On the Hardness of Decision and Optimisation Problems

John Slaney[1] and **Sylvie Thiébaux**[2]

Abstract. Recent work on phase transition has detected apparently interesting phenomena in the distribution of hard optimisation problems (find, on some measure, the least m such that the given instance x has a solution of value m) and their corresponding decision problems (determine, for a given bound m whether or not x has a solution of value $\leq m$). This paper examines the relationship between the hardness of optimisation and that of decision. We identify an expression for the latter in terms of the former together with the distance between the bound and the optimal solution size. These results explain both the appeal and the shortcomings of other accounts in the literature. We validate our analysis by showing that it predicts the hardness of Blocks World decision problems with much improved accuracy.

1 MOTIVATION

The empirical study of phase transition phenomena has proved to be a fruitful approach to the location of hard problems [2, 8, 9]. The most intensive research in this area has concentrated on satisfiability and constraint satisfaction problems, for which one typically observes, around the critical value of some order parameter, both an abrupt transition in the probability that solutions exist and a peak of hardness.

Concerning optimisation problems and their corresponding decision problems, things are however less clear. By a optimisation or measurement problem[3] we mean the problem of finding, on some measure, the least m such that a given instance x has a solution of value m. By a corresponding decision problem, we understand that of determining for a given m and x whether or not x has a solution of value $\leq m$.

For such problems, even the correlation between phase transitions and hardness is in question. For instance, Kirkpatrick and Selman observe that Traveling Salesman (TSP) and Max-Clique 'lack a clear phase boundary at which hard problems cluster' [9, p. 1300], while Gent *et al.* have recently concluded that peaks of hardness for TSP are associated with phase transitions in the probability of existence of a solution smaller than a given size [6, 4]. Similarly, according to Fu [3] 'no phase transition of any kind is found' for Number Partitioning, while Gent and Walsh [5] again claim the contrary.

This paper makes two main contributions to the clarification of this issue: we analyse the nature of the observed phase transitions and peaks of hardness in the decision problems, and we show that these phenomena are best understood in relation to the corresponding optimisation problems.

We first consider the methodology suggested by Gent *et al.* in the cited papers. We show that the phase transitions detected that way measure nothing but the average optimal solution size of the sample used. As for the corresponding peaks of hardness, they merely acknowledge that for each optimisation problem, the hardest corresponding decision problems are those with bound m close to the optimal. This, while true, misses an important dimension of what makes decision problems hard or easy, and for the same reason, gives no insight into the hardness of the corresponding optimisation problems.

As a step towards a fuller account, we identify a simple relation suggesting that the hardness of a decision problem is polynomial in the hardness of the corresponding optimisation problem and exponential in the distance separating the required bound from the optimal solution size. This relation implies that for a fixed distance between the bound and the optimal, the same parameters control the hardness of the decision and optimisation problems. Experiments by Gent *et al.* ignore these parameters, capturing only two of the three dimensions in this relation.

Throughout the paper, we will illustrate our claims using the NP-complete problem of optimal planning in the Blocks World[4](BW) [7, 11]. In the final section, we look at this example in more detail. In particular, we use the above relation to predict the hardness of BW decision problems from simply observable properties of BW instances. Appendix A describes the algorithm we used for our experiments on BW.

2 THE 1.70m PHASE TRANSITION

In their experiments on random TSP instances, Gent and Walsh [6] found that for each number n of cities, the probability of existence of a tour shorter than a given length exhibits an abrupt phase transition around some critical tour length. They also measured the hardness of deciding whether a random instance of TSP has a tour of length less than a given bound, and noticed that it peaked at a bound close to the critical tour length.

[1] Automated Reasoning Project, Australian National University, Canberra, ACT 0200, Australia. e-mail: John.Slaney@anu.edu.au
[2] CSIRO Math. & Info. Sciences, PO Box 664, Canberra, ACT 2601, Australia. e-mail: Sylvie.Thiebaux@cmis.csiro.au

[3] Strictly, optimisation involves *producing* a minimal solution while measurement merely involves determining its size. We use the word optimisation whenever this makes sense, since in those cases the algorithms we consider actually produce an optimal solution.

[4] The reader seeking familiarity with this example may gosub the first paragraph of Section 4 at this point. However, Sections 2 and 3 do not presuppose any deep knowledge of Blocks World.

