On the Selection of Constraint Satisfaction Problem Formulations

James E. Borrett, Edward P. K. Tsang

Dept. of Computer Science
University of Essex
Wivenhoe Park
Colchester CO4 3SQ

email: {jborrett, edward}@essex.ac.uk

Technical Report CSM-254

October 1995

abstract: Given a problem, there are many different possible ways of formulating it as a constraint satisfaction problem. However, once these formulations have been generated, how can we choose the most appropriate one for eventual solving? We address this problem and investigate a new tool which we call "T-factor". This new tool is shown to be potentially useful in providing a method for discriminating between some formulations.

1. Introduction

A constraint satisfaction problem (CSP) can be defined by the triple <Z, D, C> where Z represents the set of variables in the problem, D represents the domains of the variables and C represents a set of constraints on the variables [1]. When formulating a problem as a CSP the task becomes one of defining an appropriate Z, D and C. It is clear that for any given problem, there is no bound on the number of different possible CSP formulations.

The range of possible ZDC formulations for a given problem leads us to the question of how to choose the most appropriate one. Although there has been some work on the manipulation of CSP formulations such as the use of interchangeability techniques [2], there has been little effort directed towards to the key question of providing a reasoned selection of the most appropriate formulation.

This paper outlines a possible method for discriminating between formulations of the same problem. We attempt to relate different ZDC formulations in terms of their relative difficulty. This difficulty is quantified in terms of a new measure known as the T-factor. The result of our work is to demonstrate that in some cases, when very different formulations of the same problem exist, it is possible to identify the formulation that is most likely to be easiest to solve.

In the next section we define the T-factors of formulations. In section 3 we present alternative formulations of the well known N-Queens and Zebra problems, together with evaluation of their T-factors. Finally in section 4 we discuss our findings and propose directions for future work.

2. T-Factor - A Measure of Formulation Difficulty

In order to compare different ZDC formulations of the same problem, we need a measure that gives us an indication of their relative difficulty. Ideally, this measure would also be algorithm independent. One such measure is the tightness of a problem. In [1], the tightness of a CSP is defined as the following:

$$T = \text{Number of Solutions / Total Number of Possible Compound Labels}$$

The measure of tightness effectively gives us the average density of solutions in the search space and it is therefore an algorithm independent measure, particularly for complete algorithms where the search space is explored in a systematic manner.

Tightness is also a good measure of the likely computational cost since it takes into account both the complexity of the search space and the constraints in the CSP[1]. It therefore dictates a large part of the complexity of the search cost curve. For example, if T is very low then the problem is likely to be very difficult, and as it tends towards 1 the CSP is likely to be easy to solve.

[1] since the constraints dictate the actual number of solutions in the search space

Notice that there is a relationship between the tightness of a CSP and the phase transition [3]. The peak in median search cost for solving CSPs is predicted to occur at the phase transition, where the expected number of solutions is 1 [4], [5], [6]. This corresponds to a tightness of 1/*S*, where *S* is the size of the search space.

It is obvious that the measure of tightness is not an exact one since we are talking about the average solution density, and within that scope there may be particularly hard or particularly easy instances. However, tightness does give us an absolute measure that is comparable across different formulations.

The main drawback with tightness is that we do not always know how many solutions exist, before a problem is actually solved. A possible answer to this is to find a method for estimating the number of solutions in the search space.

2.1 Estimating the Number of Solutions

Estimating the number of solutions in a CSP is a difficult task. However, attempts have been made in the field of Binary CSPs. In [4], [5] an equation is proposed which estimates the number of solutions for randomly generated binary CSPs which can be defined in terms of the tuple <*n*, *m*, *p1*, *p2*> where:

n	the number of variables
m	the uniform domain size of the variables
p1	the density of constraints in the primal graph
p2	the individual constraint tightness (this corresponds to the percentage of incompatible labels in the constraint matrix relations between variables)

