Intractability in the Allen & Koomen planner

Robin Hirsch*

Department of Computing,
Imperial College of Science, Technology and Medicine,
180 Queens Gate, London SW7 2BZ.

Tel: (0)71 589 5111 ext. 58209, Fax: (0)71 594 8201

e-mail: rdh@doc.ic.ac.uk

March 16, 1995

*Supported by SERC grant reference:GR/H46343

Abstract

The Allen and Koomen planner is intractable in two ways: the Allen interval algebra is an intractable temporal reasoner and the *collapsing problem* introduces a large branching factor in the search space for a solution plan. We define *independence* and *dependence* for networks to address both problems. Independence is used to find a decomposition of an interval network and dependence is used to focus search when faced with the collapsing problem.

keywords Allen interval algebra, planning, decomposition, collapsing, intractable.

1 Introduction

The Allen interval algebra has been used extensively for temporal reasoning[2] and in planning [Allen, 1983, Allen, 1984, Allen and Hayes, 1985, Villain and Kautz, 1986, Villain *et al.*, 1989, Allen and Koomen, 1983, Pelavin, 1988]. However, a number of problems have arisen. One is the incompleteness of the propagation algorithm, another is the problem of collapsing intervals in the Allen and Koomen planner - what do you do if each clause has a potential causal explanation but it is not consistent to collapse them all simultaneously? An attempt was made to deal with the intractability of the propagation algorithm [Koomen, 1989] by the use of reference intervals. A more general criterion for decomposing a network is presented here. As far as we know, no attempt has been made to deal with the latter problem (interval collapsing), and this is a serious problem for planning. In a typical planning problem, at the point when collapsing is tried there will be a large number of intervals which are to be collapsed. But, if it is inconsistent to collapse them all simultaneously, then at least one of the causal gaps cannot be removed by collapsing - an action must be introduced, but it is not known where. This could be solved by trying each one in turn - introduce a new action for a causal gap and continue with the plan algorithm. But this involves back-tracking with an enormous branching factor and completely abandons the notion of least commitment. The groundwork to deal with this problem is laid here by defining dependence and independence. This theory is developed in section 3 to solve the collapsing problem.

2 The Allen Interval Algebra

2.1 Basic Definitions and Results

- The Allen Interval Algebra $\mathcal{I}$ is defined in [Allen, 1983] and consists of all sets taken from the thirteen atomic relations $\{eq, st, end, dur, meet, ov, <, st^{\smile}, end^{\smile}, dur^{\smile}, meet^{\smile}, ov^{\smile}, >\}$ together with a composition table e.g. $ov * dur = ov \vee st \vee dur$. The composition of non-atomic relations can be calculated using the distribution of $*$ over $\vee$.

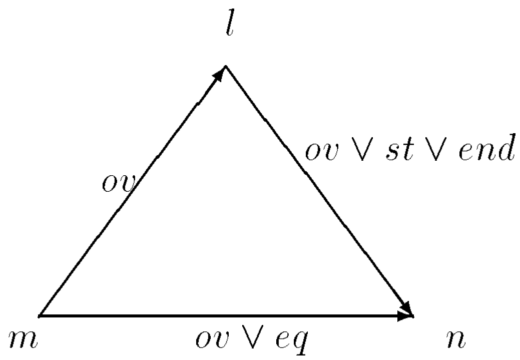
- An *interval network* N is a finite directed graph with each edge (n, m) labelled by some element of $\mathcal{I}$. We abuse the notation somewhat by letting N stand for the network itself, the set of nodes of the graph (so we can write "$n \in N$") and the labelling function (so the label on the edge (n, m) is written "$N(n, m)$").

- Let $M \subseteq N$. Define the *restricted network* $N|_M$ by $N|_M(m_1, m_2) = N(m_1, m_2)$ (for all $m_1, m_2 \in M$).

[2] The Allen interval algebra is an example of a *relation algebra*. To see how results can be obtained by looking at it in its more general, algebraic setting, try [Ladkin and Maddux, 1987].

