Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 79–89. Printed in the USA.
Copyright © 1999 Cambridge University Press 0890-0604/99 $12.50

Combining heterogeneous temporal information: A case study

SILVANA BADALONI

Department of Electronics and Computer Science, Via Gradenigo 6A, 35100 Padova, Italy

(Received May 2, 1998; Revised June 2, 1998; Accepted October 27, 1998)

Abstract

In this paper we present a specialized system for temporal reasoning representing possibly uncertain qualitative and quantitative constraints relative to temporal intervals in a homogeneous framework. The underlying logic is that of Allen, which we have extended to include quantitative temporal information. Intervals are still assumed as primitive entities while, to represent metric information, time points are introduced only at the algorithmic level. We illustrate the properties of the temporal representation, the algorithms used for constraint propagation, and their complexity. We show how the system can be used to schedule manufacturing operations in a small factory.

Keywords: Temporal Reasoning; Constraints; Task Scheduling

1. INTRODUCTION

Representing and reasoning about temporal relationships of activities and events is an essential task in many advanced industrial applications. Automatic systems based on logical models that explicitly represent time seem to be successful because they allow us to deal with activity durations, possible concurrency of activities, partial or total deadlines and, generally, allow us to represent the dynamics governing the environment.

A scheduling problem can be typically represented in terms of variables and constraints and their values. A schedule is then represented as an assignment of values to all variables that satisfy all the constraints. The resulting formulation of a scheduling problem is called a *constraint satisfaction problem* (Tsang, 1993; Zweben & Fox, 1994; Dean & Kambhampati, 1995). Constraints are interpreted in a temporal domain to develop a temporal reasoning component that is useful across different applications, such as planning and scheduling but also molecular biology, as Golumbic and Shamir (1993) have shown.

Due to its good trade-off between expressiveness and efficiency, one of the most influential works as regards temporal reasoning is the interval-based framework of Allen (1983, 1984). Intervals are assumed as primitives according to the idea that they are the natural means of human reference to time. Temporal knowledge is represented in terms of qualitative relations between any two intervals (e.g., interval *A meets* or *overlaps* interval *B*). However, in many real applications, there is also a need to represent the metric information on intervals, for example, their duration.

Assuming time intervals as primitives, two types of temporal information have to be considered:

1. qualitative information relative to the mutual position of intervals; and
2. quantitative information relative to the duration of intervals and the distances between the extremes of pairs of intervals.

Furthermore, the available information is often uncertain and coarse, so that it can be represented as the disjunction of possible relationships that can hold between two intervals, without specifying which of them actually holds or the exact position in time of the intervals.

The aim of this paper is to provide a new subalgebra, Interval Distance Sub-Algebra (IDSA), for combining metric and qualitative (possibly uncertain) temporal information relative to intervals.

The representation is restricted to pointizable subalgebra of intervals (van Beek, 1990, 1992; Nebel & Burckert, 1995)

Reprint requests: Silvana Badaloni, Department of Electronics and Computer Science, Via Gradenigo 6A, 35100 Padova, Italy. E-mail: badaloni@ladseb.pd.cnr.it

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

as regards the qualitative relations, while the metric information is expressed as a 4-tuple of distances between end points of intervals. It is then combined with the qualitative one at the algorithmic level.

We assume intervals as primitives at input/output level and we introduce time points only to deal with metrics at the algorithmic level; so, time points are relegated to a secondary status. This assumption is supported by our experience of applications in robotics (Badaloni et al., 1995) on the basis of which we claim that temporal knowledge is naturally associated to intervals rather than to time points.

In the first part of the paper, we describe the temporal reasoner system based on IDSA subalgebra and we show its properties, the algorithms used for constraint propagation and compare it with other approaches proposed to combine metric and qualitative temporal information (Kautz & Ladkin, 1991; Meiri, 1991; Becker, 1995).

In the second part of the paper we address the problem of scheduling a set of manufacturing operations in a small factory by using the temporal reasoning system. We believe that many interesting robotic manufacturing processes fall into the class represented by this simple scenario where a set of operations has to be accomplished within a given time.

The paper is organized as follows: In Sections 2 and 3 we introduce the temporal representation; in Section 4 we provide a formal description. The relation between qualitative and metric relationships is briefly analyzed in Section 5 and some properties of IDSA subalgebra are shown in Section 6. In Section 7 we examine the algorithms used and in Section 8 we show a practical application of a reasoning task.

2. TEMPORAL ENTITIES

The underlying logic is Allen's. Temporal entities are properties and occurrences such as events and processes. We briefly recall here their definitions with some comments (Allen, 1983, 1984; Allen & Ferguson, 1994).

Properties represent the static aspect of the world. A *property* can be defined according to:

$$holds(P,I) \leftrightarrow (\forall t.in(t,I) \rightarrow holds(P,t)),$$

where the temporal relationship *in* is:

$$in = starts \vee during \vee equal \vee finishes.$$

If a property P holds over an interval I, it holds over all subintervals of I.

Processes describe activities that do not have a precise ending point (or accomplishment). For example "I am running" is a process. Processes are recursively defined in terms of themselves in Allen (1984), and, as noticed by Trudel (1993), it is clear that their formal definition is rather controversial. Galton (1990) argues that it is possible to simplify the ontology by subsuming processes under properties. As a final remark, we could say that processes have been quite obscure objects from the beginning and rather useless in practical applications.

