Phase Transitions and Annealed Theories: Number Partitioning as a Case Study

Ian P. Gent[1] and **Toby Walsh**[2]

Abstract. We outline a technique for studying phase transition behaviour in computational problems using number partitioning as a case study. We first build an "annealed" theory that assumes independence between parts of the number partition problem. Using this theory, we identify a parameter which represents the "constrainedness" of a problem. We determine experimentally the critical value of this parameter at which a rapid transition between soluble and insoluble problems occurs. Finite-size scaling methods developed in statistical mechanics describe the behaviour around the critical value. We identify phase transition behaviour in both the decision and optimization versions of number partitioning, in the size of the optimal partition, and in the quality of heuristic solutions. This case study demonstrates how annealed theories and finite-size scaling allows us to compare algorithms and heuristics in a precise and quantitative manner.

1 Introduction

Phase transition behaviour has recently received considerable attention in the AI community [2, 14]. Whilst random problems are typically easy to solve, hard random problems can be found at a phase transition [2]. Problems from the phase transition are now routinely used to benchmark satisfiability and constraint satisfaction algorithms. In this paper, we outline an approach for identifying such phase transitions using "annealed" theories. In addition, we show how phase transition behaviour can be used to provide precise and quantitative comparisons between algorithms and heuristics.

To illustrate our approach, we present number partitioning as a case study. Given a bag of n positive integers we partition the bag into two disjoint bags. Let Δ be the difference between the sums of the two bags. The decision problem is to determine if there is a partition such that $\Delta \leq d$. The optimization problem is to determine the minimum Δ. If $\Delta \leq 1$ then the partition is *perfect* otherwise we call it *imperfect*. Throughout this paper, we consider numbers drawn uniformly and at random from $(0, l]$. Similar results hold, however, for other distributions (*e.g.* a Poisson distribution).

Number partitioning is of both considerable theoretical and practical importance. It is one of Garey and Johnson's six basic NP-complete problems that lie at the heart of the theory of NP-completeness [4]. As it is the only problem about numbers, it is often the natural choice for NP-completeness proofs of other number problems (*e.g.* bin packing, quadratic programming, and knapsack problems). There are also many practical applications including multiprocessor scheduling, and the minimization of VLSI circuit size and delay.

Our results correct the claim of Fu that "... at least one NP complete problem, that of random number partitioning, can be solved exactly in statistical mechanics and no phase transition of any kind is found ..." [3]. By means of an annealed theory, we identify a simple phase transition for number partitioning. It remains an open question if there is any NP-complete problem which lacks a phase transition.

2 Annealed theory

We first compute the expected number of perfect partitions (see Section 6 for an extension to imperfect partitions). Consider the binary representation of the n integers that are being partitioned (the choice of base turns out to be irrelevant). We also consider just those bags with an even sum. Each bag must add to the same target sum. A nearly identical analysis can be given for bags with an odd sum. To develop an annealed theory, we average probabilities independently over the different digit positions[3] On average, we expect just 1/2 the possible partitions to add up to a number with the same parity as the least significant bit of the target sum. We can apply the same argument to each binary digit in turn, of which there are $\log_2(l)$. If we assume independence of each bit position, a partition is perfect in a fraction, $(1/2)^{\log_2(l)}$ (that is, $1/l$) of the 2^n possible partitions. The expected number of perfect partitions, $\langle Sol \rangle$ is therefore simply,

$$\langle Sol \rangle = \frac{2^n}{l}.$$

The exact analysis of number partitioning problems has been considerably more difficult. Karmarkar *et al.* [11] have determined bounds on the probability distribution for the optimum partition for a bag of real numbers drawn from the interval $[0, 1]$. Using some complicated analysis based on second moments, they showed that the size of the median optimal partition is $\Theta(\sqrt{n}/2^n)$, but were unable to derive the mean size of

[1] Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, Scotland. ipg@cs.strath.ac.uk

[2] Mechanized Reasoning Group, IRST, Loc. Panté di Povo, 38100 Trento & DIST, University of Genoa, 16143 Genoa, Italy. toby@irst.it. Both authors thank Richard Korf for the code to the CKK and greedy branch and bound algorithms, and the members of the APES research group for discussion and feedback.

[3] This is by analogy with an annealed theory of materials which averages independently over sources of disorder. Both give good approximations in the thermodynamic limit.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

the optimal partition. Interestingly, using a simple and heuristic argument based upon a drunkard's walk, Karmarkar *et al.* were able to propose quickly the result $\Theta(\sqrt{n}/2^n)$.

