The TSP Phase Transition

Ian P. Gent
Department of Computer Science
University of Strathclyde
Glasgow G1 1XH
United Kingdom
ipg@cs.strath.ac.uk

Toby Walsh
Mechanized Reasoning Group
IRST, Trento &
DIST, University of Genoa,
Italy
toby@irst.it

In the first instance, please address correspondence to Toby Walsh at IRST, Location Panté di Povo, I38100 Trento, Italy.

Abstract

We wish to bring to the attention of the OR community the phenomenon of phase transitions in randomly generated problems. These are of considerable practical use for benchmarking algorithms. They also offer insight into problem hardness and algorithm performance. Whilst phase transition experiments are frequently performed by AI researchers, such experiments do not appear to be in common use in the OR community. To illustrate the value of such experiments, we examine a typical OR problem, the traveling salesman problem. We report in detail many features of the phase transition in this problem, and show how some of these features are also seen in real problems.

Acknowledgements

The second author is supported by a HCM Postdoctoral Fellowship. We thank Iain Buchanan for comments on a draft of this paper, and Alan Bundy, and the members of the Mathematical Reasoning Group in Edinburgh for their constructive comments and many CPU cycles donated to these and other experiments from SERC grant GR/H/23610. We also thank the MRG group at Trento and the Department of Computer Science at the University of Strathclyde for additional CPU cycles. Finally, we thank Robert Craig for providing us with his code.

1 Introduction

A perennial problem for those wishing to study algorithms is a fair means of comparison. The set of benchmark problems is often comparatively small, and it is hard to be sure that good performance is not the result of luck. Formal theorems on performance are solid, but are often hard to come by and may not be relevant to practical problems. Algorithms can, of course, be tested on random problems, but there is justifiable suspicion of such results, as random problems are not meaningful in themselves. We wish to draw the attention of the OR community to the phenomenon of "phase transitions" in randomly generated problems. This phenomenon allows us to generate random problems which are typically hard, and therefore provide a fair basis for comparison of different algorithms. They can also provide a basis for furthering our understanding of the way algorithms behave on both random and real problems. We hope therefore to contribute to the developing "empirical science of algorithms" [10].

Phase transitions have received considerable attention in the AI community [2, 13, 7]. Whilst random problems are typically easy to solve, hard random problems can be found at a phase transition [2]. AI researchers now routinely use such problems to benchmark satisfiability and constraint satisfaction algorithms. Simple scaling laws are often associated with these phase transitions. For example, scaling laws have been observed both for properties of random problems like the probability of having a solution [12, 7], and for properties of algorithms like the fitness of solutions during hill-climbing [3]. Such scaling laws are likely to prove useful in theoretical analyses of problem hardness and algorithm performance.

In this paper, we show how phase transition phenomena are of practical use in studying a typical OR problem, the traveling salesman problem (TSP). We start by showing that, contrary to earlier reports, there is a very clearly marked phase transition in TSP. We observe an "easy-hard-easy" pattern in median problem difficulty, the hard region being correlated with a change from soluble to insoluble problems. We then show empirically that a simple scaling law holds of this phase transition. We next investigate the occurrence of rare exceptionally hard problems, which can be orders of magnitude harder than those found directly at the solubility phase transition. We show that their hardness is due to early mistakes in backtracking search. Finally, we show that we can study phase transitions in real problems, and that many of the features observed in random problems occur in real problems too.

2 Computational Phase Transitions

Randomly generated problems usually have a natural order parameter. For example, for randomly generated graph colouring problems, the order parameter is the average connectivity of the graph. A rapid phase transition from colourable to uncolourable occurs at a fixed value of this order parameter. Surprisingly this value is almost completely independent of the size of the graph. Computational hardness appears to be associated with this phase transition [2]. In the colourable region, graphs have low connectivity. As almost any assignment of colours to nodes is a proper colouring, such graphs are usually easy to colour. In the uncolourable region, graphs have high connectivity. As many nodes are connected together, it is usually easy to show that there is an insufficient number of colours with which to colour the graph. By comparison, problems from the phase transition are typically hard since they are neither easily colourable nor obviously uncolourable.