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

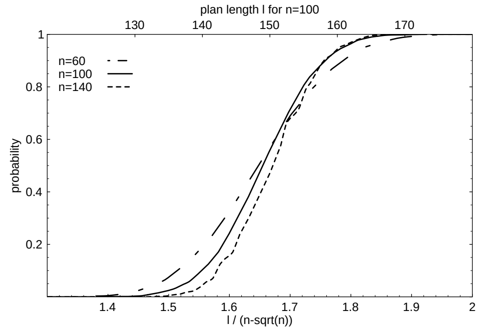


Figure 1. Probability that BW instance has a plan shorter than l, plotted against order parameter $l/(n-\sqrt{n})$, for $n = 60$, $100, 140$ blocks.[6] 3000 BW instances were used at each n.

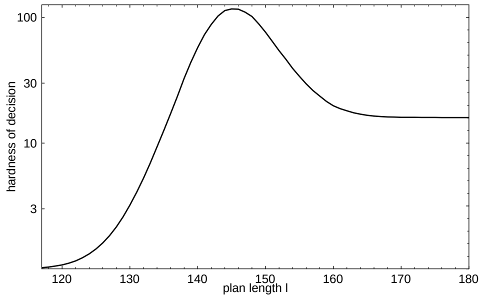


Figure 2. Average hardness (geometric mean) of BW decision problem plotted against length bound l, for $n = 100$ blocks. The same 6000 BW instances were used at each l.

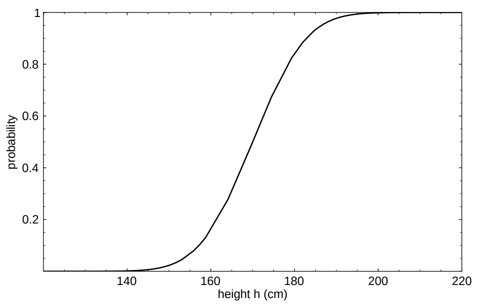


Figure 3. Phase transition in Height decision problem. A normal distribution with mean 170 and standard deviation 10 was assumed.

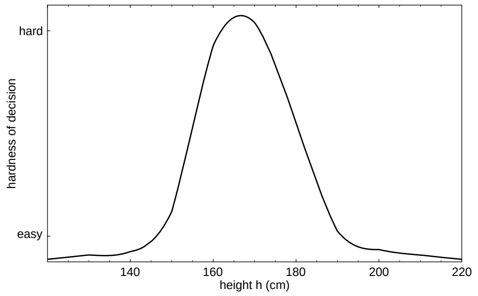


Figure 4. Hardness peak in Height decision problem. The difficulty for instance x and height bound h was assumed to be $(|\mathrm{Height(x)} - h| + \epsilon)^{-1}$

Following this methodology, we observed similar results for random BW instances,[5] confirming the remark that 'the phase transition phenomena described here appear to be widespread' [6, p. 356]. As shown in Figures 1 and 2, there is a phase transition in the probability of a BW instance having a plan shorter than a given length l, and the average difficulty[7] of deciding whether such a plan exists peaks at this phase transition. Note that as for TSP and other NP-complete problems, the sharpness and location of the phase transition increase with the size of the problem. For $n = 100$ blocks, the critical plan length is around 147.

Unfortunately, such phase transitions appear to be a bit *too* widespread. In fact, one can reproduce similar results for the decision version of basically any measurement problem, regardless of whether it involves any combinatorics or any kind of reasoning.

For instance, let us take a problem of absolutely no computational interest: measuring the heights of a random group of adults. The distribution of heights is roughly normal, 2/3 of us having heights within a standard deviation of the average and so on. In other words, when we plot the probability of an individual being smaller than a given height h (Figure 3), we get the same sort of curve as was observed for BW and TSP.

So the first point is that this sort of S curve does not in itself establish *anything* interesting about constrainedness: it just records the mean and standard deviation of something similar to a normal distribution (of optimal plan lengths or shortest journeys or whatever). Indeed it is well-known that optimal tour lengths for the type of TSP instances used in [6] cluster very strongly around their mean [1]. Experiments suggest a similar clustering in case of other types of random TSP instances [4], and in case of our uniformly distributed BW instances [11].

Now, regarding the height of any individual, there are two kinds of problem: the *measurement problem* is to establish what his height is, and the *decision problem* for a given height h is to determine whether or not his height is less than h. We know there is no variation in the difficulty of the measurement problem: it is just as hard to measure a 1.75m person as a 1.50m or a 2.0m one.

