EFFICIENCY ANALYSIS OF MULTIPLE-CONTEXT TMSs IN SCENE REPRESENTATION

Gregory M. Provan[1]

Mathematical Institute, Oxford University
Oxford, England OX1 3LB

Abstract *Multiple possible solutions can arise in many domains, such as scene interpretation and speech recognition. This paper examines the efficiency of multiple-context TMSs, such as the ATMS, in solving a scene representation problem which we call the Vision Constraint Recognition problem. The ATMS has been claimed to be quite efficient for solving problems with multiple possible solutions, even for problems with large databases. However, we present evidence that for large databases with multiple possible solutions (which we argue occur frequently in practice), such multiple-context TMSs can be very inefficient. We present a class of problems for which using a multiple-context TMS is both intrinsically interesting and ideal, but which will be computationally infeasible because of the exponential size of the database which the TMS must explore. To circumvent such infeasiblity, appropriate control must be exerted by the problem solver.*

1 Introduction

The TMS is one of the most important general AI algorithms developed, and has been applied to a wide range of areas, including qualitative process theory—[4]; circuit analysis— [6]; analog circuit design—SYN [5]; and vision—[7], [1]. [1]

In this paper we examine more closely the performance of multiple-context TMSs ([2], [10],[12]) on certain problems which generate a large number of contexts. Problems with a large number of contexts and multiple possible solutions are not artificial, and can arise in many domains, such as scene interpretation ([1],[7], [8]) and speech recognition/understanding [9]. In vision, one is typically dealing with noisy, ambiguous data with complex local/global constraint interactions. In text understanding, each sentence may, on its own, have many different interpretations, and one is attempting to piece together many such localized interpretations to develop an holistic meaning. Many equally plausible solutions arise in the presence of ambiguous constraints, giving rise to multiple possible local interpretations for each such constraint. And typically, these local interpretations interact in complex manners to produce many feasible global interpretations.

We investigate the use of the TMS in solving high-level vision problems as a means of better understanding multiple-context TMSs. High-level vision is an ideal domain for studying multiple-context TMSs, and specifically the ATMS ([2],[3]) because of the ubiquity of multiple simultaneous interpretations. It is precisely this ability to generate multiple simultaneous solutions that has prompted the use of the ATMS in a variety of areas, e.g. [4], [6]. Single-context TMSs, also known as Justification-based TMSs (JTMSs), e.g. [11], are less well-suited to solving such problems because their strict adherence to a single consistent context (interpretation) represents an inadequate method of attacking the problem.

Regarding the ATMS, de Kleer, in [2] states: " the observed efficiency of the ATMS is a result of the fact that it is not that easy to create a problem which forces the TMS to consider all 2^n environments without either doing work of order 2^n to set up the problem or creating a problem with 2^n solutions." We present the Vision Constraint Recognition System (VCRS) [13] (1) as a novel means of solving certain high-level vision problems, but (2) also as evidence that there naturally exist domains in which multiple context TMSs are forced to consider an exponential number of solutions. Also, we state results of a complexity analysis of multiple context TMSs corroborating the VCRS's evidence that, for complex visual recognition problems, such TMSs are often forced to explore a number of contexts exponential in the size of the database, this number of contexts generated by problems with an exponential number of either final or partial solutions. As a consequence, such TMSs will be inefficiently slow in solving such problems.

The rest of this paper is organized as follows. In Section 2, we briefly describe the VCRS, discussing the reasons for and advantages gained by using an ATMS for a visual recognition system which instantiates a figure in an image consisting of overlapped rectangles. Then, we conduct a simple combinatorial analysis of the effect of nogoods in reducing the search space explored by multiple-context TMSs, and hence comment on the efficiency of such TMSs.

[1]The author gratefully acknowledges the support of a Scholarship from the Rhodes Trust, Oxford.

2 Vision Constraint Recognition System (VCRS)

Perception can be considered an *interpretive process*, and a key problem is interpreting descriptions computed for a scene against a (typically large) database of models. Many examples such as Rubin's vase, Necker's cube etc. teach us that a single image (e.g. perfect line drawing) can have several equally plausible perceptual interpretations. The problem we are solving exemplifies these characteristics. We use a multiple-context TMS precisely because of its ability to generate multiple possible interpretations.