Similar annealed theories have been constructed in constraint satisfaction [17, 16, 7] (assuming independence between constraints) and in propositional satisfiability [12] (assuming independence between clauses).

3 Constrainedness

Phase transition behaviour depends on the constrainedness of problems. Problems which are very over-constrained are insoluble and it is usually easy to determine this. Problems which are very under-constrained are soluble and it is usually easy to guess one of the many solutions. A phase transition occurs inbetween when problems are "*critically constrained*" and it is difficult to determine if they are soluble or not.

How tightly constrained a problem is depends on both the expected number of solutions, $\langle Sol \rangle$ and the problem size, n. [6] defines a *constrainedness* parameter,

$$\kappa \quad =_{\text{def}} \quad 1 - \frac{\log_2(\langle Sol \rangle)}{n}. \qquad (1)$$

The "1 -" simply rescales κ so that it lies in the interval $[0, \infty)$. If κ is small then the problem is underconstrained and there are a large number of solutions compared to the problem size. If κ is large then the problem is overconstrained and there are very few or no solutions. Substituting the annealed value of $\langle Sol \rangle$ into Equation (1) and simplifying gives,

$$\kappa \ = \ \frac{\log_2(l)}{n}.$$

As in [17, 16] for constraint satisfaction, we predict that a phase transition for number partitioning will occur when $\langle Sol \rangle \approx 1$. Or equivalently when $\kappa \approx 1$. In the next section, we test this hypothesis experimentally.

This definition of a constrainedness parameter, κ is useful in other problem domains. In satisfiability, given a formula with n variables and l clauses each of which has k literals, κ in Equation (1) is $-\log_2(1 - 1/2^k)\, l/n$ (that is, a constant times l/n for fixed k). A phase transition in satisfiability occurs around a critical value of l/n [14]. In constraint satisfaction, given n variables, a domain size of m, constraint density of p_1 and a tightness of p_2, κ in Equation (1) becomes $\frac{n-1}{2} p_1 \log_m(\frac{1}{1-p_2})$. A phase transition in solubility again occurs around a critical value of this parameter [7].

Korf (personal communication) has predicted that a phase transition occurs when the median optimal difference is 1, and that this coincides with a peak in search cost. Using the asymptotic value for the median optimal difference due to Karmarkar *et al.* [11], Korf suggested that the phase transition occurs when $\frac{\sqrt{n}.l}{2^n} = 1$. This agrees asymptotically with $\kappa = 1$.

4 Phase transition

To determine the critical value of κ, we plot in Figure 4 the probability that a bag with an even sum has a perfect partition against κ for n from 6 to 30, and $\log_2(l)$ from 0 to $2n$. In this and all subsequent experiments 1000 problems were generated at each value of l and n. Similar results are seen using bags with an odd sum, and bags with both odd and even sums. As predicted, a phase transition occurs around $\kappa \approx 1$ with the transition sharpening as n increases.

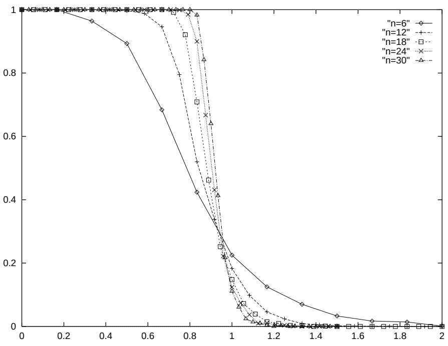

Figure 1. Probability of a perfect partition against κ.

We next applied finite-size scaling methods [1] to determine how the probability scales with problem size. Around some critical point, we predict that problems of all sizes will be indistinguishable except for a change of scale. This suggests,

$$Prob(\text{perfect partition}) = f((\frac{\kappa - \kappa_c}{\kappa_c})\,.\,n^{1/\nu}) \qquad (2)$$

where f is a fundamental function, κ_c is the critical point, and $n^{1/\nu}$ provides the change of scale. The fraction, $(\kappa - \kappa_c)/\kappa_c$ plays the rôle of the reduced temperature, $(T - T_c)/T_c$ in physical systems. Equation (2) has a fixed point where κ equals κ_c and for all n, the probability is the constant value $f(0)$. To estimate κ_c, we take the fixed point to be that where the spread in probabilities is smallest. This gives $\kappa_c = 0.96 \pm 0.02$, where the errors indicate the range giving less than 9% spread. To compute ν, we assume (2) holds at the point of 50% probability, and calculate the median estimate for ν. This gives $\nu = 1 \pm 0.3$ where errors represent the upper and lower quartiles of estimates of ν. We define a rescaled parameter,