Similar phase transition behaviour has been observed with many different randomly generated NP-complete problems using a variety of different complete and incomplete algorithms: for example, it has been seen in randomly generated satisfiability problems [2, 13], independent set problems [6], Hamiltonian circuits [2], and constraint satisfaction problems [14]. Phase transition phenomena have also been observed in real computational data like exam time-tabling [4]. Given this large spectrum of problem types and algorithms, computational hardness in AI is now often associated with the occurrence of a phase transition. The aim of this paper is to demonstrate the existence of such phase transition behaviour in a typical OR problem.

3 Random TSP problems

Given n cities, a tour length l and a matrix that defines the distance between each pair of cities, the traveling salesman decision problem (the TSP decision problem) is to determine if a tour of length l or less exists which visits all n cities. The TSP decision problem is one of the most famous NP-complete problems. Randomly generated TSP problems can be easily constructed by placing cities on a square of area A at random.

To solve the TSP decision problem, we first use an implementation of the branch and bound algorithm written by Robert Craig at AT&T Bell Labs. This uses the Hungarian algorithm for branching, minimally adapted by us to solve the decision rather than minization problem. Branch and bound is one of the best complete algorithms for the TSP decision problem.

For random TSP problems with n cities uniformly distributed over a rectangular area A, the expected optimal tour length [1] is,

$$l_{opt} = k.\sqrt{n.A} \tag{1}$$

where $k \approx 0.75$. This would suggest that a natural order parameter for the TSP decision problem is $l/\sqrt{n.A}$. Irrespective of the actual values for l, n and A, we expect a phase transition to occur in the probability of a tour existing for a value of the order parameter of about 0.75.

In Figure 1, we plot one view of the phase transition as we vary this order parameter. The number of randomly generated cities is fixed at 24 and the square side is fixed at 1000 units. We vary the tour length required, using the same

1000 sets of 24 random cities at each point. We plot results by means of contours of percentiles, where for example the 50% contour is the median, and the 99% contour represents behaviour that was only exceeded by 1% of problems (in this case 10 problems). We also include the best and worst case. As search cost varies over many orders of magnitude. we plot the *log* of the number of nodes searched. All logs in this paper are to base 10.