- Up to isomorphism $\mathcal{I}$ has only one countable representation [Ladkin and Maddux, 1987] namely the set of ordered pairs of rationals $Int(\mathbb{Q})$. A network N is *consistent* if there is an embedding f from N into $Int(\mathbb{Q})$. f is called a *model* of the network N.

Example The network



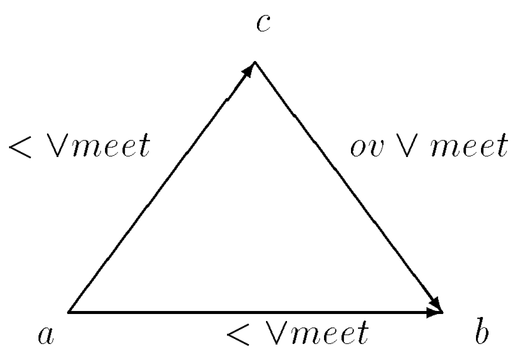
has a model in $Int(\mathbb{Q})$ given by the map

$$\begin{aligned} f : l &\rightarrow [1, 5] \\ m &\rightarrow [0, 3] \\ n &\rightarrow [1, 9] \end{aligned}$$

so this network is consistent.

- N is *complete* if for each arc (n, m) and each atomic relation $r \leq N(n, m)$ there is a model of N which realises r on (n, m).

Example The network below is not complete.



The relation 'meet' on side (a, b) cannot be realised in any model.

- N is *transitively closed*[3] if

$$(\forall l, m, n \in N)\ N(l,m); N(m,n) \geq N(l,n).$$

- The *propagation algorithm* calculates the *transitive closure* of a network, roughly, by tightening each triangle in the network and placing all triangles effected by any changes in a queue [Allen, 1983]. This algorithm has cubic complexity.

- N' is a *tightening* of N if it has the same set of nodes and for all $n, m \in N, N'(n,m) \leq N(n,m)$.

- A *labelling* of N is a transitively closed tightening N' such that $N'(n,m)$ is always an atom.

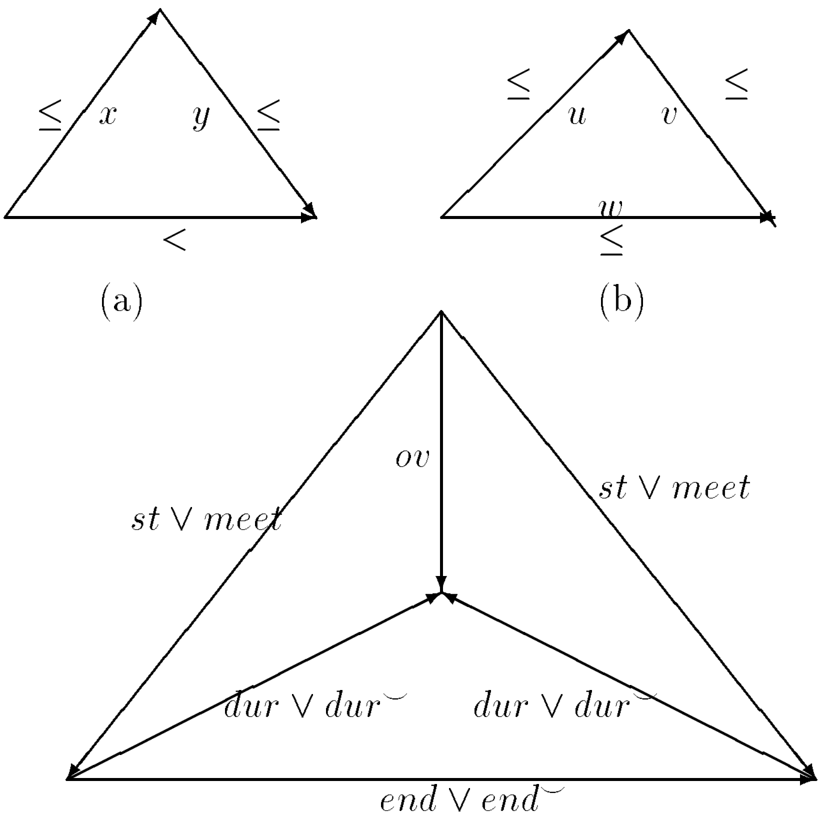
- A *point network* is just like an interval network, except the elements are taken from the *point algebra* $\mathcal{P}$ with only three atomic relations: $<, >$ and $=$. For composition we have $(< * <) = (<)$, $(< * >) = (= \vee < \vee >)$. The unique isomorphism type of a countable representation is $\mathbb{Q}$.

