perhaps the first formula for estimating the number of solutions for (fully connected random binary) CSPs appears in [Haralick & Elliott 80]. A more general form of the formula, which appears in different sources [Smith 94] [Prosser 94], is:

$$E(Soln) = m^n (p2)^{\frac{1}{2}n(n-1)p1} \qquad (1)$$

where $E(Soln)$ is the expected number of solutions, n is the number of variables, m is the uniform domain size, $p1$ is the probability that a constraint exists between a pair of variables (*constraint density*) and $p2$ is the probability that a pair of values in a (binary) constraint is consistent (*constraint looseness*). The estimated $p2$ at which phase transitions

[1] In this paper, only binary CSPs with finite domains are considered.

are expected to occur, denoted by $\hat{p}_{2crit}$,[2] is defined by Equation (2) which simply replaces $E(Soln)$ in Equation (1) by one [Smith 94].

$$m^n(\hat{p}_{2crit})^{\frac{1}{2}n(n-1)p1} = 1 \qquad (2)$$

However, different studies show that Equation (2) is inadequate in predicting where phase transitions happen, with a reasonable accuracy, for sparsely constrained problems [Prosser 94] and [Smith 94]. Sometimes the $\hat{p}_{2crit}$, at which phase transition is expected to occur, is so inaccurate that all problems characterised by the tuple $<n, m, p1, \hat{p}_{2crit}>$ are actually insoluble. Smith conjectures that the inaccuracy is due to the large variance of the number of solutions for problems with small $p1$ [Smith 94a]. She gives further evidence that the solubility of a problem is related to the topology of its constraint graph.

Dent and Mercer recently proposed a model to generate hard CSPs [Dent & Mercer 95]. The essence of their model is to use local graph topology to define the tightness of individual constraints. Their model is able to produce problems with number of solutions closer to one, which are believed to be found typically near the phase transition region. Their results provides further evidence to Smith's conjecture that the constraint graph topology of CSPs affects problems' solubility, and consequently the location of phase transition. Inspired by their work, we suggest a new predictor which incorporates local graph topology into its prediction for phase transition. The new predictor is equivalent to Smith's predictor if the regularity of constraint graph[3] and uniformity of constraint looseness/tightness are maintained. The CSPs generated in accordance with the model proposed by Dent and Mercer are expected to lie at the predicted phase transition according to the new predictor.

To make this paper more self-contained, a brief review on Dent and Mercer's model will be given in the next section. Another reason for doing this is that both pieces of work use similar notations. Details of the new predictor and its relationship with Smith's predictor and Dent and Mercer's model will be outlined in Section 3. Experimental design and results will be given in Section 4. Conclusions will be given in the last section.

2 THE LOCAL CONSTRAINT TIGHTNESS MODEL [Dent & Mercer 95]

In [Dent & Mercer 95], a model for generating hard binary CSPs is proposed. In that model, a CSP is characterised by a 4-tuple $<n, m, p1, P>$ with n, m, $p1$ defined as before.

[2] In [Smith 94], [Prosser 94] and [Dent & Mercer 95], $p2$ is defined as the constraint tightness instead of constraint looseness. Thus the interpretation of $\hat{p}_{2crit}$ has also to be adjusted accordingly.

[3] A graph is regular if each of its nodes has the same number of edges incident to it.

P is defined as a set $\{ p2_{1,2},...,p2_{1,n},p2_{2,3},...,p2_{i,j},...,p2_{n-1,n} \}$ such that $1 \le i < j \le n$ and $p2_{i,j}$ is the local constraint looseness between variables i and j. The expected number of solutions is:

$$E(Soln) = m^n \prod_{1 \le i < j \le n} p2_{i,j}$$

As a phase transition predictor, the equation becomes

$$m^n \prod_{1 \le i < j \le n} p2_{i,j} = 1 \qquad (3)$$

If there is no constraint between variables i and j, $p2_{i,j}$ is set to one.[4] Note that Equation (3) is equivalent to Equation (2) when all constraints have the same tightness.

Dent and Mercer propose to set the looseness of constraints between variables i and j according to the equation below:

$$p2_{i,j} = \frac{1}{m^{\frac{1}{a_i}} m^{\frac{1}{a_j}}} \qquad (4)$$

where a_i and a_j are the degree of nodes i and j, representing variables i and j, in the constraint graph respectively.