Smith presents the following equation for *E*, the expected number of solutions;

$$E = m^n(1-p2)^{n(n-1)p1/2} \qquad \text{Eq1}$$

Despite the fact that this estimate is based on the random binary CSP model, we propose to use it as a crude approximation for binary CSPs in general, for the purposes of our tightness measure. In this case we need to ensure that we can generate the *n*, *m*, *p1* and *p2* values for each individual ZDC formulation. We propose the following modifications to the equation;

$$E = S(1-p2_{avg})^{n(n-1)p1/2} \qquad \text{Eq2}$$

where;

S	size of the search space, $\prod_{i=1}^{n} m_i$, where *m* is the domain size of variable *i*
$p2_{avg}$	average p2 over all the constraints

Our modified version of Eq1 allows us to generate an estimate for the number of solutions for any binary CSP. We acknowledge that this estimate is not totally accurate, but this does not detract from our theory, presented in the following section. The accuracy of the estimation simply affects the accuracy of the comparison of ZDC Formulations - more accurate estimations will give us more reliable comparisons.

2.2 T-Factor Defined

When the expected number of solutions is greater than 1, then:

$$\text{Approximate tightness} = \frac{E}{S} \qquad (\text{for } E \geq 1) \text{ _____ Eq3}$$

This is a measure of the density of solutions in the search space, which is also a good measure of the difficulty of the problem when $E \geq 1$. When $E < 1$, we need a different measure of difficulty of the problem. Results in the study of phase transitions show that when the problem is over-constrained it is easier to prove insolubility of the problem as E decreases [4], [5], [7], [8]. Therefore, the following can be used as a qualitative measure of the difficulty of a problem when E is less than 1:

$$\text{Approximate difficulty} = \frac{1}{E \times S} \qquad (\text{for } E < 1) \text{ _____ Eq4}$$

Combining Eq3 and Eq4, we obtain a function which is maximum at $E = 1$, which is where the hard problems are expected to occur. We call this new version of approximate tightness the tightness factor or "T-Factor"[2]. The higher the value of the T-Factor, the easier the problem should be to solve. It can be expressed as;

$$\text{T-Factor} = \begin{cases} \dfrac{E}{S} & (\text{for } E \geq 1) \\ \dfrac{1}{E \times S} & (\text{for } E < 1) \end{cases} \text{ _____ Eq5}$$

This new measure, T-Factor, allows for flexibility in its enumeration since there may be other methods of estimating the number of solutions more accurately. The actual number of solutions may even be already known, in which case T-Factors will be at their most accurate. Finally, the relative numbers of solutions might be another alternative for E. In such cases E would be expressed in terms of a base number of solutions.

3. Examples of T-Factor Analysis

As a means for demonstrating the potential usefulness of the T-Factor approach for evaluating the relative merits of CSP formulations, we present examples of T-Factor

[2] because it is a measure of difficulty of the problem which is derived from the tightness measure

analysis for two well known problems. For each example we show different ZDC formulations followed by our estimates for the number of solutions and the eventual T-factor values for each ZDC formulation.

3.1. Formulations of the N-Queens Problem

The N-Queens problem is a well known benchmark in the CSP community. The problem consists of pacing N queens on an N×N chess board such that no queen shall be able to attack any other queen. There are many ways in which N-Queens can be formulated. We now present two binary alternatives, NQ1 and NQ2.

ZDC formulation NQ1;

Z = set of N variables, where each variable represents the individual rows of the board
D = values {1...N} representing the column in which the queen for that row is located
C = binary constraints between all the variables stating that no two queens may attack each other.

ZDC formulation NQ2;

Z = set of N variables, where each variable represents the position of one of the queens
D = values $\{1...N^2\}$ representing the square in which the is located
C = binary constraints between all the variables stating that no two queens may attack each other.