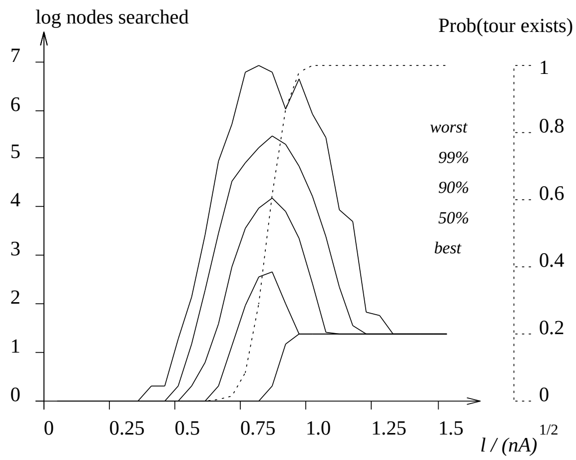

Fig 1: Varying tour length

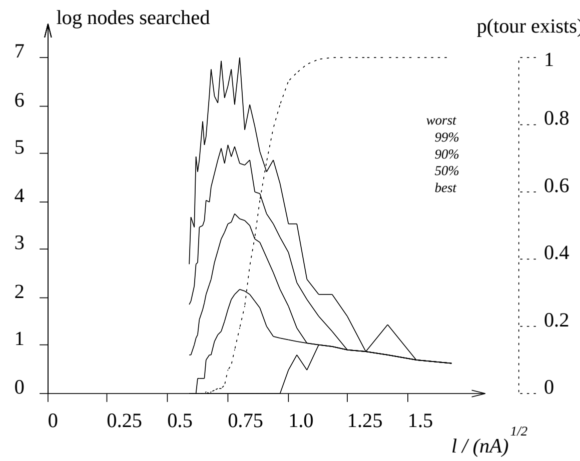

Fig 2: Varying number of cities

As with graph colouring, there is a rapid phase transition from soluble to insoluble as we vary the order parameter. The phase transition occurs, as expected from Equation 1, in the region around $l/\sqrt{n.A} \approx 0.75$. In the soluble region, problems are under-constrained and typically easy. As many tours are less than the required length, it is not difficult to find a tour that is short enough. In the insoluble region, problems are over-constrained and again typically easy. As the required tour length is a very tight bound, many tours are too long and are quickly ruled out. In the phase transition inbetween, problems are "critically constrained" and typically hard. In this region, it is difficult to determine if a tour of the required length exists without exhaustive search. This pattern of "easy-hard-easy" behaviour in the median contour is familiar from AI research into phase transitions [13]. Figure 1 clearly refutes the claim of Kirkpatrick and Selman [12] that "there are other NP-complete problems (for example, the traveling salesman problem or max-clique) that lack a clear phase boundary at which 'hard problems' cluster".

Note that whilst median problem difficulty peaks in the middle of phase transition, occasional very hard problems occur in under-constrained regions where almost all problems are soluble. The worst such problem took 4,412,760 nodes in a region where 97.7% of problems (including this one) have a tour. This is four orders of magnitude worse than the worst median problem, which took just 449 nodes in a region where 61.2% of problems have a tour. Such behaviour has previously been observed both for random graph-colouring and satisfiability problems [9, 5]. We return to such occasional hard problems in satisfiable regions in §5 and

§6.

We observe a similar phase transition when we fix the tour length and vary the number of cities visited. In Figure 2, we set $l/\sqrt{A}$ to $\frac{3}{4}.\sqrt{25}$ and vary the number of cities visited from 5 to 40 in steps of 1, testing 1000 sets of cities at each point. As expected from Equation 1, the phase transition occurs broadly around the region $l/\sqrt{n.A} \approx 0.75$. Median behaviour displayed the usual easy-hard-easy pattern through the phase transition. Compared to Figure 1, the worst case plot shows worse behaviour at small values of the order parameter, in the mostly insoluble region. Exceptionally hard problems in the mostly soluble region do not seem to occur. However, we note that as the order parameter varies, so does the number of cities. Problems in the soluble region have smaller numbers of cities and so are much easier. The worst case is bounded by a search envelope of size $O(n!)$ which is constant in Figure 1 but decreasing to the right in Figure 2. The total number of nodes searched is thus not a direct measure of *relative* problem hardness.

4 Scaling of Phase Transition

To utilise phase transition phenomena effectively, we need to determine how they scale with problem size. For instance, to test heuristic procedures like simulated annealing on random TSP problems which are larger than can be solved using complete procedures, we need a good estimate of the scaling of the probability that a random TSP problem has a tour. Analogies with phase transitions in physical systems are often useful in discovering such scaling results.

One of the most unusual and theoretically interesting phase transitions in physics occurs in spin glasses. Kirkpatrick *et al.* [11] have used an analogy with this phase transition to suggest a simple scaling result for the probability of satisfiability for a common class of randomly generated satisfiability problems. In [7], we demonstrated that this simple scaling result applied to a wide range of different classes of randomly generated satisfiability problems. Properties of algorithms like the number of constraint propagations performed on each branch of the search tree also appear to obey similar scaling results [8].

In a spin glass, each atom has a magnetic spin which can have only one of two values, "up" or "down" (1 or -1). The system therefore has a exponentially large number of possible configurations. Interactions between atoms are both ferromagnetic (promoting alignment of spins) and anti-ferromagnetic (promoting opposite spins). As a result, a spin glass is a "frustrated" system with a large number of near optimal equilibrium configurations which cannot be locally improved by flipping a spin. It is difficult therefore to get the spin glass into a state of least energy. An analogy can be made with TSP problems. A TSP problem has an exponentially large number of possible tours. It is also usually a frustrated system, having a large number of near optimal tours which cannot be locally improved (*eg.* using local changes like 2-opt or 3-opt). It is very difficult therefore to find an optimal tour. This analogy with spin glasses suggests that a macroscopic property like the probability of a tour existing obeys the following simple scaling law,

$$Prob(tour\ exists) = f((\frac{l}{\sqrt{n.A}} - \alpha) \, . \, n^{1/v}) \tag{2}$$

where f is some fundamental function, and α and v are constants.