Note It turns out, for the Allen interval algebra and the point algebra, that every transitively closed labelling is consistent (see [Hirsch, 1994]) and therefore a network N is consistent if and only if it has a transitively closed labelling.

2.2 The Idea of Dependency

Even if a network is complete there are dependencies.

[3] Transitive closure is sometimes referred to as *path consistency*.



(a) (b)

(c)

The first two networks above are point networks (The symbol $\leq$ abbreviates $< \vee =$). Both are complete (every relation has a model) but not every combination of relations is allowed. In diagram (a) x and y cannot both be equality. In (b) you cannot have equality on w and $<$ on u (or v). The combination $(u : `='), (v : `=')$, $(w : `<')$ is not allowed either. The interval network (c) is the example taken from [Allen, 1983]. The network is transitively closed (each triangle is complete) but the network as a whole is inconsistent. There is no labelling of the network which is consistent with each triangle. Running the propagation algorithm on this network would not change the network and so the inconsistency would not be detected.

An analysis of these networks requires a way of defining the inter-dependence of the choices of relations. A formal definition of independence will be used to see if a network can be decomposed into smaller subnetworks.

A further application of the concepts of dependence and independence is to the so-called 'collapsing problem' and using independence we demonstrate an alternative to dependency directed backtracking to solve this problem.

2.3 Independence and Dependence

The following definitions and theorems apply to any relation algebra $\mathcal{A}$ such that the class of all atomic, transitively closed networks embeds into a single representation of $\mathcal{A}$. However, here we consider just two cases: the Allen interval algebra $\mathcal{I}$ and the point algebra $\mathcal{P}$.

Let N be a network (either a point or an interval network). Let $E, F \subseteq N \times N$ be sets of edges.

Definition E and F are *independent* in N, written $E \#_N F$, if for every pair of labellings L_E and L_F of E and F (respectively), consistent with N (i.e. they each extend to consistent labellings of N) there is a consistent labelling of N which extends both. In other words if L_E and L_F are each consistent with N then they are jointly consistent with N.

Examples

1. In diagram (a) on page 5 $x \not\# y$ since the labellings that, respectively, put equality on edge x and equality on edge y are individually consistent with N but they do not jointly extend to N.

2. Reference intervals.

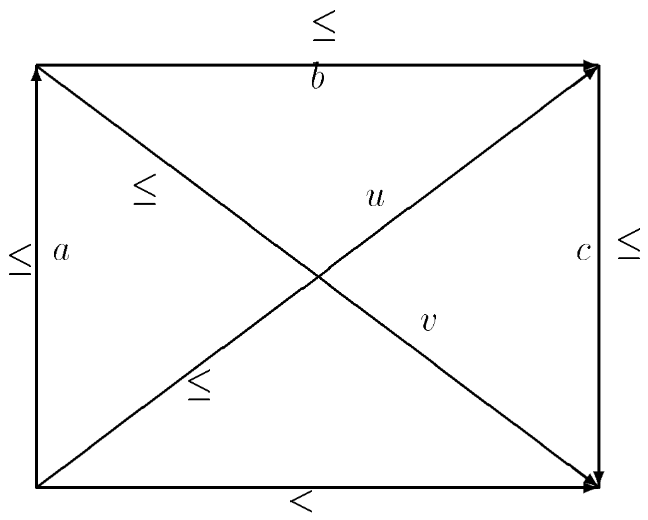
   We will show later that if i and j are disjoint *reference intervals* [Koomen, 1989] then the networks of intervals contained respectively in i and j are independent of each other.

Definition A *dependency* is a minimal inconsistent labelling. Minimal means that if the labelling is defined on a set of edges E, then any proper subset of E yields a restricted labelling which can consistently extend to N.