Having defined the formulations, we analyse each in turn. Tables 1 and 2 show the analyses. For each formulation, we present the evaluation for N=3...10 inclusive. Note that there is a substantial amount of symmetry in formulation NQ2. This is reflected in the difference in estimation of E for the NQ1 and NQ2 and can be seen in the tables 1 and 2

N	n	m	p1	$p2_{avg}$	S	E	T-Factor
3	3	3	1	0.70	27	0.71	0.052360
4	4	4	1	0.54	256	2.36	0.009230
5	5	5	1	0.44	3125	9.48	0.003033
6	6	6	1	0.37	46656	45.61	0.000977
7	7	7	1	0.32	823543	250.25	0.000304
8	8	8	1	0.28	16777216	1633.54	0.000097
9	9	9	1	0.25	3.87E+08	12313.70	0.000032
10	10	10	1	0.23	1E+10	92913.16	0.000009

Table 1 - T-Factor results for NQ1

N	n	m	p1	$p2_{avg}$	S	E	T-Factor
3	3	9	1	0.81	729	4.63	0.006351
4	4	16	1	0.75	65536	16.00	0.000244
5	5	25	1	0.60	9765625	1153.73	0.000118
6	6	36	1	0.49	2.18E+09	100197.00	0.000046
7	7	49	1	0.43	6.78E+11	4372635.00	0.000006
8	8	64	1	0.36	2.81E+14	1.28E+09	0.000005
9	9	81	1	0.31	1.5E+17	2.61E+11	0.000002
10	10	100	1	0.26	1E+20	1.56E+14	0.000002

Table 2 - T-factor results for NQ2

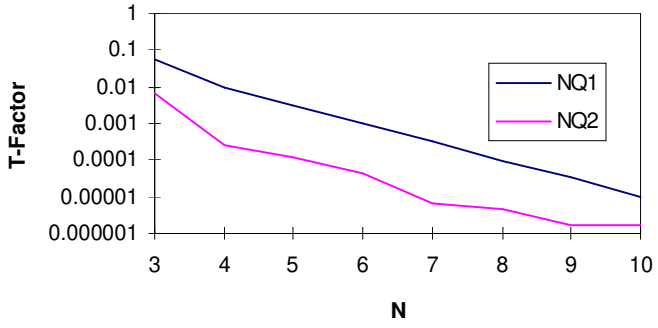

Figure 1 - T-Factor plot for formulations NQ1 and NQ2
whilst varying problem size

These results could be seen as supporting the use of NQ1, which is often chosen when the N-Queens problem is used as a benchmark. It is clear that the T-factor for NQ1 formulations is consistently higher than that of NQ2, hence the expected solution density is greater, and hence it is likely to be easier to solve.

The idea of plotting the set of T-factors for different ZDC formulations of the same problem, while varying a parameter for the problem description, is a new idea which effectively shows the response of the two formulations to that parameter. In this set of results, that response appears to leave NQ1 to be better than NQ2 for all N. We conjecture that for some problems, the T-Factor curves could cross. If this does occur it would suggest that using a general formulation for all sizes of a problem may not always be the right decision. Using T-Factor analysis we could decide where there is a change in the suitability of different formulations.

3.2. Formulations of the Zebra Problem

The zebra problem is another benchmark problem often used in the CSP community. For details of the problem see [9] where it is used for the comparison of algorithm

performance. Two alternative formulations are presented below. The first, Z1, is the most commonly encountered formulation for the problem, while Z2 is an interesting alternative.

ZDC formulation Z1;

Z = 25 variables representing the individual objects, with each to be assigned to a house
D = {1...5} for each object, representing the five possible houses to which they can be assigned.
C = Binary constraints between the groups of objects of similar type dictating that all object of the same type must belong to different houses, and further binary constraints between variables stating the specific conditions for the location of certain objects

ZDC formulation Z2;

Z = five variables representing the groups of different object types.
D = the 5-tuple resulting from the possible permutations of the five houses to which the object can be assigned. This gives two variables with domain size 4! and three with domain size 5!, once unary constraints are taken into account.
C = Binary constraints between the variables stating the specific conditions for the location of certain objects

Table 3 presents the T-Factor analysis for these two formulations.