Events describe some dynamic aspects of the world:

$$(occurs(P,I) \leftrightarrow ((occurs(P,I') \wedge (\forall t \cdot in(t,I')$$

$$\rightarrow \neg occur(P,t))))).$$

If an event P occurs over an interval I then it does not occur over any proper subintervals of I. This property represents the fact that an event is *gestalt* (Shoham, 1987).

So, in our approach, temporal facts are essentially properties and events referring to intervals. We can say nothing regarding the truth value of logical formulae that refer to time points.

As we have already argued, time points are relegated to a secondary status within our representation and are used exclusively to express the mutual position of intervals in terms of distances between end points. We have excluded points from ontology because they are mainly rather problematic (Ma & Knight, 1994; Knight & Ma, 1994). On the other hand, excluding points makes it impossible to deal with continuous change (Galton, 1990; Knight & Ma, 1994; Ma & Knight, 1994). But, in the applications of planning and scheduling which concern us, it is not essential to describe continuous changes. If it were the case, we would use the revised theory of time proposed by Ma, Knight, and Petridis (1994).

3. QUALITATIVE AND METRIC TEMPORAL INFORMATION

The temporal reasoner deals with possibly uncertain qualitative and quantitative constraints relative to temporal intervals.

Two main commitments are:

1. to blend metric information and qualitative information in a representation based on intervals as elementary entities; and
2. to represent possibly uncertain knowledge by utilizing disjunction of durations of intervals or disjunction of qualitative relations between intervals.

The notion of disjunction is also needed to represent mutually exclusive events. As an example, suppose that orders of *event1* and *event2* are exclusive. Then *event1* either follows or precedes *event2*. This is a common constraint in planning and scheduling problems. This type of constraint cannot be expressed in point based logic (van Beek & Cohen, 1990).

On the other hand, interval algebras such as Interval Algebra (IA) (Allen, 1983) and the pointizable subalgebra of IA (SAC) (van Beck & Cohen, 1990; van Beek, 1992; Nebel & Buckert, 1995) deal only with qualitative information while Distance Algebra (DA) (Dechter, Meiri, & Pearl, 1991)

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

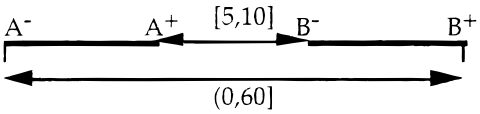


Fig. 1. Example of initial knowledge.

deals exclusively with metric knowledge in a time point representation.

Thus, we propose a new integration of qualitative and quantitative information referring to interval-based constraint network.

Before we present formal definitions, in this section let us give an example to outline the intuition. Suppose we know that an interval A precedes an interval B by 5–10 min and that the maximum duration from the beginning of A until the end of B must not exceed an hour (Fig. 1).

The qualitative relationship between A and B can be expressed by the relation *before* $\in$ IA but, as for the metric information, we have to consider the end points of intervals:

$$A = \{A^-, A^+\}, B = \{B^-, B^+\}$$

and their relative distances. Such distances can vary from a lower bound to an upper bound; unknown distances can be assumed to vary in the range $(-\infty, +\infty)$. In the present case, the input knowledge can be expressed as:

$$R_{input} = (before, (-\infty, +\infty), (0,60], [5,10], (-\infty, +\infty)),$$

where *before* describes in a qualitative way the mutual position of the two intervals and the four pairs of data represent the ranges of variability of the four distances (in the order):

$$D_{11} = B^- - A^-$$

$$D_{12} = B^+ - A^-$$

$$D_{21} = B^- - A^+$$

$$D_{22} = B^+ - A^+$$

with the intended meaning that D_{11} is the duration of A plus the distance between A and B, and so on for the other D_{ij}. As examples, $B^- - A^+ = [5,10]$ means that $5 \leq B^- - A^+ \leq 10$ and $B^+ - A^- = (0,60]$ means that $0 < B^+ - A^- \leq 60$.

Now, to blend the two types of information and take into account some further information relative to the intervals' durations (suppose that A has a duration ranging from 1 to 5—[1,5]—and B from 10 to 50—[10,50]), it is necessary:

- to transform the qualitative relation *before* into the corresponding metric one, that is in the 4-tuple:

$$((0,+\infty),(0,+\infty),(0,+\infty),(0,+\infty))$$

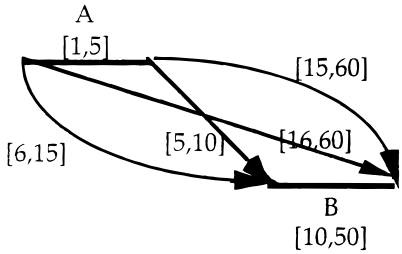


Fig. 2. Example of the point-based network.

- to perform the intersection between this 4-tuple representing the qualitative relation and the metric one;
- to consider the point-based network having the interval extremes as nodes and the elements of the 4-tuple and interval durations as arc labels; and
- to check consistency and compute the minimal network in order to find the tightest constraints over the point network.

In the example, we obtain the point-based minimal network of Figure 2 and the relation becomes:

$$([6,15],[16,60],[5,10],[15,60]).$$

Computation is carried out on metric information in a point-based network; but the results obtained from computation are reconverted into an interval-based framework to be given as output.

Then the last steps are:

- to translate the point-based network into the corresponding interval-based
- to express the qualitative part of the relationship explicitly.

In this case, the relation is:

$$R_{output} = (before, [6,15], [16,60], [5,10], [15,60]).$$

Then, at the inner level, where propagation algorithms are applied, only metric constraints are represented, in the form of distances between ending points of intervals. However, at the interface level, in input and in output, this information is split into its qualitative and quantitative components.