Since each $m^{\frac{1}{a_i}}$ occurs a_i times in

$$\prod_{1 \le i < j \le n} \frac{1}{m^{\frac{1}{a_i}} m^{\frac{1}{a_j}}}$$

as node i has a degree of a_i, the product is $(1/m)^n$. Consequently, the expected number of solutions, given the local constraint looseness as defined by Equation (4), is one. Note that this model is equivalent to Smith's for regular constraint graphs.

Empirical results reported in [Dent & Mercer 95] show that their model produces problems of a similar hardness or harder to solve than Smith's. It also produces hard problems with a wide range of local constraint tightness. More importantly, it produces problems that are closer to phase transition than with Smith's model, particularly for the sparsely constrained problems. These support their conjecture that the underlying structure of a problem needs to be included into models of hard problems.

[4] The local constraint tightness model assumes the underlying constraint graph is connected.

3 PREDICTING PHASE TRANSITION WITH LOCAL GRAPH TOPOLOGY

As local graph topology has been shown to be useful in generating hard CSPs, the same kind of information should be equally important to a predictor of phase transition. This is because phase transitions are closely related to the solubility of a problem, which is affected by the topology of the constraint graph. Considering the two graph colouring problems in Figure 1.

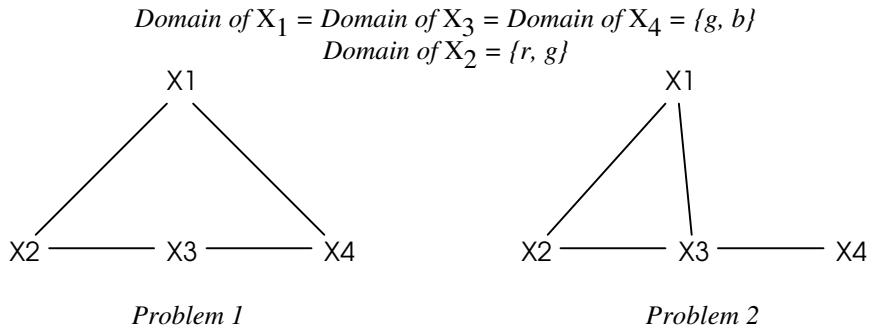

Figure 1. How graph topology affects the number of solutions of CSPs

Problem 1 and Problem 2 as depicted in Figure 1 only differ in their constraint graphs. As the tightness of the constraints between X_1, X_4 and X_1, X_3 are the same, according to Equation (1), both problems are expected to have the same number of solutions. In reality, Problem 1 has three solutions whereas Problem 2 has two solutions. It is because X_3 in Problem 2 involves in more constraints than the other nodes and finding a locally consistent value for it is harder than the others. This simple example illustrates that the topology of a constraint graph could affect the number of solutions of a CSP. Thus it is reasonable to believe that the location where a phase transition occurs would also be affected by the topology of constraint graphs.

In order to take local graph topology into consideration, we suggest using the following equation to predict where phase transitions occur. Note that the notations are the same as those described in the previous section.

$$m^n \prod_{1 \le i < j \le n} (p2_{i,j})^{\frac{2}{\left(\frac{1}{a_i}+\frac{1}{a_j}\right)\left(\frac{a_1+\ldots+a_n}{n}\right)}} = 1$$

or

$$m^n \prod_{1 \le i < j \le n} (p2_{i,j})^{\frac{2a_i a_j n}{(a_i+a_j)(a_1+\ldots+a_n)}} = 1 \qquad (5)$$

Lemma 1: Equation (5) is equivalent to Equation (3) for regular constraint graphs.

Proof: For regular constraint graphs, $a_1 = a_2 = \ldots = a_n = a$.

$$\frac{2a_i a_j n}{(a_i + a_j)(a_1 + \ldots + a_n)} = \frac{2a^2 n}{2a(an)} = 1$$

Thus, Equation (5) becomes

$$\begin{aligned} & m^n \prod_{1 \le i < j \le n} (p2_{i,j})^{\frac{2a_i a_j n}{(a_i + a_j)(a_1 + \ldots + a_n)}} \\ &= m^n \prod_{1 \le i < j \le n} p2_{i,j} \\ &= 1 \end{aligned}$$

which is Equation (3). ■

As mentioned before, if uniformity of constraint tightness is further enforced, Equation (5) is reduced to Equation (2).