To test this hypothesis, we generated random TSP problems with $n = 9$, 16, 25, and 36, and measured the probability that a tour exists as we traverse the phase transition. In Figure 3, each probability curve is plotted against l. The solid line represents $n = 36$. As n increases, the phase transition occurs at larger l but over a smaller range. By rescaling the x-axis for each problem size, we can quickly test (2). In line with (1), we set $\alpha = 0.75$. We varied v, and observed the best fit to (2) at $v = 2$. Figure 4 shows the data from Figure 3 rescaled so that the probability of a tour existing of length l for a random n-city tour is plotted at the x-ordinate $(\frac{l}{\sqrt{n.A}} - 0.75) \, . \, n^{1/2}$. Equation (2) is obeyed if all the probability curves coincide. Figure 4 shows that the data fits (2) very well indeed. Note that (1) is only an asymptotic limit. Using Figure 4, we can for the first time predict the probability of a tour of a given length existing for *all* values of l, n, and A.

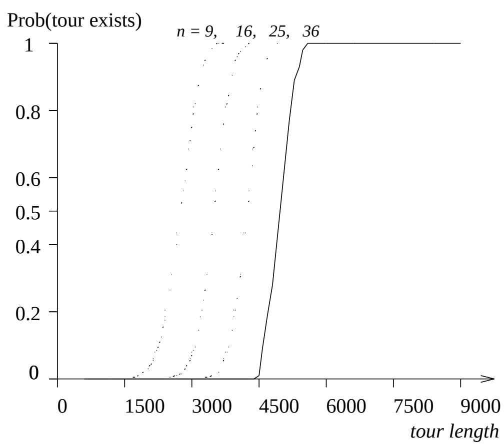

Fig 3: Prob(tour) as n varies

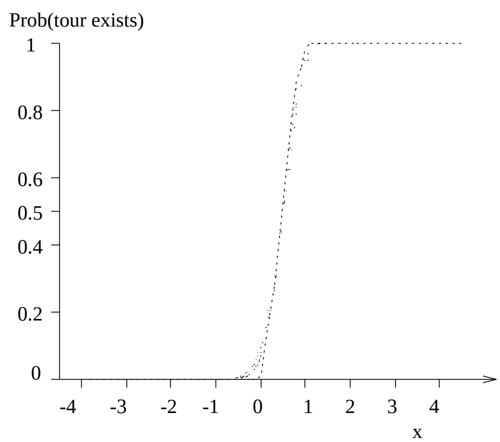

Fig 4: Rescaled data (see text)

Our results very strongly suggest that the probability of a tour existing obeys (2) with the parameters $\alpha \approx 0.75$, $v \approx 2$. So far, we have computed values for f by experimentation. Finding a closed form for f remains an interesting open problem.

5 Optimal tours

The phase transition in Figure 1 results from a mixture of behaviours on soluble and insoluble problems. To isolate the different behaviour, we found the optimal tour for the first 100 sets of cities tested in Figure 1. We could then set the tour length required to be a known distance, d from the optimal tour for each problem.

At $d = 0$ only optimal tours will be found whilst at $d = -1$ no tours will be found at all. By design, the phase transition thus occurs abruptly between $d = -1$ and 0 (indicated by the dashed line). This gives a clear picture of how problem difficulty is related to the distance from the individual transition in solubility. In Figure 5, we vary d, the distance from the optimal tour from -1000 (insoluble) to $+1000$ (soluble) in steps of 50, and for finer detail from -50 to $+50$ in steps of 5. In addition, we plot the cost at -1. For each distance d, we tested 100 sets of capitals, with $l = l_{opt} + d$.

Figure 5 demonstrates two different types of behaviour. In the insoluble region, there is exponential growth as the phase boundary is approached. In the soluble region, problems typically become harder as we approach the phase boundary since we can accept increasingly fewer sub-optimal tours. However, some of the hardest soluble problems have regions where search increases exponentially as we move *away* from the phase boundary. With these problems, poor branching decisions early on result in an exhaustive (and unsuccessful search) for a tour of given length. Naïvely, accepting longer tours should make soluble problems easier, but Figure 5 clearly shows that the opposite can happen.