4. DEFINITIONS

Let us now formally define **unary** and **binary constraints** relative to a temporal network, whose nodes are labelled by temporal intervals, whose directed arcs by unary and binary temporal constraints.

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

DEFINITION. A **unary constraint** UC_i is defined as the disjunction of possible durations for an interval:

$$UC_i = D_1 \vee D_2 \vee \ldots \vee D_n,$$

where D_i represents the range of values of the interval duration, in the form $d(L,U)$, where $L \in \Re$, $U \in \Re$ and $L \leq U$. ■

As an example, if an interval has a duration included between 5 and 10 (w.r.t. a given timeline), the unary constraint is expressed as $d(5,10)$, while if it lasts 5 then the constraint becomes $d(5,5)$. The form $d(5,10)$ is equivalent to $[5,10]$ throughout the text.

DEFINITION. A **binary constraint** BC_i between two intervals A and B is defined as a disjunction of binary relations that can hold between those intervals:

$$BC_i = R_1 \vee R_2 \vee \ldots \vee R_n,$$

where $R_i \in$ IDSA (Interval Distance Sub-Algebra), for $i = 1..n$. ■

The binary relation R_i between A and B is an ordered tuple of four intervals belonging to the real axis. It defines the mutual position of the intervals A and B, blending together the quantitative and qualitative temporal information in the following way. Let A and B be two intervals and let us consider their extremes:

$$A = \{A^-, A^+\}, B = \{B^-, B^+\}.$$

DEFINITION. Any two intervals A and B satisfy a binary relation R_i if the distances D_{11}, D_{12}, D_{21}, and D_{22} belong to the four intervals composing the 4-tuple R_i. The distances D_{11}, D_{12}, D_{21}, and D_{22} (open, closed or semi-open) represent the durations of:

$$D_{11} = B^- - A^-$$

$$D_{12} = B^+ - A^-$$

$$D_{21} = B^- - A^+$$

$$D_{22} = B^+ - A^+$$

Because uncertainty is allowed, each distance may have a range of variability defined by a lower and an upper bound. It follows that $D_{ij} = d_{ij}(L_{ij}, U_{ij})$, where L_{ij} and U_{ij} are the lower and the upper bounds respectively, for $i,j = 1,2$; obviously, if the distance is exactly known then L_{ij} and U_{ij} coincide. We use the notation $o(L_{ij})$ or $o(U_{ij})$ to indicate that $D_{ij} > L_{ij}$ or $D_{ij} < U_{ij}$, respectively. ■

All IDSA relations R_j in a binary constraint BC_i are pairwise not-unitable (*canonical form)*: if two IDSA relations are unitable, as they are processed, they are expressed as a single IDSA relation. This is an important step to keep the number of disjunctions for each constraint as small as possible, so as to reduce computational effort when performing operations on the network. The time needed to propagate information over the network, to infer new constraints, is closely dependent on the number of relations per arc.

DEFINITION A binary constraint BC_i is in *canonical form*, if it is the disjunction of IDSA relations with intersections mutually empty, that is, if the intersection of any two relations R_j and R_k composing the constraint, is empty. ■

As we have previously seen, the input knowledge concerning any pair of intervals can be represented as a relation R_{input}:

$$R_{input} = (RQ, D^m_{11}, D^m_{12}, D^m_{21}, D^m_{22}),$$

where the index m indicates metric information only.

Thus, one can specify either only the qualitative relation RQ, or only some distances between the four end points or, finally, RQ plus some distances.

The qualitative relation $RQ \in$ SAC, where SAC is the pointizable subalgebra of Interval Algebra (van Beck & Cohen, 1990; vanBeek, 1992; Nebel & Buckert, 1995). Obviously, SAC contains the 13 atomic relations of IA, and it is composed by those relations of the interval algebra that can be translated into a conjunction of relations of PAC (continuous point algebra) between the beginning and the ending points of the intervals; for example, *before* $\vee$ *overlaps* is allowed, but *before* $\vee$ *after* is not. To this set (SAC) belong the relations proposed by (Freksa, 1992), where Allen's 13 mutually exclusive relations are grouped to process directly coarse temporal knowledge (e.g., *precedes* represent *before* $\vee$ *meets)*.

To use SAC subalgebra, instead of IA, to express qualitative relations between intervals, means to compact the knowledge associated with each arc, thus reducing considerably the computational effort.

The input relation is translated into an IDSA relation according to the canonical form definition. Of course, to identify an IDSA relation, the 4-tuple of distances must be consistent. To check the consistency of a single IDSA relation, the system follows these steps:

1. the four distances composing the 4-tuple are represented over a network whose nodes are points (the extremes of the intervals) and whose arcs are labelled by the distances specified in the tuple, thus obtaining a Distance Algebra network. If this network is consistent, the tuple is consistent, too, and it represents an IDSA relation between two intervals;
2. a minimal network algorithm (Mackworth, 1992) is then applied to this point-based network to compute the tightest constraints over the point network: a new 4-tuple is obtained, which is equivalent to the initial one (i.e., represents the same set of solutions for the point network).

This corresponds to determine the canonical form of a binary relation: let us say, the minimal 4-tuple.