However the same is *not* true of the decision problem. It is trivial to decide whether our favorite basketball player is less than 1.50m tall, just by casual looking. On the other hand, it is quite difficult to detect whether a person of 1.51m is less than 1.50m tall: indeed it is virtually as hard as having to measure the person. Obviously, the closer to 1.50m the subject is, the more effort it takes. In general, the difficulty of the decision problem for individual x and height h is antitone with $|\mathrm{Height(x)} - h|$, with an upper bound where the two are indiscriminable by all available methods.

Recall that the heights of people cluster around the mean. So, when one takes people at random and plots, for each choice of h, the average difficulty of the decision problem, one gets something like Figure 4. What does this tell us about the hardness of the Height Problem? Nothing! It just shows that most people have height close to h if h is near the mean, while most of them have heights very different from h if h is a long way above or below the mean.

[5] These instances are uniformly distributed and generated using the method described in [11].

[6] Our order parameter follows the conjecture in [11] that average optimal plan lengths tend to $K(n - \sqrt{n})$ in the limit, for some constant $K \leq 2$.

[7] Difficulty is measured as the number of branches explored by the algorithm described in Appendix A.

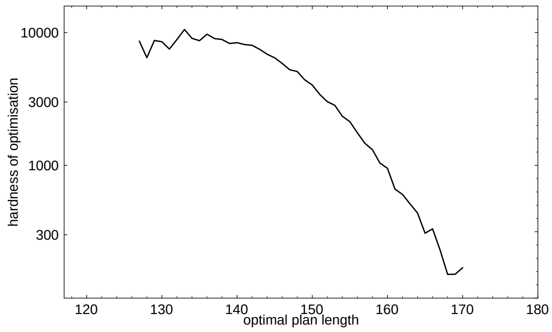


Figure 5. Average hardness (geometric mean) of BW optimisation problem plotted against optimal plan length, for $n = 100$ blocks. 100000 BW instances were used.

To sum up, there are several messages from this thought experiment. The first is that one should be very careful to avoid results such as the one concerning height[8] These misleading curves are likely where the order represented on the x-axis of the graph encodes (more or less explicitly) the bound of the decision problem [4, p. 249] [5] [6]. Once we abstract from the clustering around the mean, all that remains from the experiments is that for each instance x, the hardest decision problems are those with bound close to x's optimal solution. This only gives a partial insight into what makes decision problems hard or easy. In particular, it does not tell us anything about the relative hardness of two decision problems with different x, nor about any difference in hardness between the corresponding optimisation problems.

The second message then, is that the above type of peak of hardness in the decision problem does not transfer to the optimisation or measurement problem. It may be that this latter does not exhibit any such peak of hardness, as is the case for the Height measurement problem, and as seems to be the case for optimally partitioning numbers drawn from a fixed interval. It can also happen that there is a peak of hardness but at a place unrelated to the average optimal solution size. As shown in Figure 5, an example of this is the BW optimisation problem, whose average hardness peaks significantly below the critical plan length of 147.

3 THE THIRD DIMENSION

In order to confirm the above analysis and to explore the relation between hardness of optimisation and decision problems in more detail, we conducted the following experiment. We examined the same 6000 BW instances as in Figure 2. For each instance, we recorded the hardness of the optimisation problem and the optimal plan length, and then recorded the hardness of the corresponding decision problems for bounds varying from optimal plan length -20 to optimal plan length $+10$. Figure 6 shows the average hardness of the decision problem as a function of the hardness of the corresponding optimisation problem and the distance between the given bound and the optimal plan length.

[8] Should the reader be concerned that the height example lacks a parameter n giving the size of the problem, we invite him to imagine that instead of measuring people we measure a set of rods, the length of each rod in cm being the number of heads in n tosses of a fair coin ...

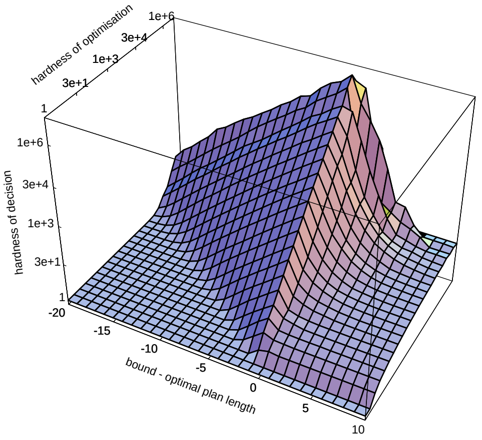


Figure 6. Average hardness (geometric mean) of BW decision problem plotted against difference between bound and optimal plan length, and hardness of optimisation, for $n = 100$ blocks. 6000 BW instances were used.