Lemma 2: According to Equation (5), the problems produced by Dent and Mercer's model are expected to have one solution.

Proof: Since the looseness of a constraint between variable *i* and *j* generated by Dent and Mercer's model is governed by Equation (4), the left hand side of Equation (5) can be re-written as

$$\begin{aligned} & m^n \prod_{1 \le i < j \le n} \left(\frac{1}{m^{\frac{1}{a_i}} m^{\frac{1}{a_j}}} \right)^{\frac{2a_i a_j n}{(a_i + a_j)(a_1 + \ldots + a_n)}} \\ &= m^n \prod_{1 \le i < j \le n} \left(\frac{1}{m} \right)^{\frac{2a_i a_j n}{(a_i + a_j)(a_1 + \ldots + a_n)} * \frac{a_i + a_j}{a_i a_j}} \\ &= m^n \prod_{1 \le i < j \le n} \left(\frac{1}{m} \right)^{\frac{2n}{(a_1 + \ldots + a_n)}} \end{aligned}$$

As the number of constraints in the problems is equal to half of the sum of all degrees in the constraint graph, we then have

$$m^n \prod_{1 \le i < j \le n} \left(\frac{1}{m} \right)^{\frac{2n}{(a_1 + \ldots + a_n)}}$$

$$= m^n \left(\frac{1}{m}\right)^{\frac{2n}{(a_1+\ldots+a_n)} * \frac{(a_1+\ldots+a_n)}{2}}$$

$$= m^n \left(\frac{1}{m}\right)^n$$

$$= 1$$

■

From Equation (5), given that the number of edges remains unchanged, an increase in the variance of degrees of nodes *i* and *j* in the constraint graph would require $p2_{i,j}$ to be increased in order to maintain the right hand side of the equation to one.

4 EXPERIMENTS

The aim of the experiments was to investigate the accuracy of the new phase transition predictor for a wide range of CSPs. In particular, we are interested to compare the accuracy of the new predictor with Smith's predictor. Two series of experiments were done for this purpose. An algorithm named *forward checking with backmarking and conflict-directed backjumping* (FC-BM2-CBJ) was used.[5] It was chosen because it usually performs less compatibility checks than *forward checking with conflict-directed backjumping* [Prosser 93] which was found to be quite good for tackling CSPs near phase transition. A variant of fail-first heuristic (FFP) [Haralick & Elliott 80] named Brelaz heuristic (BZ) [Turner 88], which has been reported to work particularly well in graph colouring problems, was also used. BZ differs from FFP in handling ties. In BZ, ties are broken by selecting the future variable connected to the most other future variables whereas FFP simply picks one randomly.

The CSPs generated for experimentation are characterised by a 4-tuple <*n*, *m*, *p*1, *p*2> where the meaning of the four parameters have been mentioned in Section 1. Note that within a CSP, *m* was fixed for all its variables and *p*2 was fixed for all its constraints.

The first series of experiments was designed to study the accuracy of the new predictor for a wide range of CSPs. To do this, a number of CSPs, characterised by <*n*, *m*, *p*1, *p*2> were generated. *p*1 was set to 0.2, 0.5 and 0.8 so that problems with low, median and high constraint density were tested. *p*2 was varied from 0.04 to 0.96, with a step of 0.04. 50 problems were generated at each of the following data points. Note that asterisks mean all values within the range as mentioned above were used.

<10, 5, *, *>, <20, 5, *, *>, <30, 5, *, *>, <40, 5, *, *>, <60, 5, *, *>

[5] Note that FC-BM2-CBJ is a variant of FC-BM-CBJ as described in [Prosser 94a] except the former used a 2-dimensional data structure to store the *minimum backup level* whereas the latter uses an one-dimensional array. FC-BM2-CBJ should be able to keep more accurate backmarking information than FC-BM-CBJ.