$$\gamma =_{\text{def}} \quad \frac{\kappa - 0.96}{0.96}\,.\,n.$$

In Figure 2, we plot the probability of a perfect partition existing against γ. This graph suggest that finite-size scaling provides both a simple and accurate model for the scaling of probability with problem size. A similar rescaling of the constrainedness parameter, κ describes the finite-size scaling of the phase transition in satisfiability [12], constraint satisfaction [7] and traveling salesman problems [9].

5 Optimization cost

As in many other combinatorial problems, a peak in search cost is associated with the phase transition in solubility. Korf has proposed a branch and bound algorithm for finding the optimal partition, called CKK [13]. This uses the Karmarkar Karp (KK) heuristic based on set differencing to branch. He has shown this can give orders of magnitude better performance than a simple branch and bound algorithm with a greedy heuristic for branching. He claims that CKK outperforms the best previously-known algorithms, and will partition arbitrarily large bags of integers with up to 12 decimal digits.

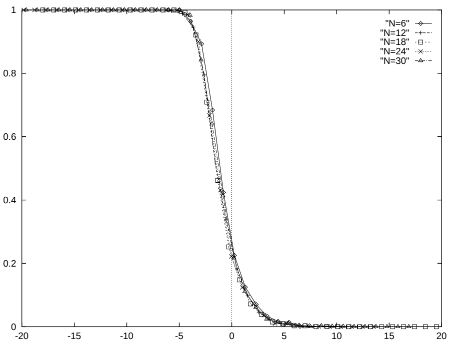

Figure 2. Probability of a perfect partition against γ.

In Figure 3, we plot the average number of nodes searched by the CKK algorithm against the rescaled parameter, γ, for n from 6 to 30 and $\log_2(l)$ from 0 to $2n$. We have observed a similar result for Korf's greedy branch and bound algorithm [13]. As in satisfiability [15] and constraint satisfaction [7], finite-size rescaling of search costs offers a clear and consistent view of how search costs varies through the phase transition for different problem sizes. In the soluble phase problems are, on average, easy. Problem hardness increases as we approach the phase boundary. Interestingly, problems appear to remain uniformly hard for CKK in the insoluble phase away from the phase boundary. Experiments out to larger l do not show problems becoming significantly easier (or harder) well away from the phase transition. This reflects the fact that a lot of search is needed to determine that a problem is insoluble. More sophisticated pruning techniques (for example, using bounds based upon modular arithmetic) can make such over-constrained problems easier and turn this easy-hard pattern into the more traditional easy-hard-easy pattern [8].

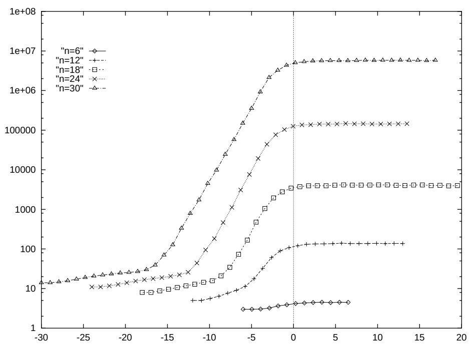

Figure 3. Average nodes searched by CKK against γ.

By plotting the maximum average search cost for all l at each value of n for both the CKK and the greedy branch and bound algorithms [13], we estimate that the worst average search costs grow as approximately $2^{0.85N}$ for the CKK algorithm and approximately $2^{0.90N}$ for the greedy branch and bound algorithm. Note that simply computing all possible partitions would give a maximum search costs that grows as 2^N. This confirms quantitatively Korf's claim that "CKK is asymptotically more efficient than the standard [greedy branch and bound] algorithm" [13].

6 Imperfect partitions

We repeat our derivation of an "annealed" theory to identify a constrainedness parameter for partitioning into less than perfect partitions (*i.e.* where the difference, d is bigger than 1). For simplicity assume d is a power of 2 and $l > d$. The argument extends with little change if we drop these assumptions. We now don't care about the bottom $\log_2(d)$ bits. We do, however, insist that the top $\log_2(l) - \log_2(d)$ bits in each bag add up to a particular parity. This gives $\langle Sol \rangle = 2^n/(l-d)$. Substituting this annealed value into Equation (1) gives,

$$\kappa = \frac{\log_2(l/d)}{n}.$$

As expected, a phase transition again occurs around the value $\kappa \approx 1$. This phase transition rescales identically to that for perfect partitioning. In Figure 4, we plot the probability that a bag has an imperfect partition against κ for $n = 24$, $\log_2(d)$ from 1 to 5 and 10, and $\log_2(l)$ from 0 to $2n$.