Examples

1. In diagram (a) on page 5, the labelling which assigns equality to x and y is a dependency.

2.



In the point network above, the following are dependencies:
$(a \rightarrow =; b \rightarrow =; c \rightarrow =)$, $(a \rightarrow =; v \rightarrow =)$, $(a \rightarrow =; b \rightarrow =; u \rightarrow <)$ and several more.

We now give the main theorem relating the concepts of dependency and independence.

THEOREM 1 *Let N be a network and $E, F \subseteq N \times N$.*

1. *$E \#_N F$ if and only if there is no dependency D contained in $E \cup F$ such that D uses at least one edge from E and at least one from F.*
2. *Let L be a labelling of some of the edges in N. L consistently extends to a labelling of N if and only if L is not an extension of some dependency.*

PROOF:

1. ($\Rightarrow$) Suppose there is a dependency D contained in $E \cup F$ and intersecting both. We must show that $E \not\#_N F$. D is minimal, so restricting D to the edges of E gives a consistent labelling i.e. one that consistently extends to N. Call this L_E and similarly define L_F. L_E and L_F are consistent labellings of E and F but with no joint extension (if there was then D would not be inconsistent).

   ($\Leftarrow$) Suppose $E \not\#_N F$ and we will show that there is a dependency D contained in $E \cup F$ and intersecting both. There are labellings L_E and L_F of E and F which are consistent with N but which do not jointly extend. So the labelling $L_E \cup L_F$ of $E \cup F$ is inconsistent. Delete edges from this labelling till you get a minimal inconsistent labelling (a dependency). This is clearly contained in $E \cup F$ but cannot be wholly contained in E (or F) since L_E (and L_F) are consistent.

2. Similar.

3. Straight-forward.

$\square$

THEOREM 2 *Let N be a network. A labelling L of N is consistent iff it does not contain a dependency of size ≤ 3 which is contained in some triangle of the network.*

PROOF:

L is transitively closed iff it is consistent iff it contains no dependencies at all. If L is inconsistent then it is not transitively closed which means that there are three elements i, j, k and $L_{\{i,j,k\}}$ is inconsistent. Therefore $L_{\{i,j,k\}}$ contains a dependency.

$\square$

2.4 A Generalisation of Reference Intervals

We wish to decompose the transitively closed network N into two (or more) independent parts E and F. That is $N \times N = E \dot{\cup} F$ and $E \#_N F$. By theorem 1 this is the case iff there is no dependency D which intersects both E and F.

Note In determining when two sets of edges E and F form such a decomposition, theorem 2 informs us that we need only check that there are no dependencies involving E and F contained in some *triangle.*

Let i and j be disjoint intervals such that every interval in the network is contained (eq, dur, st or end, written $\subseteq$) in one or other of i and j. i and j are called *reference intervals* [Koomen, 1989]. Every interval in the network is contained in exactly one of i and j.

- Suppose first that the relation between i and j is $\{<\}$. If $a \subseteq i, b \subseteq j$, then, by transitive closure, the relation between a and b can only be $\{<\}$. Claim: if L, M are respectively the network of intervals contained in i and j then $L \# M$. We prove a slightly more general result: if the network N splits into two sets of nodes A, B such that for any $a \in A, b \in B$ the relation between a and b is atomic then the networks $N|_{A\times A}$ and $N|_{B\times B}$ are independent. Note first that if a set of labelled edges from N is inconsistent with N and contains an edge from A to B then it is still inconsistent after that edge has been deleted, as that edge has only one atom on it. So any dependency of N cannot involve an edge from A to B. Suppose for contradiction that $N|_{A\times A}$ and $N|_{B\times B}$ are not independent. By theorem 1 there is a dependency D involving only edges in $A \times A$ and $B \times B$. By theorem 2 this dependency must be contained in some triangle (a, b, c). But it is not possible for such a dependency to involve no edges between A and B and have one edge from $A \times A$ and one from $B \times B$.

- If the relation between i and j is $\{<, >\}$ it can be shown that we get a decomposition into three mutually independent sets of edges: those relating two intervals contained in i; those relating two intervals contained in j and those relating an interval contained in i to an interval contained in j.