As an example, the worst case behaviour of 5,011,786 branches occurs at $l_{opt} + 750$, well away from the phase transition, in a region where over 90% of the problems are trivial. This worst case behaviour is a result of poor initial branching decisions followed by an inability of branch and bound to cut off search until a very late stage. In this case, the sixth branching decision made by the Hungarian heuristic was to go from city 1 to city 14, thereby missing out an essential side-trip between the two. In addition, the bound did not cut off search in this blind alley until only two branching decisions were left. At the optimal tour length, by comparison, the very tight bound enables backtracking to be initiated much earlier. The correct tour is thus found in just 22,129 branches. Sudden drops in difficulty in the worst case contour follow the appearance of new and significantly longer sub-optimal tours.

Problems such as the worst case in Figure 5 are often called “exceptionally hard problems”. They are of great interest as they appear *away* from the phase transition in satisfiable regions where almost all problems are very easy [9, 5]. Poor early branching choices followed by the inability to cut off search early has also been proposed as the source of exceptionally hard problems in both satisfiability [5, 8] and constraint satisfaction [16]. It remains an open question whether this behaviour is seen with all algorithms or is restricted to those like back-tracking which commit to early decisions. These results demonstrate that, contrary to conventional wisdom, the hardest problems can be *soluble*.

6 Real TSP problems

Random problems may, of course, not be representative of the problems met in practice. For example, they may lack features that can make real problems very hard. With time-tabling problems, we discovered that problem difficulty can be strongly influenced by the presence of large exam cliques [4]. There large scale structures were rare in our randomly generated time-tabling problems but present in our real data. Such large scale structures can make problems hard even well away from the phase transition. Phase transitions in randomly generated data should therefore be compared with phase transitions in real data.

As an example, we take some standard benchmark data from TSPLIB [15], the capitals of the 48 contiguous states of the U.S.A. To compute the order parameter for this data, we take A to be 7,825,118 km^2, the area of the 48 contiguous states. In Figure 6, we plot one view of a phase transition in this real data as we vary this order parameter, again using the branch and bound algorithm. We fix the number of capitals n at 24 and vary the tour length l from 500km to 15,000km in steps of 500km. We used the same 1000 sets of 24 capitals at each point. A rapid phase transition can be seen in Figure 6 at around $l \approx 8000$km and $l/\sqrt{n.A} \approx 0.6$. The worst case needed 2,933,071,577 nodes for a tour of length 8500km or less, in a region where 99.6% of the problems have a tour, and more than 1446km longer than the optimal tour for this problem. This is over *five* orders of magnitude worse than the worst median of 5,096 nodes at tour length 7500km, where 31.1% of tours were possible and worst case behaviour was 232,077,515 nodes. A similar phase transition occurs when the tour length is fixed and the number of capitals visited is varied [4].