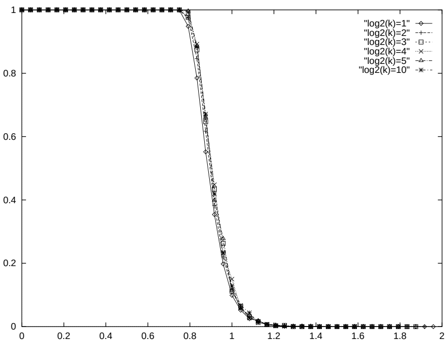

Figure 4. Probability of an imperfect partition against κ.

Search cost again peaks at the phase transition. In Figure 5, the average number of nodes searched by the CKK algorithm against κ again for $n = 24$, $\log_2(d)$ from 1 to 5 and 10, and $\log_2(l)$ from 0 to $2n$. We run the CKK algorithm as a decision procedure, terminating search immediately a partition less than or equal to d is found. Note that all problems in this graph are of the same size, n so that we do not need to rescale.

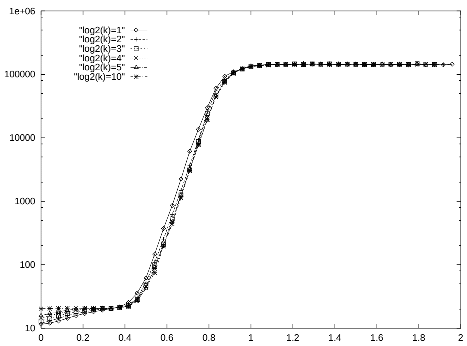

Figure 5. Average number of nodes searched by CKK finding an imperfect partition against κ.

7 Multi-way partitioning

We repeat our derivation of an "annealed" theory to identify a constrainedness parameter for multi-way partitioning. We wish to partition the bag of n numbers into m bags ($m \geq 2$) such that the maximum difference between any two bags is d. We consider just perfect partitions. The extension to imperfect partitions is analogous to that for 2-way partitioning. We assume that the bag has a sum which is an exact multiple of m and $d = 0$ (this assumption can be dropped if we let $d = 1$). We repeat our argument about digit positions to derive the expected number of perfect partitions. If we are partitioning a bag of numbers m-ways, then at each digit position, the first $m-1$ bags must each have a given sum modulo 2 and this is expected to happen with probability $(1/2)^{m-1}$. The last bag is guaranteed to have the right digit sum if the first $m-1$ do. As there are m^n possible m-way partitions and $\log_2(l)$ bit positions, the expected number of perfect partitions is,

$$\langle Sol \rangle \ = \ m^n . (\frac{1}{2})^{(m-1)\log_2(l)}.$$

Substituting this annealed value into Equation (1) gives,

$$\kappa \ = \ \frac{(m-1)\log_m(l)}{n}.$$

We again expect that a phase transition occurs around $\kappa \approx 1$ and that it will rescale identically to 2-way partitioning. In Figure 6, we plot the probability that a bag has a 3-way partition against γ for $n = 6$ to 24 and $\log_2(l)$ from 0 to $2n$.

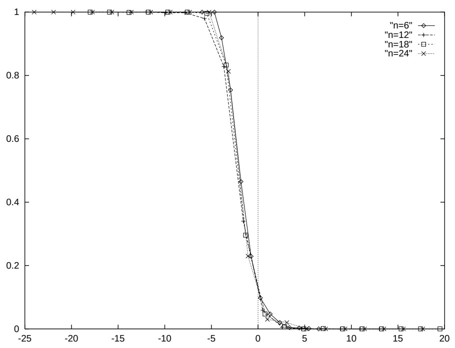

Figure 6. Probability of a perfect 3-way partition against γ.