To predict where phase transition happens using Equation (5), each CSP was examined and its expected number of solutions was roughly estimated using the right hand side of Equation (5).[6] Then the geometric mean of the expected number of solutions of the 50 problems at each data point was calculated. The $p2$ at which phase transition is expected to happen was taken as the $p2$ at which the geometric mean is closest to one.[7]

For comparison purpose, we also identified the $p2$ at which FC-BM2-CBJ with BZ took most number of compatibility checks and the approximate location of the crossover point (at which half of the 50 sampled problems, *i.e.*, 25, are soluble) [Crawford & Auton 93]. It has been reported in literature that the search cost for a solution is on average highest at the crossover point. Thus the perfect case is to have a prediction which coincides the crossover point at which the search cost is maximal.

The results of the experiments are summarised in Table 1. Since the domain size of the tested problems was 5, the smallest p2 variation we can achieve is 0.04 ($1/5^2$). If we cannot identify a single $p2$ associated with the phenomenon that we are interested, *e.g.*, crossover point, an interval of $p2$ is used instead. If the result appears to be biased to one end of an interval, that end is boldfaced. Note that there are two parenthesised figures in each entry of the 'Crossover point' column. Each of the two figures specifies the number of soluble problems for the 50 instances of CSPs generated at the $p2$ right before it. For example, the first entry of the 'Crossover point' column is interpreted as follows: out of the 50 CSPs characterised by <10, 5, 0.2, 0.20>, 15 of which are soluble whereas there are 38 soluble CSPs out of the 50 CSPs characterised by <10, 5, 0.2, 0.24>. Thus we expect that the crossover point is somewhere between $p2 = 0.20$ and $p2 = 0.24$.

Three observations can be drawn from Table 1. First, Smith's predictor appears to be less accurate for CSPs with sparse constraint graph ($p1 = 0.2$), particularly for problems with small number of variables, say $n \leq 20$. For those problems, the new predictor is able to produce more accurate predictions which are closer to the $p2$ associated with maximal search cost (in terms of average or geometric mean) as well as the crossover point. Second, the new predictor performs as good as Smith's predictor for CSPs with non-sparse constraint graph in general. Third, it appears that both predictors generate similar predictions as n increases.

[6] It is important to note that the right hand side of Equation (5) may not be able to provide an accurate estimation of the number of solutions of CSPs. However, since we are interested in the trend rather than the absolute values, a rough estimation should be good enough to serve our purpose.

[7] Average can be used instead of geometric mean. However, it appears that using geometric mean produces results a little bit closer to the crossover points than using average.

CSP Features			Predicted phase transition $p2$		Empirical $p2$ associated with the greatest search cost		
n	m	$p1$	$\hat{p}2_{crit}$ (Smith's)	New (local graph info.)	Average	Geometric mean	Crossover point
10	5	0.2	0.167	0.20	0.20	0.20	0.20(15)-0.24(38)
10	5	0.5	0.481	0.48-0.52	0.52	0.48-0.52	0.48(11)-0.52(29)
10	5	0.8	0.640	0.64	0.64	0.64	0.64(12)-0.68(41)
20	5	0.2	0.429	0.44-**0.48**	0.48-0.52	0.48	0.48(17)-0.52(37)
20	5	0.5	0.713	0.72	0.72	0.72	0.72(5)-0.76(49)
20	5	0.8	0.809	**0.80**-0.84	0.80	0.80	0.76(0)-0.80(27)
30	5	0.2	0.574	0.56-**0.60**	0.64	0.60	0.60(7)-0.64(45)
30	5	0.5	0.801	**0.80**-0.84	0.80	0.80	0.80(1)-0.84(50)
30	5	0.8	0.870	0.88	0.88	0.88	0.84(0)-0.88(27)
40	5	0.2	0.662	0.68	0.68	0.68	0.68(8)-0.72(50)
40	5	0.5	0.848	**0.84**-0.88	0.84	0.84	0.84(0)-0.88(50)
40	5	0.8	0.902	0.88-0.92	0.92	0.88	0.88(0)-0.92(50)
60	5	0.2	0.761	**0.76**-0.80	0.80	0.76	0.76(0)-0.80(50)
60	5	0.5	0.897	0.88-0.92	0.88	0.88	0.88(0)-0.92(50)
60	5	0.8	0.934	0.92-0.96	0.92	0.92	0.92(0)-0.96(50)

Table 1. The accuracy of the new phase transition predictor versus Smith's predictor for CSPs with different number of variables (sample size at each data point = 50)

Next, CSPs of larger domain sizes were tested. Only problems with sparse constraint graph were generated and those problems are characterised by the following tuples.