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

The results obtained by applying the constraint propagation algorithms are presented in output utilizing the same input format, that is, specifying explicitly the qualitative relation *RQ* and the four distances that compose the quantitative relation between intervals. Note that the qualitative relation may be redundant, but it is convenient, for many practical purposes, to render it explicit. In this way, there is an input/output convenience: at the user level temporal knowledge is expressed in the same way.

The fact of considering intervals at the user level and time points at the computational one is indeed close to the representation presented in IxTeT (Ghallab & Laruelle, 1996). However, we consider intervals as primitives and the formalism IDSA has been defined to blend metric information relative to intervals with qualitative information homogeneously.

5. THE RELATIONSHIP BETWEEN *RQ* AND THE METRIC ONE

Let us analyze the relationship between the qualitative component *RQ* and the metric component, let us call it *RM*, in a relation R_i between two intervals *A* and *B*. What we want to show is that *RQ* can be expressed as a quantitative relation *RM* (4-tuple of four distances between the end-ponits of the two intervals), and, vice-versa, every *RM* relation identifies a *RQ* relation.

Qualitative → quantitative

Each relation belonging to SAC ($RQ \in$ SAC) is completely specified by means of PAC relations ($<,\leq,>,\geq,=$) and can therefore be represented in the DA algebra by a set of distances between the end points of the two intervals *A* and *B*. Hence, there is a mapping between each SAC relation and a 4-tuple of distances, which is a quantitative relation *RM* belonging to IDSA.

Quantitative → qualitative

The four distances composing a 4-tuple can be transformed into qualitative constraints between points according to the Table 1 (the symbol ? represents the universal relation).

The conjunction of these qualitative relations between the end points of the intervals *A* and *B* identifies a relation belonging to the SAC.

We can conclude that it is always possible to pass from an interval-based representation to a point-based representation without loss of information.

6. PROPERTIES OF IDSA

As previously explained, a relation of IDSA is a tuple of four distances between the extremes of two intervals:

$$R_i = (D_{11}, D_{12}, D_{21}, D_{22}),$$

where $D_{ij} = d_{ij}(L_{ij}, U_{ij})$, for $i, j = 1,2$.

Table 1. *Relation between quantitative and qualitative constraint*

Quantitative constraint (DA)	Intermediary quantitative constraint	Qualitative constraint (PAC)
[a,b]	$(0,+\infty)$	$<$
(0,b]	$(0,+\infty)$	$<$
[0,b]	$[0,+\infty)$	$\leq$
[−a,b]	$(-\infty,+\infty)$	?
[−a,−b]	$(-\infty,0)$	$>$
[−a,0)	$(-\infty,0)$	$>$
[−a,0]	$(-\infty,0]$	$\geq$
[0,0]	[0,0]	$=$

A tuple is inconsistent when it implies a null or negative duration for the constrained intervals. For example, the tuple:

$$(d_{11}(5,10), d_{12}(3,4), d_{21}(-\infty,+\infty), d_{22}(-\infty,+\infty))$$

is inconsistent.

The empty relation, ϵ, is represented by the set of all the inconsistent tuples.

The universal relation is given by:

$$\Phi = (d(-\infty,+\infty), d(-\infty,+\infty), d(-\infty,+\infty), d(-\infty,+\infty)).$$

The identity relation is:

$$I = (d_{11}(0,0), d_{12}(o(0),+\infty), d_{21}(-\infty,o(0)), d_{22}(0,0)),$$

where $o(0)$ means that the extreme is open.

In general, the union of two IDSA relations is not an IDSA relation. Hence, IDSA is a subalgebra, not an algebra, because it is closed under operations of intersection and composition, contains the empty relation, the universal relation, and the identity relation, but it is not closed under union.

7. POSSIBLE QUERIES

7.1. Consistency

A temporal network is consistent if there is a consistent labelling, that is, if at least one singleton labelling exists that satisfies all the constraints. A singleton labelling is obtained by considering only one relation for each binary constraint and only one duration for each interval. It is consistent if there is a possible assignment of the position of the intervals onto the time line, such that all the constraints forming the labelling are satisfied.

For each singleton labelling, the network can be translated into a network over time points (extremes of intervals), that is, an STP (Simple Temporal Constraint Problem) (Dechter et al., 1991). The network is consistent if and only if a consistent STP exists. Checking consistency requires an

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

algorithm complexity $O(n^3)$, where n is the number of nodes in the STP (path-consistency) (Tsang, 1993, van Beek & Manchak, 1996).

The problem of checking the consistency of an IDSA network is an NP-hard problem: in the worst case it is necessary to check the consistency of all singleton labellings, to conclude that the network is inconsistent. If e is the number of edges on the graph, and k is the maximum number of IDSA relations per arc, k^e is the maximum number of singleton labellings to be checked. For each singleton labelling, a path-consistency algorithm is applied, so that the complexity of the whole problem, in the worst case, is $O(n^3k^e)$. Obviously, the lower the number of relations per arc, the lower the number of singleton labellings k^e. Thus, we can understand the reason why it is important to maintain the knowledge as "compact" as possible, that is to represent a binary constraint by means of a small number of disjunctive relations, as previously mentioned.