The curve displays two distinct regions: the negative decisions region where the given bound is lower than the optimal plan length and so the answer to the decision problem is negative, and the positive decisions region. As expected, for a fixed hardness of the optimisation problem, the hardness of the decision problem decreases exponentially with the distance separating the bound from the optimal length. Further experiments indicate that this holds not only on average but also typically for individual instances. The hardest decision problems are at the border between the two regions, for a bound equal to optimal plan length $- 1$.

A qualitative explanation is as follows. Let $\mathrm{L}(x)$ be the optimal plan length for instance x, $\mathrm{N}(x)$ be the number of optimal solutions for x, d the difference between the bound given to the decision algorithm and $\mathrm{L}(x)$, and b the branching factor of the search tree (which we assume constant). If $d < 0$, i.e. we are in the negative decision region, the algorithm has to prove that there is no solution in a tree of size $b^{\mathrm{L}(x)+d}$. Even without further analysis, we may therefore expect the number of branches explored to increase exponentially with d. When $d = 0$, the algorithm needs on average to explore $b^{\mathrm{L}(x)}/\mathrm{N}(x)$ branches to find a solution. If there are enough optimal solutions so that $\mathrm{N}(x) > b$, then we expect this to be easier than the negative decision for $d = -1$. When $d > 0$, the number of solutions increases by a factor of more than b from one level of the tree to the next, because each solution at a given level subsumes many others at the next level (including its own b successors), and new solutions are added. At some level, the leftmost solution in the tree is found, after which no decrease in hardness occurs.

So much for the first two dimensions of this three-dimensional relation. Now, for a fixed difference d between the bound and the optimal length, the hardness of the decision problem increases roughly polynomially with that of the optimisation problem[9] This arises because as noted in [6] solving an optimisation problem amounts to solving the two decision problems with bounds immediately either side the optimal solution size, and because the hardness of these is polynomially related to that of the decision problem at any other fixed d (see above).

[9] There is a linear relationship between the log of the two figures, hence they are polynomially related.

Putting it all together, for each of the two regions, there exist numbers α, β, and γ such that for a large range of non-trivial decision problems for instance x:

$$D(x, L(x) + d) \simeq \alpha^d \, (O(x))^\beta \, \gamma$$

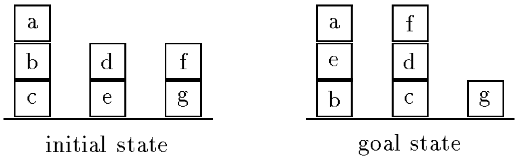
where d and $L(x)$ are as before, $\mathrm{O}(x)$ is the hardness of the optimisation problem for instance x and $\mathrm{D}(x, l)$ is the hardness of the decision problem for instance x and bound l. For instances x with $n = 100$ blocks and for d in the negative region, least square interpolation with the data in Figure 6 yields $\alpha \simeq 1.92$, $\beta \simeq 1.15$ and $\gamma \simeq 0.85$.

The curve in Figure 2 and the analyses of TSP given in [4, 6] amount to averaging the above relation over x. By so doing, they sacrifice most of the information clearly visible in Figure 6. In particular, they obscure the fact that whatever parameters control the hardness of optimisation problems also control that of the corresponding decision problems at each d. This fact is important because it implies that any feature that is a predictor of the former can also be used to predict the latter. In the case of TSP, the standard deviation of inter-city distances is such a feature [2, 13]. In the next section, we shall identify features which play this rôle for BW and use them to predict the hardness of BW decision problems.

4 BLOCKS WORLD

A BW instance consists in an initial and a goal configuration of blocks (see Figure 7). The problem is to transform the one into the other by moving one block at a time from the top of a tower onto another tower or to the table. Each block is either *in position* (the sequence of blocks leading down from it to the table is already the same as in the goal) or else *misplaced*. All and only the misplaced blocks have to move in order to solve the problem. Once a block has been moved into position, it need not be moved again. If at some point no block can directly be moved into position, then there is a *deadlock* which must be broken by moving one of the blocks involved to the table. Hence, the length of the plan is equal to the number of misplaced blocks plus the number of deadlock-breaking moves. The key to finding a shortest plan is therefore to find the smallest set of blocks sufficient to break all the deadlocks, that is, to find a minimal cardinality *hitting set* of the deadlocks. Technically, a deadlock is a set $\{b_1, \ldots, b_k\}$ of blocks such that for all $i < k$ a certain relation $B(b_i, b_{i+1})$ holds, and to complete the loop $B(b_k, b_1)$. $B(x, y)$ means that x and y are misplaced and there is some block z currently somewhere below y which will be below x in the goal. See [7, 11] for more details.