<10, 10, 0.2, *>, <10, 20, 0.2, *>, <20, 10, 0.2, *>, <20, 20, 0.2, *>,
<30, 10, 0.2, *>, <30, 20, 0.2, *>

$p2$ was varied in the same way as in the previous experiments. By examining the results of applying the same algorithm-heuristic combination to the CSPs, a rough idea of where phase transition happens was obtained. Since $p2$ can be varied at a finer scale for those problems, more data points (9-12) were taken around the approximate phase transition region, by varying $p2$ at a step of 0.01, so that the $p2$ at which phase transition happens can be located more precisely. The results of these experiments are summarised in Table 2.

CSP Features			Predicted phase transition $p2$		$p2$ associated with the greatest search cost		
n	m	$p1$	$\hat{p}2_{crit}$ (Smith's)	New (local graph info)	Average	Geometric mean	Crossover point
10	10	0.2	0.077	0.10	0.08	0.08	0.10(21)-0.11(27)
10	20	0.2	0.036	0.05	0.04	0.04	0.04(13)-0.05(32)
20	10	0.2	0.298	0.33-0.34	0.34	0.34	0.34(20)-0.35(28)
20	20	0.2	0.207	0.23-0.24	0.22	0.22	0.23(19)-0.24(33)
30	10	0.2	0.452	0.47-0.48	0.48	0.48	0.48(11)-0.49(29)
30	20	0.2	0.356	0.38	0.38	0.38	0.37(10)-0.38(29)

Table 2. The accuracy of the new phase transition predictor versus Smith's predictor for CSPs with larger domain sizes (sample size at each data point = 50)

Table 2 shows that the new predictor's predictions are in general more accurate than Smith's predictor's in locating the crossover point. Note that the crossover points where CSPs characterised by <10, 10, 0.2, *> and <20, 20, 0.2, *> do not correspond to the maximal search costs very well. Although it appears that Smith's predictor makes predictions closer to the problems having maximal search cost than ours for CSPs with n = 10,[8] readers are reminded that the $p2$ corresponding to the point of maximal search cost might be subject to small variation, say ±0.01, for the choice of algorithm-heuristic combinations [Kwan *et. al.* 95] whereas the crossover point is algorithm-heuristic independent .

The second series of experiments was designed to study the effect of varying the constraint graph topology on the location of phase transition. We were interested in how well the new predictor performs on constraint graphs with more variation node degree.

In order to carry out such an investigation, we added an extra option to our problem generator. For the nominal case, our generator initially minimally connects the nodes in a random fashion. Having ensured that the graph is connected, the remaining arcs are added by randomly selecting pairs of nodes from a pool populated equally by each node in the problem. If the selected pair is not already in the graph, it is added to it. Using this technique, the likelihood of any given arc are being in the CSP graph is equal for all arcs.

If we are using the bias option on our generator, then we take required variation level and include a bias into the selection pool according to that level. This bias is included randomly across all the nodes and in a ramped fashion. This is demonstrated in Figure 2.

[8] When m = 20, the $p2$ increment can be tuned to an even finer step of 0.0025 but this has not been done. Had it been done, we might be able to obtain better results than those displayed in Table 2.