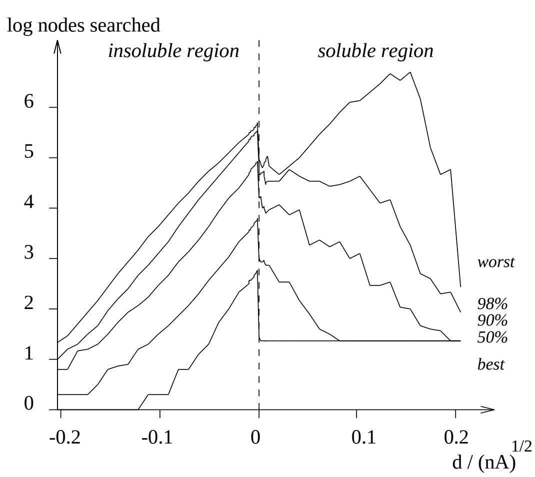

Fig 5: $l_{opt} + d$ for random problems

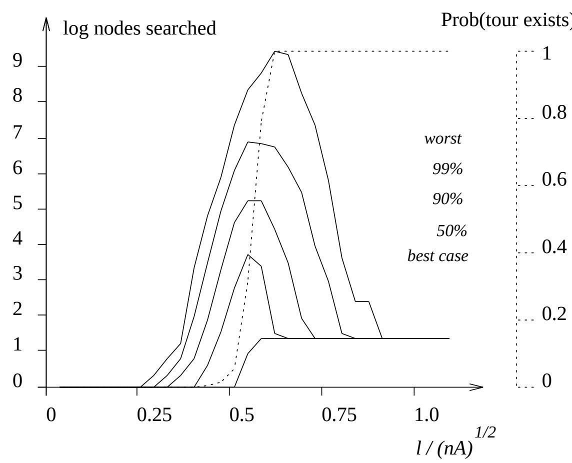

Fig 6: Varying tour length for real problems

The phase transition for real data is very similar to that seen with random data. Median problem difficulty again follows an easy-hard-easy pattern through the phase transition, whilst exceptionally hard problems occur in an under-constrained region where nearly 100% of problems are soluble. The major difference between real and random data is that real TSP decision problems are significantly harder than random ones. In [4], we conjecture that this may be because of the different distribution of cities in randomly generated problems compared to the real data. Exceptionally hard problems in satisfiable regions appear for similar reasons to random data: an early incorrect branching decision followed by the bound not cutting off search until very deep in the search tree.

7 Conclusions

Phase transition phenomena have been of considerable practical value in AI but have yet to receive much attention in OR. In this paper we have shown that for complete procedures like branch and bound applied to the traveling salesman problem, median problem difficulty typically follows an easy-hard-easy pattern through the phase transition. Exceptionally hard problems can, however, occur in under-constrained regions where nearly 100% of problems are soluble.

We have shown that phase transitions can be seen with both real and random data. Real data is, however, often significantly harder than random data. Phase transitions in real data are thus a very good source of hard problems for benchmarking algorithms. In addition to locating the really hard problems, phase transition phenomena obey some fascinating scaling laws, and we have empirically demonstrated one such law in this paper. Such scaling results promise to improve our understanding of both problem hardness and algorithm performance. As a consequence, phase transition experiments have an important role to play in both AI and OR.

References

[1] J. Beardwood, J.H. Halton, and J.M. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.

[2] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, pages 331–337. International Joint Conference on Artificial Intelligence, 1991.

[3] Ian P. Gent and Toby Walsh. An empirical analysis of search in GSAT. *Journal of Artificial Intelligence Research*, 1:47–59, September 1993.

[4] Ian P. Gent and Toby Walsh. Computational phase transitions in real problems. Research Paper 724, Dept. of Artificial Intelligence, Edinburgh, 1994. Submitted to IJCAI-95.

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

[6] Ian P. Gent and Toby Walsh. The hardest random SAT problems. In Bernhard Nebel and Leonie Dreschler-Fischer, editors, *KI-94: Advances in Artificial Intelligence. 18th German Annual Conference on Artificial Intelligence*, pages 355–366. Springer-Verlag, 1994.

[7] Ian P. Gent and Toby Walsh. The SAT phase transition. In *Proceedings of ECAI-94*, pages 105–109, 1994.

[8] Ian P. Gent and Toby Walsh. The satisfiability constraint gap. Research Paper 702, Dept. of Artificial Intelligence, Edinburgh, June 1994. Revised version to appear in *Artificial Intelligence* special issue on phase transitions in problem spaces.

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

[10] J. N. Hooker. Needed: An empirical science of algorithms. *Operations Research*, 42:201–212, 1994.

[11] S. Kirkpatrick, G. Györgyi, N. Tishby, and L. Troyansky. The statistical mechanics of k-satisfaction. *Advances in Neural Information Processing Systems*, 6, 1993.

[12] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, May 27 1994.

[13] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proceedings, 10th National Conference on Artificial Intelligence*. AAAI Press/The MIT Press, July 12-16 1992.

[14] Patrick Prosser. Binary constraint satisfaction problems: Some are harder than others. In *Proceedings of ECAI-94*, pages 95–99, 1994.

[15] G. Reinelt. TSPLIB - a traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991.

[16] Barbara M Smith and Stuart A. Grant. Sparse constraint graphs and exceptionally hard problems. Research Report 94.36, School of Computer Studies, University of Leeds, 1994.