Formulation	n	m	p1	p2	S	E	T-Factor
Z1	25	5	0.21	0.31	2.98E+17	7.77E-15	4.29E-04
Z2	5	4!, 5!, 4!, 5!, 4!	0.80	0.93	1.99E+08	0.08	4.0E-10

Table 3 - T-factors for the two zebra problem formulations.

These results suggest that Z1 would be the best formulation. However, they do show a drawback with the method. There is a major discrepancy between the expected number of solutions for this problem, as can be seen in table 3. This discrepancy is clearly wrong since it is known that for the zebra problem, there is only one solution. This is the case for both Z1 and Z2. If we were to make use of this knowledge, then this would give us T-Factors of 2.98×10^{-17} and 1.99×10^{-8} respectively, suggesting that Z2 is the best choice of formulation. The source of this error in estimating may be due to the low *p1* since *E* is known to be less accurate for lower *p1* values.

4. Discussion

The formulation of constraint satisfaction problems is an important part of the overall problem solving process which has received little attention. In this piece of work we have

introduced the idea of using T-factors for comparing ZDC formulations. The T-factor of a formulation gives us an indication of the likely solution density which is an algorithm independent measure. If the T-factors of two different formulations are compared we obtain a likely relative cost for solving them, and hence we have a criterion for deciding which formulation to choose.

Another new concept we have introduced here is that of response curves of formulations. The response of different formulations of the same problem, when plotted as T-factor against some parameter, P, can show us how the relative cost of formulations are likely to vary as the parameter is changed. It could be the case that the ranking of two formulations actually switches, suggesting that one formulation is good for low values of P and the other formulation is better for higher values of P. This idea is currently conjecture, but it is one of the areas we intend to investigate in further work.

The main drawback with our method is the accuracy of the expected number of solutions, *E*. However, there are possible ways round this problem. For example we could;

i improve the estimation method
ii take advantage of knowledge of the actual number of solutions
iii determine the relative numbers of solutions to the problem

These are all possible areas for future work.

Of course, there may be occasions when the T-Factors of formulations are very similar. In such cases, other criteria for discriminating between them would be required. However the fact that there is such scope for such varied formulations shows how important it is to be able to discriminate between them. We believe that if the notion of T-factors, or a similar measure, can be further refined, there is hope that this meaningful comparison of formulations can be achieved.

Acknowledgements

The authors would like to acknowledge some of the ideas suggested to them by Steve Brain. In particular, he was instrumental in suggesting that the most effective formulation for the zebra problem might not be the one that is most commonly used.

References

[1] Tsang, E. P. K., Foundations of Constraint Satisfaction, *Academic Press*, London 1993

[2] Freuder, E. C., Eliminating Interchangeable Values in Constraint Satisfaction Problems, *Proceedings 9th National Conference on Artificial Intelligence*, 1991, 227-233.

[3] Cheeseman, P., Kanefsky, B. & Taylor, W., Where the Really Hard Problems are, *Proceedings* 12th *International Joint Conference on Artificial Intelligence*, 1991, 331-337

[4] Smith, B. M., Phase Transition and the Mushy Region in Constraint Satisfaction Problems, *Proceedings 11th European Conference on Artificial Intelligence*, 1994, 100-104

[5] Smith, B. M., Locating the phase transition in binary constraint satisfaction problems, Research Report 94.16 School of Computer Studies, University of Leeds, 1994, modified version to appear in *Artificial Intelligence*

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

[7] Gent, I. P. & Walsh, T., Phase Transitions from Real Computational Problems, *Proceedings CP '95 Workshop - Studying and Solving Really Hard Problems,* 1995, 70-83

[8] Gent, I. P. & Walsh, T., The satisfiability constraint gap, Research Paper 702 Dept. of AI, University of Edinburgh, 1994, modified version to appear in *Artificial Intelligence*

[9] Prosser, P., Binary Constraint Satisfaction Problems: Some are Harder than Others, *Proceedings 11th European Conference on Artificial Intelligence*, 1993, 95-99