7.2. Finding a solution

The problem of finding a solution for a given network can be dealt with in different ways. One can consider a "solution" as a consistent singleton labelling, for example. But even within a singleton labelling there is some uncertain information: if we say that an interval can assume only one possible duration selected from a disjunction of durations, we must remember that this duration itself is affected by uncertainty, because it is expressed as $d(L, U)$, which means that the interval duration can vary from L to U. Then, as a possible solution of a network, we intend the set of exact values that the extremes of intervals can assume. We refer to a given time line, which is defined as a particular interval called I_0, such that all intervals of the network are contained within I_0. If we assign the value zero to the starting point of I_0, when a singleton labelling is fixed, a range of variability for each extreme of each interval can be found, and a precise value belonging to this range can be imposed for each point. Thus, finding a solution of a given network means to assign to each extreme of each interval a precise value within the time line, in such a way that all temporal constraints are satisfied: in other words, finding a scenario consistent with the temporal information.

7.3. Minimal network

The minimal network is a network equivalent to the initial one, but having the tightest temporal constraints (i.e., all inconsistent relations have been eliminated). The algorithm used is:

1. search for all consistent singleton labellings;
2. for each singleton labelling:
   (a) translate it into an STP, whose nodes represent the interval extremes and whose arcs are labelled with the range of variability of the distances between nodes;
   (b) apply path consistency over the STP to obtain its minimal network;
   (c) translate the time points network into the equivalent interval network; and
3. the minimal network is obtained as the union of all these networks.

The algorithm complexity is still $O(n^3k^e)$. The time complexity is equivalent to the ones defined for the worst case of the problems of checking consistency and finding a solution seen above. In this case, all singleton labellings must be examined to find the minimal network. Thus, the time needed is always greater than that needed to check consistency or find a solution. All of the algorithms above have been implemented by using the CLP(R) language (Jaffar & Lassez, 1987, Jaffar et al., 1992).

8. OTHER APPROACHES

Comparing IDSA representation with IA algebra with regard to their expressive power, we can easily see that an IDSA network is as expressive as an IA network, if its use is limited to represent only qualitative information.

Comparing IDSA with DA, we can say that, if each point is viewed as an extreme of an interval, all relations that can be expressed within a TCSP can also be expressed within an IDSA network, while the contrary is not true because in a TCSP metric relations between intervals cannot be represented.

We can therefore conclude that an IDSA network has a greater expressive power than IA and DA representations.

Other approaches to temporal representation and reasoning allow us to represent explicitly metric information and to combine it with the qualitative information. In Kautz and Ladkin (1991), the information is represented in two separate networks: the first with qualitative information, whose nodes represent intervals, and the second with metric information, whose nodes represent ending and starting points of intervals. Maintaining information in this way is quite expensive as it is necessary to propagate the constraints imposed by one network on the other one in an iterative way. The algorithm terminates in $O((n^2(m + n^3))$ time, where n is the number of intervals in the network resulting from the union of the two networks and m is the time required to compute the minimal network of the qualitative network (which is itself a NP-complete problem). Within this framework it is possible to represent single time points, that is, time points that are not the extremes of an interval. This is not allowed in our representation. Apart from this, our representation has a greater expressive power than Kautz and Ladkin's. The sublanguage L_a (only qualitative) can be represented in an IDSA network, because all IA relations can be expressed in our formalism. Moreover, simple metric constraints between end points can surely be represented in an IDSA network, because distances between end points can be specified within an IDSA relation. The contrary is not

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

true: an IDSA network cannot be translated into Kautz and Ladkin's formalism because in their formalism a disjunction of metric relations between intervals cannot be represented, neither can a disjunction of metric relations between a pair of points be represented. Only one singleton labelling of an IDSA network can be expressed within this formalism, because it is equivalent to an STP (thus, only the sublanguage L_m is sufficient, in this case).

In Meiri (1991), only one network with two kinds of nodes is used: one representing intervals, the other temporal points. Constraints between intervals consist of Allen's disjunctive qualitative relations, while constraints between points consist of the set ($<$, $=$, $>$) of qualitative relations or metric relations (ranges of variability of the temporal distances). Meiri also defines qualitative relations (*before, starts, during, finishes, after)* between a point and an interval. This approach allows the direct representation of qualitative and metric information, but it forces us to consider three nodes for each interval in the network: one representing the interval, and two representing its extremes.

Our representation can be considered hybrid in a different way: at the inner level, where propagation algorithms are applied, only metric constraints are represented, in the form of distances between ending points of intervals. However, at the interface level, both in input and output, this information is split into the qualitative and metric forms. This is because it may be useful for the user, in input, to represent only the partial knowledge about temporal relations at hand (e.g., that something is before something else) and, in output, to easily interpret and use the qualitative constraints between intervals. In this way, the expressiveness of the system is kept homogeneous.

The complexity for the consistency check and for the computation of minimal network is exactly the same for our formalism and Meiri's approach (NP-hard); in fact, in both cases, the network is decomposed into singleton labellings, which are transformed into an STP. It is worth noting, however, that in our framework the number of singleton labellings to be examined is always less than or equal to the number of singleton labellings of a Meiri's network: indeed, each IDSA relation (the disjunction of IDSA relations forms a constraint on an IDSA network) may represent the union of two or more relations that are expressed as a disjunction on a Meiri network. IDSA constraints are maintained in a form such that IDSA relations are pairwise non-unitable. For example, given the constraint between intervals *P* and *Q*:

$$before \vee during \vee overlaps \vee meets \vee starts$$

in a Meiri network, it generates five singleton labelling (one for each basic relation). It is translated into our formalism as the single SAC relation *sb* (*survivedby)*, which generates only one singleton labelling. This feature is very important because the computation time depends greatly on the number of labellings to examine. Within each labelling, the computation time (to check consistency and calculate the minimal network) is polynomial.