The specific high-level vision problem we have studied is called the Constraint Recognition problem, and is a generalization of the PUPPET problem, first studied by Hinton [8]. The problem solved by the VCRS is as follows: given a set of (2-D) randomly overlapping rectangles and a relational and geometric description of a figure (as described by a set of constraints over the overlap patterns of k of these rectangles), find the best figure if one exists. Our use of a TMS generalizes Hinton's integer relaxation-based techniques by recognizing that the set of justifications which the TMS maintains for any database assertion is isomorphic to an explicit perceptual interpretation for that assertion.

Our plan for applying a multiple-context TMS to this problem is as follows:

1. A TMS generates a justification structure for each node, the structure indicating how that node was assigned a label. This justification structure corresponds to a perceptual structure (e.g. rectangle A is *seen as* a trunk, because rectangle B is *seen as* a neck) by appropriate spatial relationships, etc.

2. Different perceptual interpretations correspond to different contexts.

3. Locally plausible visual fragments can be interpreted in many ways and each interpretation is accorded a context.

Taken together, the above points imply a large number of contexts even for moderately complex visual input.

2.1 Advantages of Using a TMS for Visual Interpretation

Let us now outline the advantages over relaxation-based methods (e.g. [8]) afforded by a multiple-context TMS.

Explicit semantics for images via justifications. TMSs explicitly store justifications for all labeling assignments. Thus if rectangle G is assigned the label "hand," *the reason for that* assignment, e.g. that rectangle F, labeled "forearm," overlapped G according to some constraint, is stored.

Studying many different alternative solutions. An algorithm which can explore multiple interpretations simultaneously is more useful than one which explores one context at a time, as locally contradictory interpretations (which are outlawed for a single-context TMSs) may not necessarily indicate global inconsistency but multiple global interpretations.

Utilizing updated input. The *truth maintenance* aspect of TMSs enables updating of databases with the input of new information. Both [1] and [7] use the TMS for creating a consistent interpretation of stereo data, for example.

Constraint-exposing. Such a notion of semantics forms the basis for a powerful constraint-exposing process, one example of which is contradiction flagging. By tracing justification paths for the nodes in a nogood back to the assertions causing the contradiction, (identical to dependency directed backtracking), we can identify incorrect/impossible assertions, rule these out, and consequently eliminate all possible solutions based on these global inconsistencies. In this manner, we can rule out large portions of the search space.

Robust given noise. Scenes with noisy data occur frequently, and a TMS can extract interpretations from noisy/ambiguous situations. This is achieved by the TMSs' justification structure "cutting through" noise. Rectangles extraneous to the figure (e.g. a puppet) will not be included in the justification structure and will be ignored by the system.

Robust given occluded/incomplete scenes. This occurs via two mechanisms:

1. Automatic default mechanisms: these are incorporated in the TMS and can be used to fill out incomplete (but plausible) figures.
2. Justified default mechanisms: the justification structure has an explicit notion of "completeness" of a figure, and can flag an almost-perfect figure using a "closeness relationship" with respect to this notion of completeness. This gives a semantics for the notion of defaults; for example, we might have "this default is an arm because this figure would be a perfect puppet if such an arm were present".

Explicit (domain dependent) constraints. An explicit notion of domain-dependent constraints has been found necessary to provide a powerful means of reducing the search space. For example, in the detection of puppet figures, such constraints include the representation of geometric structure in terms of *posture* and *global scaling.* A puppet having a right and left side, an upright or reclining posture introduces much more powerful constraints (which can significantly reduce the search space) than if those concepts were not present. Hence, an arm being a right arm rather than a left arm determines the allowable angle of the elbow joint quite specifically. A sense of global scaling is also crucial, as a thigh can be a thigh only in proportional relation to the trunk and calf to which it is attached.

2.2 Performance of TMS within the VCRS

Let us now briefly outline some simple examples of problems the VCRS can solve. [2] In Figure 1, we see a sample input for the VCRS, randomly overlapping rectangles in which the target figure, a puppet, is distinguishable. Figure 1 shows a much-simplified example of the program's operation. Here we have a situation in which four orientations can produce a puppet, taking either A, B, C, or D as a head. For example, if B is chosen as a head, the partial puppet (head, neck, trunk) consists of rectangles B, B', E. Moreover, it is ambiguity such as shown in this Figure that gives rise to multiple contexts during search for a solution, as well as multiple possible solutions. When processing complicated scenes in searching for puppets, a multiple context TMS builds a context for each possible puppet figure interpretation. Even for relatively simple cases we have discovered that the number of contexts formed can be unreasonably large. In the above example, four environments are necessary for just a small number of rectangles (A, A', B, B', C, C', D, D', E).