Search cost again peaks at the phase transition. In Figure 7, we plot the average number of nodes searched by the CKK algorithm for 3-way partitioning [13] against γ. The secondary peak at $n = 18$ in the soluble region is almost entirely due to one hard problem which took 94896 nodes to solve following a poor branching decision early in search. For comparison, the median problem at this point took just 17 nodes (*i.e.* no search was needed to find a perfect 3-way partition). We conjecture that the greater incidence of such "hard" problems in 3-way partitioning compared to 2-way partitioning is a consequence of the larger branching rate and the resulting larger probability of a branching mistake. In graph-colouring and satisfiability [10, 5], similar "hard" problems occur in the soluble phase following early branching mistakes.

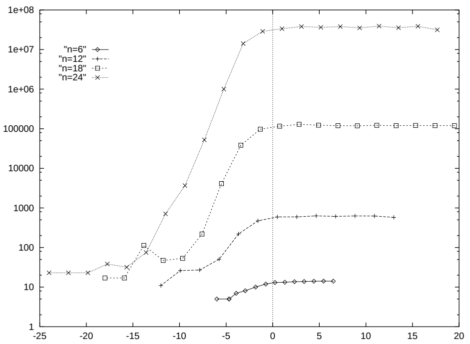

Figure 7. Average number of nodes searched by CKK finding the optimal 3-way partition against γ.

8 Optimal partition

Finite-size scaling also offers a good view of the size of the optimal partition. In Figure 8, we plot the average size of the optimal 2-way partition against γ for $n = 6$ to 24 and $\log_2(l)$ from 0 to $2n$. For $\gamma << 1$, all problems have a perfect partition. As half the problems have an even sum and half have an odd sum, $\Delta_{optimal} = 1/2$.

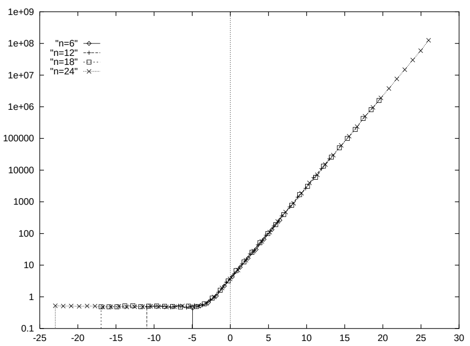

Figure 8. Average size of optimal partition against γ.

By measuring the gradient and intercept in Figure 8, and substituting Equation (2) for γ, we estimate that the average size of the optimal partition is given by,

$$\langle \Delta_{opt} \rangle \approx \max(\frac{1}{2}, 2.\frac{l}{2^n}).$$

9 Heuristic partitions

Finite-size scaling also provides a quantitative method for comparing heuristics. To use finite-size scaling to compare heuristics, we need to substitute different values of κ_c into the definition of γ. In Figure 9, we plot the average size of the KK partition (that is, the possibly sub-optimal partition found by the KK heuristic) against γ for $n = 6$ to 24 and $\log_2(l)$ from 0 to $2n$. To obtain a good fit, we set $\kappa_c = 0.40$ and not 0.96 as previously. Recall that κ_c is the fixed point for the rescaling. It gives the value of the constrainedness parameter at which we move between different phases. For $\kappa < 0.40$, the KK heuristic almost always returns the perfect partition. For $\kappa > 0.4$, the quality of the solution returned decreases as

n increases. In the region $0.40 < \kappa < 1$, the KK heuristic performs poorly as n increases. This is despite the fact that these problems have perfect partitions that can often be found with little search.

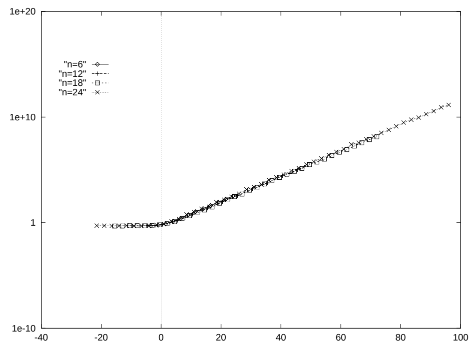

Figure 9. Average size of KK partition, $\langle\Delta_{KK}\rangle$ against γ.

Performance guarantees for optimization procedures are often given as a ratio of the optimal value. Since the optimal partition can have a difference of zero, such a ratio would be undefined for perfect partitions. We therefore make a "mean-field" approximation that, $\langle\frac{\Delta_{KK}}{\Delta_{opt}}\rangle \approx \frac{\langle\Delta_{KK}\rangle}{\langle\Delta_{opt}\rangle}$. The mean optimal partition size, $\langle\Delta_{opt}\rangle$ is at least $1/2$ so the performance ratio is always defined. In Figure 10, we plot the average performance ratio for the KK heuristic against γ again for $n = 6$ to 24 and $\log_2(l)$ from 0 to $2n$. The maximum performance ratio grows approximately as a simple exponential in n.