In Badaloni et al. (1996), where a reduced version of the present work was published, we showed how a typical temporal problem proposed by Meiri can be solved using our approach.

A study proposed by Becker (1995) deals with a hybrid formalism to treat both qualitative and quantitative temporal and nontemporal constraints. The aim is to combine Allen's and Rit's formalisms (Rit, 1986) and nets of arbitrary constraints. It seems to us that because events in Allen's and Rit's approaches are quite heterogenous entities from an ontological point of view, their unification in a framework might be problematic. Instead, in our formalism, we extend the Allen temporal logic only by including, let us say, a measure over intervals. We preserve the ontology concerning the temporally qualified facts (i.e., properties, events, and processes). Moreover, bearing in mind that an SOPO (events are defined as sets of possible occurrences) consists of six components, stating a minimal (−) and a maximal (+) value for start (S), duration (D) and finish (F) (Rit, 1986), and that this information has to be integrated with other information of a different type (qualitative, nontemporal) (Becker, 1995), the propagation algorithm complexity seems to increase rapidly. However, a detailed analysis of complexity is lacking in (Becker, 1995).

9. SCHEDULING OF MANUFACTURING OPERATIONS

Let us now consider an example regarding the scheduling of manufacturing operations to show how a typical industrial scenario can be dealt with by our temporal reasoner based on IDSA subalgebra. We believe that many interesting robotic manufacturing processes fall into the class represented by this simple scenario where a set of operations has to be accomplished within a given time. For the moment, we do not address the problem of resources allocation.

Each operation will have a specific processing time and, with respect to other operations, will have one or more of the following characteristics:

1. it must precede certain other operations;
2. it is independent of certain operations;
3. it is concurrent with some operations; and
4. it is synchronized with external events.

This problem can be regarded as a particular instance of the (so-called) *generalized job shop scheduling* (Chu & Ngai, 1993) where different operations in a job can be processed in parallel and different jobs have different release times and deadlines.

In a small factory there is a production line to make plastic lids for car batteries. The main operations are the injection moulding by means of a machine, the control of the quality of the product, and the packaging of the final product. After the moulding operation, carried out by the machine, a first robot, R1, takes the lid from the machine and

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

controls its quality; a second robot, R2, prepares a packing and packages the lid. This second robot, moreover, has to fit two lead bushings into the machine before the beginning of the subsequent moulding (to produce another lid). As the durations of the various operations are not homogeneous, machines and robots have some idle time.

What we want to show is how the use of the temporal reasoner, which can manage the real temporal constraints among operations, leads to an effective reduction of idle times.

The operations of the two robots to produce a lid are the following:

1. R1 takes the lid from the machine, when it is ready; for this operation the robot has to synchronize with an external event, that is the end of the machine moulding. This operation takes a time varying from 5 to 10 s. The machine takes from 60 to 70 s to execute the moulding.
2. R2 fits the lead bushings into the machine before the beginning of the subsequent moulding. This operation immediately follows the first one, and takes from 5 to 10 s.
3. R2 prepares a packing, which requires from 15 to 25 s.
4. R1 controls the quality of the lid at least 20 s after operation (1) to let the lid cool down. This control takes from 25 to 30 s.
5. R2 packages the lid in 10–20 s. Of course, this operation has to be executed after operations (3) and (4).

Each robot can perform only one task at a time. We define a reference time line; we suppose that the machine starts working at the beginning of the time line.

The problem we address is to find a possible solution for the production of one lid, satisfying a given deadline, which, for example, might be 120 s.

All the available knowledge can be represented by a temporal network, as shown in Figure 3. Each node represents the interval during which a task is performed (for the sake of simplicity, the interval has the same name as the task associated with it). Each interval has a duration which can vary between a lower and an upper bound (in the form $[L,U]$ or $d_{ij}(L,U)$), except for the time line and for the interval `lid ready` that starts as soon as the lid is packed (relation $m = meets$ with the interval `package`) and lasts indefinitely. As an example, the information that the interval `take lid` has a duration from 5 to 10 s is represented by the pair [5,10] composed by the bounds of the range of variability of the interval.

The deadline we have set is represented by the constraint:

$$[ini, d11(0,120)]$$

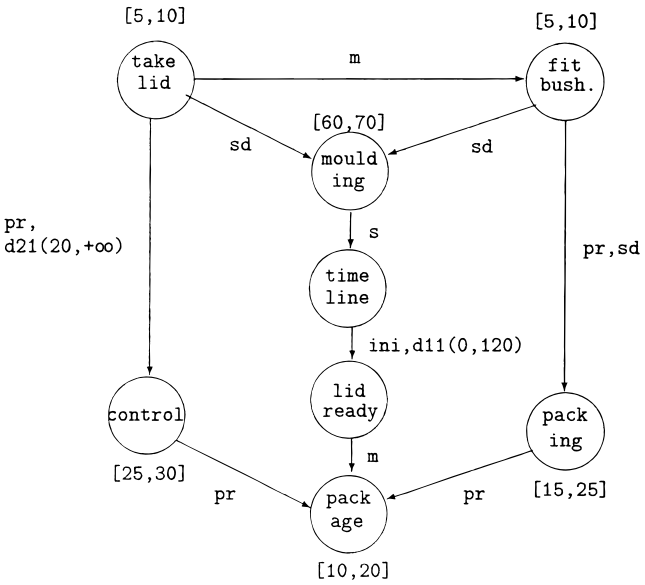


Fig. 3. The initial temporal network.

between the time line and the interval during which the cover is ready. This constraint states that:

- the interval `lid ready` is contained within the time line by means of the relation *ini*, which is the inverse of the relation:

$$in = starts \vee during \vee finishes \vee equal$$

- the distance between its starting point and the starting point of the time line varies from 0 to 120.