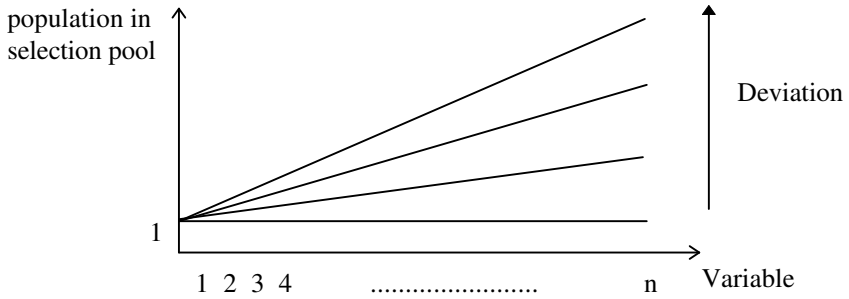

Figure 2. How the population pool is varied with requested deviation[9]

As the deviation is increased, on a scale from 0 - 10, the level of bias towards certain nodes in the pool is increased. The effect of this is to increase the likelihood of certain nodes in the constraint graph being selected for arcs in the constraint graph and hence there is a tendency for more variation of the degree of nodes in the graph

It should be noted that this scale of variation is purely an indication of the expected level of variation of degree of the nodes in the constraint graph. It is not an absolute value of, for example, the actual standard deviation of degree in the graph. However, it is a useful facility in that it does reliably induce a predictable and controllable trend in the sets of problems generated.

Using this option for varying the variance in node degree, we generated CSPs with different degree of variation in their constraint graphs. The problems tested are characterised by the following tuples.

<30, 10, 0.1, *>, <30, 10, 0.2, *>, <40, 10, 0.1, *>, <40, 10, 0.2, *>,
<60, 10, 0.1, *>, <60, 10, 0.2, *>

For each of the above tuples, degree variation was set to 3, 6 and 9. This allowed us to investigate the sensitivity of the new predictor to the local constraint graph topology of the problems. Results are summarised in Table 3.

Table 3 shows that the crossover point and the $p2$ with the maximal search cost vary with CSPs with sparse constraint graphs of different degree variations. In general, the difference between the crossover point (as well as the maximal search cost $p2$) and the predicted value according to Smith's predictor grows with the degree variation of the CSP's constraint graph. As said earlier, Smith's predictor is sometimes so inaccurate in sparsely constrained CSPs that no sampled CSPs at the predicted phase transition location are soluble. For example none of the 50 CSPs was soluble at the $\hat{p}2_{crit}$ computed

[9] Note that the ordering of variables is made random for the purposes of the bias generation.

according to Smith's predictor displayed in any row in Table 3. However, the new predictor never made a guess outside the 'mushy region' [Smith 94] in the experiments.

CSP Features				Predicted phase transition $p2$		$p2$ associated with the greatest search cost		
n	m	$p1$	deg. var.	$\hat{p}2_{crit}$ (Smith's)	New (local graph info)	Average	Geometric mean	Crossover point
30	10	0.1	3	0.201	0.24-**0.25**	0.26	0.26	0.26(16)-0.27(28)
30	10	0.1	6	0.201	0.24-**0.25**	0.30	0.26/0.27	0.27(20)-0.28(30)
30	10	0.1	9	0.201	0.24-**0.25**	0.26/0.27	0.26/0.27	0.27(24)-0.28(33)
40	10	0.1	3	0.307	0.35-0.36	0.37/0.39[10]	0.36	0.37(22)-0.38(33)
40	10	0.1	6	0.307	0.35-**0.36**	0.37/0.38	0.37	0.37(14)-0.38(30)
40	10	0.1	9	0.307	**0.36**-0.37	0.38	0.38	0.38(23)-0.39(33)
60	10	0.1	3	0.458	0.49-**0.50**	0.51	0.51	0.51(21)-0.52(38)
60	10	0.1	6	0.458	**0.50**-0.51	0.53	0.51	0.51(17)-0.52(31)
60	10	0.1	9	0.458	0.50-**0.51**	0.51	0.51	0.52(25)
30	10	0.2	3	0.452	0.48-0.49	0.50	0.49	0.49(15)-0.50(26)
30	10	0.2	6	0.452	0.48-**0.49**	0.51	0.50	0.50(19)-0.51(37)
30	10	0.2	9	0.452	0.49-0.50	0.51	0.49/0.51	0.50(19)-0.51(32)
40	10	0.2	3	0.554	**0.58**-0.59	0.59	0.59	0.59(23)-0.60(41)
40	10	0.2	6	0.554	0.58-**0.59**	0.59	0.59	0.59(10)-0.60(30)
40	10	0.2	9	0.554	0.58-**0.59**	0.60	0.59	0.59(11)-0.60(30)
60	10	0.2	3	0.677	0.69-**0.70**	0.70	0.70	0.70(12)-0.71(42)
60	10	0.2	6	0.677	**0.70**-0.71	0.70/0.71	0.70/0.71	0.70(4)-0.71(28)
60	10	0.2	9	0.677	0.70-**0.71**	0.71	0.71	0.71(20)-0.72(43)