Note The previous theorems take a global view of the network. A dependency is inconsistent with the *whole* network, but for intervals determining consistency is intractable. What is needed is a *local* criterion which gives the global result. The next theorem is a local *approximation* for finding independent subnetworks.

THEOREM 3 *Let $N \times N = E \cup F$. E and F are independent in N if it is not possible to find a partial labelling L defined on two or three edges of some triangle $\tau = \{i, j, k\}$ such that*

1. *L is inconsistent* **with** *τ (no need to consider the rest of the network),*
2. *L is minimal such and*
3. *L is defined on at least one edge of E and one of F.*

PROOF:

Suppose $E \# F$. There are consistent labellings L_E and L_F of E and F, but $L = L_E \cup L_F$ is an inconsistent labelling of N, so L is inconsistent on some triangle τ. Find a minimal set of edges from τ which is inconsistent *in* τ. Clearly these edges must involve both E and F since L_E and L_F are consistent with N, so certainly consistent with τ.

□

Note To decompose a network into independent sets of edges we could define an equivalence relation $\sim$ by taking the transitive closure of the relation defined by $i \sim j$ if i, j belong to some minimal inconsistent labelling of a triangle τ as in the previous theorem.

We can now partition N into independent $\sim$-classes. Because the theorem has only a one-way implication, we may fail to get the *finest* decomposition, but we are guaranteed that the equivalence classes are independent.

3 The Allen and Koomen Planner

The Allen and Koomen planner [Allen and Koomen, 1983] is an elegant device based on the Allen interval algebra. The following description is based on the original but does contain some alterations and generalisations.

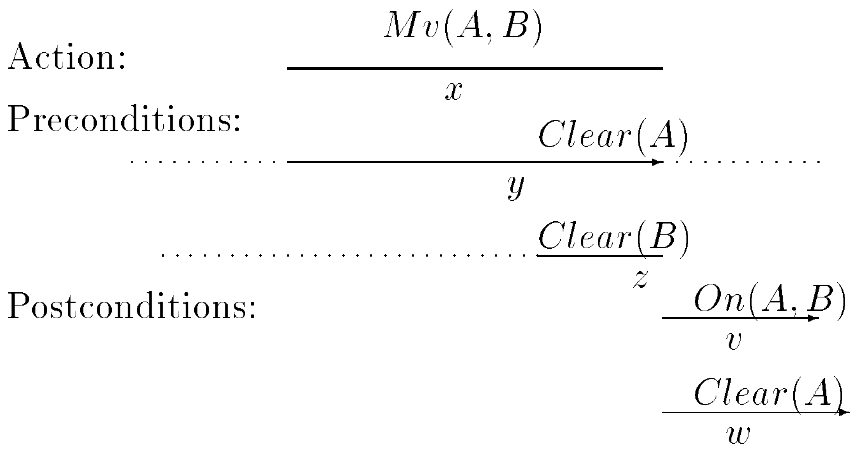
A *planning problem* is defined to be a *planning environment* (PE), a *goal state* (G) and some temporal constraints defined by an *interval network* (IN). The planning environment and goal state are sets of temporal assertions i.e. they consist of formulas of the form $p@i$ where p is a proposition[4] and i is an interval name. Here, following [Pelavin, 1988] we generalise Allen and Koomen's concepts and do not require that the initial state comprises properties true at the initial time nor that the goal state has only properties true at the final time. There may be various temporal constraints on these intervals defined in the interval network. For the present we assume that the interval network is complete i.e. each relation which occurs on an edge is realised in some model of the network.

There is also a stock of available *actions* each of which consists of an action assertion $ac@x$, effect assertions (or post-conditions) and executability conditions (or pre-conditions) together with the temporal constraints between all of these.

Example An action to move block A onto B could be defined like this:-

action	$Move(A, B)@x$
preconditions:	$Clear(A)@y$
	$Clear(B)@z$
post-conditions:	$On(B, A)@v$
	$Clear(A)@w$
temporal constraints:	$x\{dur\}y, x\{eq, end, end^{\smile}\}z, x\{meet\}v,$
	$x\{meet\}w$

Or in a diagram



Because an action may be used any number of times in a plan, the intervals in the definition of the actions should be considered as variables which get instantiated by Skolem constants if and when the action is introduced into the plan.