The operations of fitting the bushings and preparing the packing cannot occur simultaneously because they are both executed by the second robot. Thus, between the corresponding intervals a constraint *precedes* $\vee$ *succeeds* ($pr \vee sd$ in the figure) must hold: one interval must precede or succeed the other, and no other relation can hold between the two.

The relation $m = meets$ between the intervals `fit bushings` and `take lid` translates the information that the operation of fitting must immediately follow that of grasping. To represent the fact that the machine starts working at the beginning of the time line, a relation *starts* (s in the figure) is specified between the interval `moulding` and the time line.

Once we have described the initial knowledge in terms of temporal entities, temporal constraints and deadlines to be satisfied, the temporal reasoner can solve a variety of queries such as checking consistency, finding a solution, and computing the minimal network.

Let us now apply the algorithm to find a solution as defined in Section 7.2. The system computes two possible solutions as two sets of assignment of values to all variables that satisfy all the constraints. The network has two consis-

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

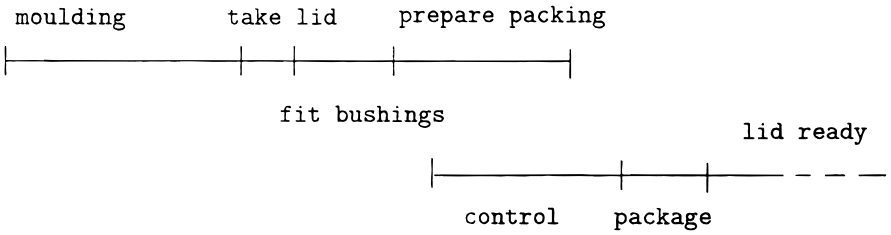


Fig. 4. First schedule.

tent labellings. In the first case (shown in Fig. 4), the interval `fit bushings` precedes the interval `prepare packing`, while, in the second one (shown in Fig. 5), the interval `fit bushings` succeeds the interval `prepare packing`. In the first solution, intervals have the following extremes (each interval A is considered as $\{A^-, A^+\}$, where A^- and A^+ are the exact values of extremes relative to the interval I_0):

`take lid` = {60, 65}
`fit bushings` = {65, 75}
`prepare packing` = {75, 90}
`control` = {80, 95}
`package` = {95, 105}
`lid ready` = {105, +∞}
`moulding` = {0, 60}

In the second case a possible solution is:

`take lid` = {60, 65}
`fit bushings` = {65, 75}
`prepare packing` = {0, 15}
`control` = {80, 95}
`package` = {95, 105}
`lid ready` = {105, +∞}
`moulding` = {0, 60}

We can see that the task of preparing the packing can be accomplished in parallel with that of quality control and with that of moulding. These are two possible schedules of the operations computed by the temporal reasoning system.

We can also observe that the idle time between the grasping operation and the quality control for letting the lid cool down is utilized in both cases to perform a different operation.

We can also note that, in the second scenario, the moulding operation is processed in parallel with the packing preparing operation.

The minimal network computed by the system can be represented by a table with an entry for each node (Table 2). The minimal network is a complete graph, so that we know the feasible relationships between any pair of intervals. For example, we can infer that interval `fit bushings` must always (that is, in each solution) precede `control`, while initially the relationship between these two intervals was unknown.

Note that in the table some qualitative relations belonging to SAC are not elementary relations of the Interval Algebra. These are: *disi*, which stands for the disjunction $di \vee si$, *difi*, which is the disjunction $di \vee fi$, *ppnli* is the disjunction $bi \vee mi \vee oi \vee fi$ and *lpnni* the disjunction $si \vee oi \vee mi \vee bi \vee e \vee s \vee d \vee f$.

We can conclude that, even with initial conditions of uncertain and inaccurate knowledge, the system manages to work out an optimal ordering for the operations considered, taking into account all the deadlines, partial and total.

10. CONCLUSION

In this article, we have presented a general frame for representing qualitative and quantitative constraints relative to

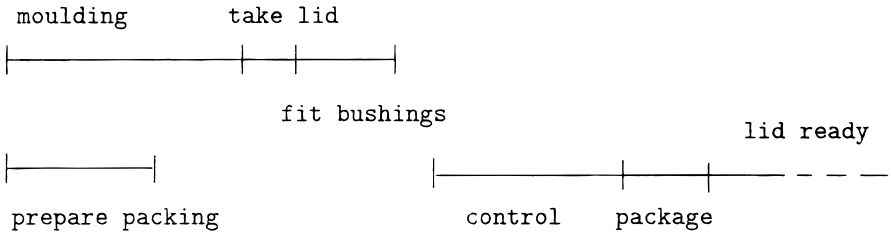


Fig. 5. Second schedule.