Let us now look at two inter-related reasons why a very large number of environments will need to be constructed for this problem, which results in the ATMS creating an exponentially large number of contexts.

Size of nogoods expected For complex figures, once we have found a seed, it is reasonably easy to form the first few elements of the figure, and it then becomes increasingly difficult, with inconsistencies more liable to occur. This means that, of the seeds found, the majority of the nogoods found will be of size $\geq k$, with k dependent on the complexity of the problem. Thus, if there are 100 seeds found and $k \approx 10$, the actual space which must be searched is extremely large, as the nogoods of large size, as shown in Section 3, will not reduce the search space very much even if there are many such nogoods.

Expected number of partial solutions The number of environments constructed increases rapidly as problem-solving progresses. Consider a partial puppet consisting of a head, neck and trunk (A, A', E respectively). As in Figure 1, if this trunk has 4 overlaps which could be upper arms and 4 which could be thighs, we can have $\binom{4}{2}^2 = 36$ possible interpretations. Now, if an upper arm and a thigh have 2 possible fore-arms and calves respectively, this gives $\binom{5}{2}^2 = 100$ interpretations. Even for this very simple example we can already see the combinatorial explosion of the number of necessary environments. This combinatorial explosion grows even faster (i.e. is more serious) the more complex the scene and the figure for which we are looking. This points out that, even if we end up finding just a few full figures, there may be an exponential number of environments for the partial figures at an intermediate stage of the solution process.

[2] For full details consult [13].

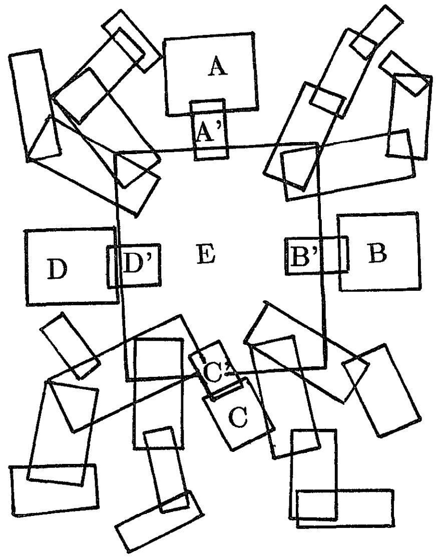

Figure 1: VCRS Example for Detecting a 15-element Puppet

We now present theoretical evidence to corroborate these empirical results.

3 Combinatorial Analysis of Multiple Contexts

Questions concerning the complexity of the ATMS were first mentioned with reference to a parity problem ([3], [11]). We will now analyse some issues raised by problems such as the parity and visual constraint recognition problems. But before beginning this analysis, we shall formally state the problem.

3.1 Problem Definition

Consider that we have a problem with n distinct facts, forming the fact set A. We call the set of *environments* the power set of A, $\mathcal{A} = \mathcal{P}A$. Within this power set there are subsets which are inconsistent. We call such inconsistent subsets *nogoods*, and the consistent subsets *contexts*. We denote the set of contexts $C \subseteq \mathcal{A}$. There are 2^n environments and $\binom{n}{k}$ environments with k facts. A *minimal nogood* is a subset B from which removing a single fact will leave either the null set or a context. It is important to note that all supersets of a nogood set are also nogood sets. Let us call the size of the minimal (or "seed") nogood set α, size meaning the number of facts contained in the nogood.

In the following discussion, we shall be referring to a general algorithm which attempts to determine all maximal contexts, where a *maximal context* is a set $C^* \subseteq C$ such that either: (1) $\mid C^* \mid = n$, or (2) $C^* \cup \{a\}$ is inconsistent for all facts $a \in A \setminus C^*$. Such an algorithm proceeds by forming all subsets (representing partial solutions), first of size 1, then of size 2, etc. until we produce maximal contexts. Nogoods are used to prune the search space by eliminating all supersets of minimal nogoods from the search space. It must be noted that, in its full generality, this algorithm, referred to as interpretation construction in [2], is isomorphic to the minimum set covering problem (which is NP-complete). The ATMS utilizes the most efficient method of interpretation construction given the specific problem, but for certain problems the exponential complexity is unavoidable, and is unavoidable for any algorithm searching multiple contexts.