[4] In [Hirsch, 1991] this is generalised to allow predicates $p(x)@i$. This gain in expressive power can actually be used to enhance the efficiency of the planner. Given a range of choices for some object, the idea is to use a domain variable and instead of making a choice and back-tracking later you incrementally restrict the range of choices for that domain variable.

3.1 The Planning Algorithm

Assertions in the goal set are removed until there are none left. The resulting planning environment and the interval network constitute a *solution plan.* Goal assertions are removed by one of two methods: either an action is *introduced* which has the required assertion as one of its post-conditions or the goal assertion is *collapsed* with an assertion in the planning environment.

Introducing an action If $p@i$ is a goal assertion and the action Ac has $p@x$ as one of its post-conditions then the action may be introduced to remove the goal assertion. This is done by setting the variable x equal to the interval i and instantiating all the other interval variables in the action definition with new Skolem names. The temporal constraints between them are added and propagated through the interval network of the plan. The action assertion itself $Ac@j$ and all its post-conditions are added to the planning environment and all the pre-conditions are added to the goal set.

Collapsing If $p@i$ is in the goal set and $p@j$ is in the planning environment and the temporal constraint between i and j includes the equality relation, then we may remove $p@i$ from the goal set by collapsing i and j, that is, by asserting that $i = j$.

Notice that a goal assertion may be removed in more than one way: it may be that there is more than one interval with which it can collapse and there may be more than one action which has the desired post-condition. To clarify and simplify we introduce a definition.

Definition If $p@i$ is in the goal set and $p@j$ is in the planning environment and the temporal constraint between i and j includes the equality relation then we say that $p@j$ is a *potential causal explanation* (PCE) for $p@i$. A goal assertion with a PCE may be removed by collapsing, but a goal assertion without a PCE can only be removed by the introduction of an action.

So now we can state the planning algorithm.

1. For each goal assertion without a PCE introduce an action to remove it.
2. When every goal assertion has a PCE attempt to collapse each goal assertion with its PCE. If this produces a consistent interval network then terminate successfully.

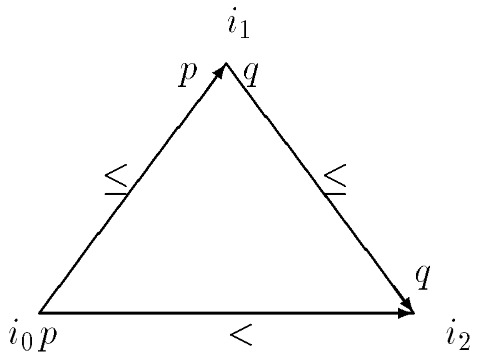
Let us simplify the problem by assuming that there is only one action which can bring about a given effect, then step 1 can be taken without the possibility of having to back-track later on.

3.2 The Collapsing Problem

The problem, which was not clear in Allen and Koomen's original, is now staring us in the face: what do we do if step 2 above fails? What do we do if each goal assertion *individually* can collapse with its PCE but collapsing them *all at once* is inconsistent? A method of tackling this will be given later on.

For the moment an exhaustive search, though wasteful, is guaranteed to find a solution plan if one exists. If it is inconsistent to collapse all the goal assertions with their PCE then at least one of them must be removed by introducing an action. So introduce an action for one of the goal assertions and then go back to the beginning of the planning algorithm. It may be necessary to back-track and choose another goal assertion so this search must be conducted breadth first. Clearly this defines an intractable procedure.

Example Using the point algebra instead of the interval algebra, let the initial planning problem be
(planning environment = $(\{p@i_0, q@i_1\}$, goal = $\{p@i_1, q@i_2\}$,
point network = $\{i_0 \le i_1 \le i_2, i_0 < i_2\})$.



Each goal has a potential causal explanation and each (individually) can be collapsed, but they cannot be collapsed together.