Under what conditions is this problem hard? Well, it is easy if there are few deadlocks, and it is easy if all the deadlocks are small. Unfortunately, these two conditions tend to work against each other. Few deadlocks exist if the B relation is sparse, which typically occurs where there are many short towers. Small deadlocks occur where the B relation loops very easily, which happens when most of the blocks are found in a very few tall towers. The hard problems are therefore squeezed between the "under-constrained" area in which there are many towers and the "over-constrained" area in which there are few (see Figure 8). The number of towers may therefore be used to estimate the hardness of optimisation.



plan: 1. move a to table, 2. move b to table, 3. move d to c, 4. move e to b, 5. move a to e, 6. move f to d

Figure 7. BW planning problem and optimal plan

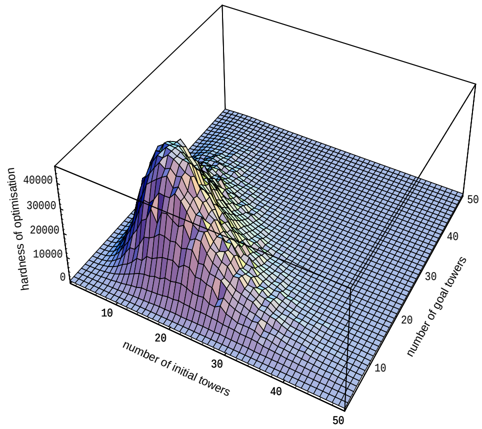


Figure 8. Average hardness of BW optimisation problem plotted against the number of towers initially and in the goal, for $n = 100$ blocks. 4 million BW instances were used. The hardness peaks around 14 towers in the initial and goal states.

In order to predict the hardness of a decision problem, we need to estimate not only the hardness of the corresponding optimisation problem but also the optimal plan length. Fortunately, a good prediction may be obtained from the upper bound returned by a near-optimal algorithm such as those described in [11]. These have excellent average performance ratios. For instance, even in the hard region near 14 towers, they give plans that are on average only about 3% longer than the optimal. By correcting the returned near-optimal plan length with the average performance ratio for the region determined by the number of towers in the instance, we are able to estimate the optimal plan length even more closely.

The number of towers can be computed in $\mathcal{O}(n)$, and so can the near-optimal plans [11]. Insofar as these are good predictors, we can use them together with our three-dimensional relation to predict hardness of decision. We therefore conducted the following experiment. We generated 2000 new random BW instances, counted their initial and goal towers, and used the data in Figure 8 to estimate their hardness. We generated near-optimal plans for these instances, and predicted the optimal plan lengths. We then used the data in Figure 6 to predict the hardness of decision for these instances, with bounds varying from -20 to $+10$ of the predicted optimal.

The solid line in Figure 9 compares this prediction to the actual hardness. The least-square fit approximates the actual hardness reasonably well (see dotted line). Note however that since the number of towers is only a rough guide to the hardness of optimisation, the R^2 of the fit is only around 50%. Given higher-order polynomial computation time, substantially better predictors can be obtained [12]. A full account of these is beyond the scope of this paper.

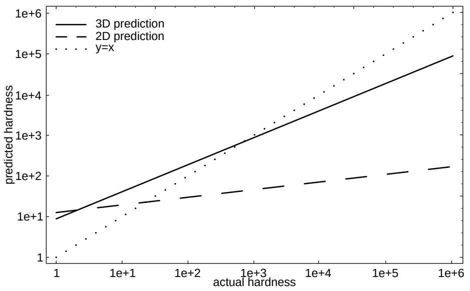


Figure 9. Least-square fit to predicted hardness of decision, plotted against actual hardness, for $n = 100$ blocks. 2000 BW instances were used yielding 62000 data points

The two-dimensional analysis derived from [6] suggests that the bound itself is a predictor of hardness. The dashed line in Figure 9 shows the results of using the data in Figure 2 in place of those in Figure 6, on the same decision instances. It is clear that predictions based on the three-dimensional analysis are a much better reflection of reality.

5 CONCLUSION AND FUTURE WORK

We have analysed the relationship between the difficulty of decision problems, the corresponding optimisation problem, and the bound within which a solution is sought. Our account clarifies recent results reported in [6]. It also gives some evidence that for a fixed distance between the bound and the optimal, the factors influencing the hardness of decision are those influencing the hardness of optimisation. This latter point has been demonstrated for the BW example; further experimentation is necessary to confirm our expectation that it will also hold in other domains.