The example of algorithm which we shall be using is the ATMS, although this analysis is equally valid for algorithms which use a similar multiple-context approach. We will now isolate the factors necessary to avoid exponential growth of the search space. In this analysis, we show the power of nogoods of small size in cutting down the number of contexts, and hence the size of search space. We also see that even for problems in which the number of solutions is non-exponential in the problem size n, the number of partial solutions could still be very large, and hence produce an unreasonably large number of contexts.

3.2 Analysis of Search-Space Reduction Using Nogoods

We begin this combinatorial analysis by looking at how nogoods reduce the search space. We introduce the problem with the simplest case, that in which the seed nogoods are non-overlapping. An *overlap* occurs between two seed (or minimal) nogoods ng_1 and ng_2 if $ng_1 \cap ng_2 \neq \emptyset$. A non-overlapping problem is one in which none of the seed nogoods have overlaps: for the set $\mathcal{N}$ of seed nogoods, $ng_i \cap ng_j = \emptyset$, $\forall ng_i, ng_j \in \mathcal{N}$, $i \neq j$. We then proceed to more general cases, analyzing the complex nogood interactions when we have overlapping of seed nogoods. Due to space limitations, we provide just a sample of our results without proofs, and refer the reader to [13] for these proofs and a more intelligible analysis.

3.2.1 Non-overlapping Nogood Analysis

Lemma 1 *For a problem with n distinct facts, x "seed" (minimal) non-overlapping nogoods each of size α produces $\Phi(x,\ \alpha)$ total nogoods, where*

$$\Phi(x,\ \alpha) \ = \ 2^{n-\alpha}\Big\{x - 2^{[-x(\alpha-1)+\alpha]}\Big(1 - \frac{(1+x)}{2^x}\Big)\Big\}.$$

Lemma 1 describes the size of space generated by non-overlapping nogoods all of equal size.

Lemma 2 *For a non-overlapping problem with a nogoods of size α, b nogoods of size β, c nogoods of size γ, etc., an upper bound for the number of nogoods formed is given by*

$$\Phi((a,\alpha),(b,\beta),(c,\gamma),..) \ \leq \ 2^n(a2^{-\alpha} + b2^{-\beta} + c2^{-\gamma} + ...).$$

Lemma 2 extends Lemma 1 to cases of non-overlapping nogoods of different sizes.

Given that we know the search-space reduction achieved by non-overlapping nogoods, we next investigate the reduction achieved by nogoods of specific sizes.

Corollary 1 *For all n, $\alpha > 0$, if $\Delta_x = \frac{\Phi(x,\ \alpha)}{2^n}$, $x \geq 2$, then, to a close approximation, $\tilde{\Delta}_x$,*

1. *$\tilde{\Delta}_x$ is constant independent of n,*
2. *The effect of a nogood in reducing the search space is inversely proportional to its size,*
3. *$\frac{\Phi(x,\ \alpha+1)}{\Phi(x,\ \alpha)} \simeq 1/2$.*

Corollary 1 shows that the size of the nogood has a significant effect on this reduction. More importantly, Corollary 1 implies that the reduction in the size of the search space is inversely proportional to the size of the seed nogood, and in fact diminishes by 1/2 as the size of the nogood is increased by 1. This means that, for the largest reduction of the search space, it is best to have nogoods as small as possible.

We have completed a simulation of this combinatorial analysis which provides empirical confirmations to our analytic results. Namely, the % reduction is independent of n, the size of the problem, and it is most advantageous to have minimal nogoods of as small a size as possible.

3.2.2 Overlapping Nogood Analysis

We now turn to an analysis of multiple overlapping nogoods. The difficult aspect is modeling the complex interactions of the nogoods, namely taking account of the complicated manner in which overlapping of nogoods occurs when several nogoods are present; it is important not to double-count supersets of nogoods.

Lemma 3 *A problem in which overlaps of nogoods occur is convertible to one in which they do not occur.*

Lemma 3 implies that many of the results which we have obtained so far for non-overlapping problems can be used for this more complicated case. Let us now state one of the major results of [13], an upper bound on the size of the search space reduction by a set of nogoods.