3.3 Solving the Collapsing Problem

Suppose we have a planning problem and every goal assertion has a PCE. We wish to collapse a (large) number of arcs, but we find it inconsistent to do so. We know that an action must be introduced for at least one arc in the causal gap but we don't know which one. Instead of trying each one in turn and backtracking we would like to either decompose the problem using the notion of independence (section 2.4) or to focus our search in some other way. We use a special version of a *dependency* which involves only the equality relation.

Folding Let N be an interval network and $U \subseteq N \times N$ be a set of arcs which we wish to collapse. Suppose this is inconsistent with N.

Definition Define a *fold* as a dependency involving only the equality relation. A fold can be represented simply as a set of edges since there is no ambiguity about the relation.

Examples

- In the diagram above there is only one fold - $\{(i_0, i_1), (i_1, i_2)\}$.
- In diagram 2 on page 7 there are four folds: $\{a, b, c\}, \{a, v\}, \{u, c\}, \{u, v\}$.

3.4 Using folds for collapsing

Idea: suppose we wish to collapse 100 arcs but arc47 and arc83 cannot collapse simultaneously. We know that an action must be introduced somewhere and a fold can help focus our attention. Our fold is {arc47, arc83} and this tells us that an action must be introduced for one of these two intervals. We may still have to backtrack, but only once instead of 100 times. In this way the principle of least commitment is preserved as far as possible.

Modified planning algorithm We have a planning problem (G, PE, IN) where G is a set of goal clauses, PE is the planning environment and IN is the interval network. A goal clause $p@i$ has a potential causal explanation if there is an assertion $p@j$ in the planning environment with the same property (p) such that the set of relations in the interval network between i and j includes equality. We will refer to the *causal explanation* of $p@i$ by the arc (j, i).

1. Continue to introduce actions for goal clauses without PCEs until every goal clause has a PCE.
2. Let $U = \{\text{causal explanations } (j_g, i_g)\}_{g \in G}$. If there are no folds then U is consistent so collapse all the arcs and terminate successfully. Otherwise, find a fold in U (a tractable algorithm which finds this is described later).
3. Back-tracking point. Choose one arc from the fold and introduce an action for it. Go back to step 1

Note : if the fold is of size one (only one arc) then equality is not consistent on that arc. This is not a genuine causal explanation and an action must be introduced. If the propagation algorithm were complete it would have been deleted.

Question : Is it possible, in a transitively closed interval network, that the size of the smallest fold can exceed two? (Answer: probably yes, though examples of networks with a fold of size three of more have not been discovered by the author).

3.5 Algorithm to calculate folds

We have an interval network N and a set of arcs
$U \subseteq N \times N$. Let $U = \{(j_k, i_k) : k = 1 \ldots n\}$.

1. Base case. If N is already inconsistent then $\emptyset$ is a fold.
   Else,
2. Propagate $j_1 = i_1$ in N to get N^*.
3. Recursively, take N^* and find a fold D^* in N^* contained in $U \setminus \{(j_1, i_1)\}$.
4. (a) If collapsing D^* is inconsistent with N then D^* is a fold in N.
   (b) If collapsing D^* is consistent with N then $\{(j_1, i_1)\} \cup D^*$ is a fold in N.

Note :

- this algorithm has linear complexity, over the complexity of propagation. If we use a polynomial but incomplete propagation algorithm (like the Allen propagation algorithm) we will still produce an inconsistent set of collapsing arcs, but it may not be minimal. This may mean that we have slightly more back-tracking than needed. It can still be very efficient at focusing the search space to a small set of choices.
- to *minimise* back-tracking we require not just *any* fold (minimal inconsistent collapsing set of arcs) but a fold of smallest size (global minimum). Global minima are always harder to find than local minima, but if we assume that the size of a smallest fold is bound by a relatively small integer B, say, our search space will be $O(n^B)$. Simply try to collapse all arcs singly, then try all pairs, then all triples and so on. We must find an inconsistent set by the time we check all sets of size B (of which there are ${}^nC_B = O(n^B)$). The first set we find to be inconsistent must be a global minimum.