Table 3. The accuracy of the new phase transition predictor versus Smith's predictor for CSPs with different degree variations

5 CONCLUSIONS

We have suggested a new phase transition predictor which utilises local graph topology for its prediction. The relation between the new predictor and Smith's predictor has been discussed. The new predictor expects that the CSPs generated by Dent and Mercer's model lie at phase transition. Experiments were carried out and the results showed that the new predictor is as good as Smith's predictor for CSPs with non-sparse constraint graph. For sparsely constrained CSPs, especially when the degree variation of their constraint graphs is high, the new predictor performs considerably better than Smith's predictor. Since there are real-world applications which are sparsely constrained, with some entities (*e.g.* resources) more constrained than the others, we believe that our finding may be useful for them.

[10] Double peaks exist for this set of CSPs.

Although the empirical results show that the new predictor performs reasonably well for randomly generated binary CSPs with a fixed $p2$ for all constraints, more experiments have to be done on problems with different local constraint tightness in order to confirm the accuracy of the predictor. Those experiments are at a planning stage.

Acknowledgements

Thanks are given to Michael Dent for his useful discussion.

References

[Cheeseman *et. al.* 1991]
Cheeseman *et. al.* 1991. Where the Really Hard Problems Are. In *Proceedings IJCAI-91*, 331-337.

[Crawford & Auton 93]
Crawford, J.M. & Auton, L.D., Experimental Results on the Crossover Point in Satisfiability Problems. In *Preceedings AAAI-93*, 21-27.

[Dent & Mercer 95]
Dent, M.J. & Mercer, R.E. 1995. Using Local Graph Topology to Model Hard Binary Constraint Satisfaction Problems. Technical Report UWO-CSD-439, University of Western Ontario, Canada. (To appear in *Procceedings CP'95 Workshop on Studying and Solving Really Hard Problems*.)

[Haralick & Elliott 80]
Haralick, R.M. & Elliott, G.L. 1980. Increasing Tree Search Efficiency for Constraint Satisfaction Problems. *Artificial Intelligence* 14, 263-314.

[Kwan *et. al.* 95]
Kwan, A.C.M., Tsang, E.P.K. & Borrett, J.E. 1995. Phase Transition in Finding Multiple Solutions in Constraint Satisfaction Problems. (To appear in *Procceedings CP'95 Workshop on Studying and Solving Really Hard Problems*.)

[Prosser 93]
Prosser, P. 1993. Hybrid Algorithms for the Constraint Satisfaction Problem. *Computational Intelligence* 9(3), 268-299.

[Prosser 94]
Prosser, P. 1994. Binary Constraint Satisfaction Problems: Some are Harder than Others. In *Proceedings ECAI-94*, 95-99.

[Prosser 94a]
Prosser, P. 1994. Forward Checking with Backmarking. In *Proceedings ECAI Workshop on Constraint Processing*, 59-68.

[Smith 94]
Smith, B.M. 1994. Phase Transition and the Mushy Region in Constraint Satisfaction Problems. In *Proceedings ECAI-94*, 100-104.

[Smith 94a]
Smith, B.M. 1994. Locating the Phase Transition in Binary Constraint Satisfaction Problems. Research Report 94.16, School of Computer Studies, University of Leeds, United Kingdom.

[Tsang *et. al.* 95]
Tsang, E.P.K., Borrett, J.E. & Kwan, A.C.M. 1995. An Attempt to Map the Performance of a Range of Algorithm and Heuristic Combinations. In *Proceedings AISB-95*, 203-216.

[Turner 88]
Turner, J.S. 1988. Almost All K-Colorable Graphs are Easy to Colour. *Journal of Algorithms* 9, 63-82.

[Williams & Hogg 94]
Williams, C. & Hogg, T. 1994. Exploiting the Deep Structure of Constraint Problems. *Artificial Intelligence* 70, 73-117.