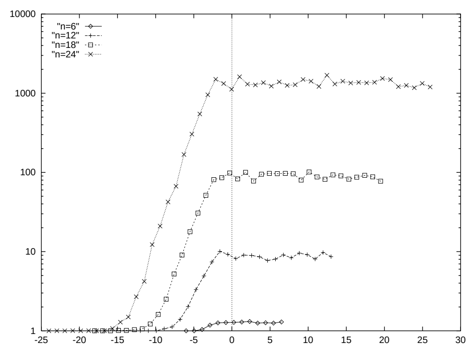

Figure 10. Average performance ratio for the KK heuristic, Δ_{KK}/Δ_{opt} against γ.

We see very similar behaviour with the greedy heuristic. This assigns the largest remaining number to the smaller partition. However, we now need to rescale around $\kappa_c = 0.15$. Finite-size scaling thus provides us with a very simple and quantitative method for comparing heuristics. For $\kappa < 0.15$ both the greedy and the KK heuristics return a perfect partition. For $0.15 < \kappa < 0.40$, the greedy heuristic but not the KK heuristic performs poorly as n increases. This is despite the fact that these problems have perfect partitions that are usually found by the KK heuristic. And for $0.40 < \kappa < 1$, both the greedy and the KK heuristics perform poorly as n increases, again despite the fact that these problems have perfect partitions that can be found often with little search.

10 Conclusions

We have outlined a technique for studying phase transition behaviour based upon "annealed" theories and finite-size scaling. Using an annealed theory, we identified a "constrainedness" parameter for number partitioning. Contrary to the claims of Fu [3], a phase transition occurs at a the critical value of this parameter. Hard number partitioning problems are associated with this transition. Finite-size scaling methods developed in statistical mechanics describe the behaviour around this critical value. We were able to identify phase transition behaviour in both the decision and optimization versions of number partitioning, in the size of the optimal partition, and in the quality of heuristic solutions. This case study demonstrates how annealed theories and finite-size scaling allows us to compare algorithms and heuristics in a precise and quantitative manner.

REFERENCES

[1] M.N. Barber. Finite-size scaling. In *Phase Transitions and Critical Phenomena, Vol. 8*, 145–266. Academic Press, 1983.

[2] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *Proc. of the 12th IJCAI*, 331–337, 1991.

[3] Y. Fu. The uses and abuses of statistical mechanics in computational complexity. In D. Stein, ed., *Lectures in the Sciences of Complexity*, 815–826. Addison-Wesley Longman, 1989.

[4] M. R. Garey and D. S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* W H Freeman, 1979.

[5] I.P. Gent and T. Walsh. Easy problems are sometimes hard. *Artificial Intelligence*, 70:335–345, 1994.

[6] I.P. Gent, E. MacIntryre, P. Prosser, and T. Walsh. The constrainedness of search. Technical report, IRST, Trento, 1996. To appear in Proceedings of AAAI-96.

[7] I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. Scaling effects in the CSP phase transition. In *Principles and Practice of Constraint Programming*, 70–87. Springer-Verlag, 1995.

[8] I.P. Gent and T. Walsh. Pruning with modular arithmetic in NP problems. Technical report, IRST, Trento, 1995.

[9] I.P. Gent and T. Walsh. The TSP phase transition. Research report 95/178, Dept. of CS, University of Strathclyde, 1995.

[10] T. Hogg and C. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence*, 69:359–377, 1994.

[11] N. Karmarkar, R. Karp, J. Lueker, and A. Odlyzko. Probabilistic analysis of optimum partitioning. *Journal of Applied Probability*, 23:626–645, 1986.

[12] S. Kirkpatrick and B. Selman. Criticial behaviour in the satisfiability of random boolean expressions. *Science*, 264:1297–1301, 1994.

[13] R.E. Korf. From approximate to optimal solutions: A case study of number partitioning. In *Proc. of the 14th IJCAI*, 266–272, 1995.

[14] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. of the 10th National Conference on Artificial Intelligence*, 1992.

[15] B. Selman and S. Kirkpatrick. Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence*, 81:273–295, 1996.

[16] B.M. Smith. Phase transition and the mushy region in constraint satisfaction problems. In *Proc. of ECAI-94*, 100–104, 1994.

[17] C.P. Williams and T. Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:73–117, 1994.