Is it reasonable to assume that the size of the smallest fold is small? In the Allen and Koomen planner we search for goal clauses with no potential causal explanation and introduce actions for them. In our terminology, we find a fold of size one. Having dealt with all these they assume that it is consistent to collapse all the goal clauses. In other words, the size of the smallest fold is assumed to be one. Now, we know this isn't always the case. But it does seem fair to relax their assumption but keep the size of the smallest fold quite small (perhaps two or three).

So, in the planning algorithm (above) at step 2 we should look first for a fold of size 1, then 2 and so on till we find an inconsistent set of collapsing arcs which will be not only a fold, but a global minimum.

4 Conclusions

The reader may wonder whether the collapsing problem is not just an awkward problem that can be avoided by changing the planning architecture. However, comparison with a number of alternatives indicates that this is a more profound problem than that. In the Time Map Management (TMM) system of [Dean and McDermott, 1987] the same problem occurs here as an intractability involved with stretching and clipping (read [Materne and Hertzberg, 1991] for an account). In that system a property p is asserted to hold for a certain interval but there is uncertainty about the limits of the interval. In order to obtain a solution to a set of temporal constraints it is permissible to *stretch* the interval, which means that p is assigned an interval that is *consistent* with the constraints, but not necessarily *implied* by them. This is a non-monotonic step and is equivalent to the collapsing of maximal intervals in the Allen and Koomen set-up.

The event calculus [Kowalski and Sergot, 1986] also faces this difficulty once it is made to handle interval reasoning. Here, the property p is assumed to hold at time t if some event earlier than t initiated p and nothing terminated p in between. This is the so-called *persistence axiom* which says that in the absence of any evidence to the contrary we can assume that p stays true once it is made true, provided nothing stops p being true. The problem is that further iterations of the planning algorithm may indeed force that p is made false. The use of the persistence axiom is equivalent to collapsing maximal intervals.

At its most general level this can be seen to be a manifestation of the persistence problem which is, itself, an aspect of the frame problem. If p is known to be true at time $t = 0$ and we require that p should be true at the goal state $t = 10$ then we can solve the planning problem by either saying that the maximal interval for the goal assertion and for the initial state are the same interval (collapsing) or, equivalently, by saying that p persists until the goal time. Thus the solution to the collapsing problem described herein, that maintains tractability by accepting only *approximate* solutions, should provide quite a powerful tool for a range of temporal reasoners and planners.

References

[Allen and Hayes, 1985] J F Allen and P J Hayes. Moments and points in an interval-based temporal logic. Technical Report TR180, Department of Computer Science, University of Rochester, 1985.

[Allen and Koomen, 1983] J Allen and J Koomen. Planning using a temporal world model. In *Proc 8th IJCAI*, 1983.

[Allen, 1983] J F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[Allen, 1984] J F Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[Dean and McDermott, 1987] T Dean and D McDermott. Temporal database management. *AI*, 32:1–55, 1987.

[Hirsch, 1991] R Hirsch. Some comments on allen and koomen's planner. Technical report, Imperial College Planning Project, 1991.

[Hirsch, 1994] R Hirsch. *Relation Algebras of Intervals*. PhD thesis, Imperial College, 1994.

[Koomen, 1989] J Koomen. Localising temporal constraint propagation. *Knowledge Representation*, 1989.

[Kowalski and Sergot, 1986] R Kowalski and M Sergot. A logic-based calculus of events. *New Generation Computing (Japan)*, 4, 1986.

[Ladkin and Maddux, 1987] P Ladkin and R Maddux. The algebra of convex time intervals. Technical report, The Kestrel Institute, Palo Alto, California, 1987.

[Materne and Hertzberg, 1991] S Materne and J Hertzberg. MTMM - correcting and extending time map management. In *Proceedings of the European Workshop on Planning*, March 1991.

[Pelavin, 1988] R Pelavin. *A formal approach to planning with concurrent actions and external events*. PhD thesis, University of Rochester, 1988.

[Villain and Kautz, 1986] M Villain and H Kautz. Constraint propagation algorithms for temporal reasoning. *Knowledge Representation*, pages 377–382, 1986.

[Villain *et al.*, 1989] M Villain, H Kautz, and P van Beek. Constraint propagation algorithms for temporal reasoning - a revised report. *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381, 1989.