Theorem 1 *An upper bound for a problem defined by the parameters* $((a,\alpha),(b,\beta),(c,\gamma),..)$, *with the nogoods overlapping randomly, is given by*

$$\Phi((a,\alpha),(b,\beta),(c,\gamma),..) \leq 2^n(a2^{-\alpha} + b2^{-\beta} + c2^{-\gamma}+....).$$

Our (worst-case) problem is still $O(2^n)$ over a wide range of nogood parameters $\left((a,\ \alpha),(b,\beta),...\right)$. It must be emphasized that the value of 2^n for $n = 100$ is 1.26×10^{30}, so even for relatively large search-space reductions, a huge amount of the search-space still remains. From the previous section, we see that nogoods cut down this number. However, any problem which forces the ATMS to construct a substantial portion of the environment lattice will cause inefficient ATMS performance.

The real problem is that ATMS interpretation construction is intrinsically NP-complete. We have just described a problem which brings out this exponential behaviour. The solution to such a combinatorial explosion of the solution space is either ensuring the constraints will generate small nogoods or carefully controlling the problem-solving. The principal aim of this latter course of action is to constrain the ATMS to look at one solution at a time, using a dependency-directed backtracking mechanism or employing consequent reasoning and stopping when a single solution is found (i.e. to revert to JTMS-style behaviour). This, however, appears to be an extreme reaction, since for problems such as this, exploring multiple solutions would be ideal.

4 Conclusions

Our two main complexity results are the following: first, problems such as the visual constraint recognition problem described here can have a very large number of solutions, and such problems are not pathological (as claimed by de Kleer in [2]) but occur naturally. To the contrary, we argue that the most challenging problems facing AI are exactly those with multiple possible solutions. Second, as again cited by deKleer [2], you do not need a problem with 2^n solutions to make the ATMS infeasibly slow. Even with a fraction of these solutions the ATMS can "blow up." This is because cases exist in which problems with a moderate number of complete solutions may have an exponential number of partial solutions, forcing the ATMS to construct an exponential number of intermediate contexts.

One important contribution of this research is the beginning of a classification of problems for which different TMSs are suited. The performance of JTMSs and ATMSs is highly problem-specific, and as yet little or no empirical or theoretical work has been done to define a better problem classification based on TMS efficiency.

There is no doubt that for moderately-sized problems there are many cases for which the ATMS is the most efficient TMS algorithm. However, for large and complex problems (e.g. vision and speech-understanding problems), this efficiency can be lost in constructing an environment lattice whose size is often exponential with respect to the database size.

Acknowledgements

I have received a great deal of comments and encouragement from Mike Brady. Many thanks to Johan de Kleer and Ken Forbus for providing me with their TMSs.

References

[1] J. Bowen and J. Mayhew. *Consistency Maintenance in the REVgraph Environment.* Technical Report AIVRU 020, University of Sheffield, 1986.

[2] J. de Kleer. An assumption-based TMS. *AI Journal*, 28:127–162, 1986.

[3] J. de Kleer. Problem solving with the ATMS. *AI Journal*, 28:197–224, 1986.

[4] J. de Kleer and J. Brown. A Qualitative Physics Based on Confluences. *AI Journal*, 24:7–83, 1984.

[5] J. de Kleer and G. Sussman. Propagation of Constraints Applied to Circuit Analysis. *Circuit Theory and Applications*, 8, 1980.

[6] J. de Kleer and B. Williams. Diagnosing Multiple Faults. *AI Journal*, 1987, to appear.

[7] M. Herman and T. Kanade. Incremental Reconstruction of 3D Scenes from Multiple, Complex Images. *AI Journal*, 30:289–341, 1986.

[8] G.E. Hinton. *Relaxation and its Role in Vision.* PhD thesis, University of Edinburgh, 1977.

[9] V.R. Lesser and L.D. Erman. A Retrospective View of the Hearsay-II Architecture. In *Proc. IJCAI*, 1977.

[10] J. Martins and S. Shapiro. Reasoning in Multiple Belief Spaces. In *Proc. IJCAI*:370–373, 1983.

[11] D. McAllester. A Widely Used Truth Maintenance System, unpublished, 1985.

[12] D. McDermott. Contexts and Data Dependencies: a Synthesis. *IEEE Trans. PAMI*, 5(3):237–246, 1983.

[13] G. Provan. Using Truth Maintenance Systems for Scene Interpretation: the Vision Constraint Recognition System (VCRS). Robotics Research Group RRG-7, Oxford University, 1987.