Note that using our three-dimensional relation to predict the hardness of decision requires us to predict the optimal solution size fairly accurately on average. Although we can do this for the type of BW, TSP and number partitioning instances considered in this paper, it will not be the case in all domains.

Another intriguing possibility offered by this relation is to predict the optimal plan length for an instance by observing the hardness of a few decision problems. Indeed, note that a slice through Figure 6 consists roughly of two straight lines intersecting just below the optimal plan length, and that this is also tends to be the pattern for individual instances. Hence, given a reasonable initial estimate of the optimal plan length, four relatively easy decisions (two negative and two positive) should suffice for a near-perfect prediction. This in turn yields an estimate of the hardness of optimisation. Again, we expect that this technique will prove useful for a range of other NP-complete problems.

ACKNOWLEDGEMENTS

We thank Jeremy Frank, Eric Jacopin and Toby Walsh for helpful comments. Discussions with Jeremy Frank contributed to some of the ideas presented in this paper.

REFERENCES

[1] J. Beardwood, J. H. Hamilton, 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. Taylor, 'Where the *Really* hard problems are', in *Proc. IJCAI-91*, pp. 331–337, (1991).

[3] W. Fu, *The uses and abuses of statistical mechanics in computational complexity*, 815–826, Addison-Wesley Longman, 1989.

[4] I. Gent, E. MacIntyre, P. Prosser, and T. Walsh, 'The constraindness of search', in *Proc. AAAI-96*, pp. 246–252, (1996).

[5] I. Gent and T. Walsh, 'Phase transitions and annelead theories: Number partitioning as a case study', in *Proc. ECAI-96*, pp. 170–174, (1996).

[6] I. Gent and T. Walsh, 'The TSP phase transition', *Artificial Intelligence*, **88**, 349–358, (1996).

[7] N. Gupta and D. S. Nau, 'On the complexity of blocks-world planning', *Artificial Intelligence*, 223–254, (1992).

[8] T. Hogg, B.A. Huberman, and C.P. Williams, 'Phase transitions and the search problem', *Artificial Intelligence*, 1–15, (1996).

[9] S. Kirkpatrick and B. Selman, 'Critical behaviour in the satisfiability of random boolean expressions', *Science*, **264**, 1297–1301, (1994).

[10] J. Slaney and S. Thiébaux, 'How best to put things on top of other things', Technical Report TR-ARP-06-96, Automated Reasoning Project, Australian National University, (1996).

[11] J. Slaney and S. Thiébaux, 'Linear time near-optimal planning in the blocks world', in *Proc. AAAI-96*, pp. 1208–1214, (1996).

[12] J. Slaney and S. Thiébaux, 'Phase transitions and optimality: Sense and nonsense', Technical Report TR-ARP-09-97, Automated Reasoning Project, Australian National University, (1997).

[13] W. Zhang and R.E. Korf, 'A study of complexity transitions on the asymetric traveling salesman problem', *Artificial Intelligence*, 223–239, (1996).

A THE ALGORITHM

Solving a BW problem instance optimally amounts to finding a minimal cardinality hitting set for the set of deadlocks in the instance. The deadlocks are initially unknown and are computed as needed. Our algorithm performs a depth-bounded depth-first traversal of a tree whose nodes are labelled with sets of blocks and whose edges with single blocks. The set at the root node is the union of the deadlocks of cardinality 1 (this can be computed in quadratic time) and the edges linking each node to its successors are labelled with the blocks in some known deadlock, minus any already on edges above and to the left of the node so as to avoid visiting the same set twice. Each successor of a node N is labelled with the set labelling N plus the block on the relevant edge.

If all known deadlocks are hit by the set at a node, that set is tested against the problem instance to see whether it is a solution. If not, a deadlock which it does not hit is found (in quadratic time) and added to the list of those known. Only inclusion minimal deadlocks are used. If no solution exists within a given depth bound, the bound is incremented and the search repeated—more efficiently given that more deadlocks are now known. For the decision version of the problem, the depth bound is set in advance and not incremented.

Thus the search may be viewed as traversing the lattice of sets of the blocks, with a high degree of pruning. For further efficiency, we apply the 'most constrained variable' heuristic, which in this case means branching on smaller deadlocks before larger ones. For more details of the algorithm see [10].