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

Table 2. *The minimal network*

Int.	Take lid	Fit bush	Packing	Control	Package	Lid ready	Mould
time line	*di*, *d*(60, 75), *d*(65,80), *d*(−240,−225), *d*(−235,−220)	*di*, *d*(65, 80), *d*(75,95), *d*(−235,−220), *d*(−225,−205)	*disi*, *d*(0,65), *d*(15,80), *d*(−300,−235), *d*(−285,−220) *difi*, *d*(75, 285), *d*(90, 300), *d*(−225,−15), *d*(−210, 0)	*di*, *d*(80,95), *d*(95,110), *d*(−220,205), *d*(−205,−190)	*di*, *d*(95,110), *d*(105,120), *d*(−195,−180), *d*(−195,−180)	*difi*, *d*(105,120), *d*(o(105),300), *d*(−195,−180), *d*(−195,0)	*si*, *d*(0,0), *d*(60,70), *d*(−300,−300), *d*(−240,−230)
take lid		*m*, *d*(5,10), *d*(15,25), *d*(0,0), *d*(10,15)	*ppnli*, *d*(−75,−5), *d*(−60,10), *d*(−80,−15), *d*(−65,0) *b*, *d*(15, 225), *d*(30, 240), *d*(10, 220), *d*(25, 235)	*b*, *d*(20,35), *d*(35,50), *d*(15,30), *d*(30,45)	*b*, *d*(35,50), *d*(45,60), *d*(30,45), *d*(40,55)	*b*, *d*(45,60), *d*(o(45),240), *d*(40,55), *d*(o(40),235)	*sd*, *d*(−75,−60), *d*(−15,0), *d*(−80,−65), *d*(−20,−5)
fit bush			*sd*, *d*(−80,−15), *d*(−65,0), *d*(−95,−25), *d*(−80,−10) *pr*, *d*(10,220) *d*(25,235), *d*(0, 210), *d*(15,225)	*pr*, *d*(15,30), *d*(30,45), *d*(0,20), *d*(15,35)	*b*, *d*(30,45), *d*(40,55), *d*(15,35), *d*(25,45)	*b*, *d*(40,55), *d*(o(40),235), *d*(25,45), *d*(o(25),225)	*bi*, *d*(−80,−65), *d*(−20,−5), *d*(−95,−75), *d*(−35,−15)
packing				*b*, *d*(30,95), *d*(15,80), *d*(30,95) *nr*, *d*(−205,20), *d*(−190,35), *d*(−220,5), *d*(−205,20)	*b*, *d*(55,120), *d*(30,95), *d*(40,105) *nr*, *d*(−190,35), *d*(−180,45), *d*(−205,20), *d*(−190,30)	*b*, *d*(o(55),300), *d*(40,105), *d*(o(40),285) *nr*, *d*(−180,45), *d*(o(−180),225), *d*(−195,30), *d*(o(−195),210)	*lpnni* *d*(−5,70), *d*(−80,−15) *d*(−20,55) *bi*, *d*(−285,−75), *d*(−225,−15), *d*(−300,−90), *d*(−240,−30)
control					*pr*, *d*(15,30), *d*(25,40), *d*(0,15), *d*(10,25),	*b*, *d*(25,40), *d*(o(25),220), *d*(10,25), *d*(o(0),205),	*bi*, *d*(−95,−80), *d*(−35,−20), *d*(−110,−95), *d*(−50,−35)
package						*m*, *d*(10,20), *d*(o(10),205), *d*(0,0), *d*(o(0),195),	*bi*, *d*(−110,−95), *d*(−50,−35), *d*(−120,−105), *d*(−60,−45)
lid ready							*bi*, *d*(−120,−105), *d*(−60,−45), *d*(−300,o(−105)), *d*(−240,o(−45))

temporal intervals. Temporal knowledge is represented in a temporal network where nodes are associated with intervals and arcs are labelled by binary constraints referring to intervals. To specify quantitative information, we refer, on the algorithmic level, to the extremes of intervals, that is, to time points.

We have shown that our representation, by blending together the two types of information homogeneously, has an

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press

expressive power that is greater than both Interval Algebra and Distance Algebra representations.

On the contrary, the complexity seems to be lower than that of Meiri's representation (Meiri, 1991) because the number of singleton labellings to be examined in our framework is always less than or equal to the number of singleton labellings of a Meiri network.

Another important feature of this representation is that temporal information is kept as compact as possible, that is, the number of disjunctions for each constraint is as small as possible, so as to reduce computational effort. This is particularly important when utilizing the reasoning system in planning and scheduling problems where knowledge is uncertain and the notion of disjunction is needed to represent alternatives and/or mutual exclusions.

We have applied the system to solve a scheduling problem in a manufacturing scenario where a set of operations have to be accomplished within a given time. We have shown how the temporal reasoning system can provide a schedule of manufacturing operations by managing any type of real temporal constraints, qualitative and quantitative.

ACKNOWLEDGMENTS

This work has been developed at LADSEB-CNR of Padova. It has been partially funded by the National Council of Research and by the Italian Ministry of University, Science and Technology.

REFERENCES

Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *Comm. of the ACM 11(26)*, 832–843.

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

Allen, J.F. (1991). Planning as temporal reasoning. *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, 3–14.

Allen, J.F., & Ferguson, G. (1994). Actions and events in interval temporal logic. *J. Logic and Computation 4*, 5.

Allen, J.F., Kautz, H.A., Pelavin, R.N., & Tenenberg, J.D. (eds.) (1991). *Reasoning about plans*. Morgan Kaufmann Publishers Inc., San Mateo, CA.

Badaloni S., & Berati, M. (1996). Hybrid temporal reasoning for planning and scheduling. *Proc. of the 3rd Int. Workshop on Temporal Representation and Reasoning (TIME-96)*. 39–44.

Badaloni, S., Berati, M., & Pagello, E. (1995). Templar: A temporal planner for monitoring the actions of an autonomous robot used in plant experiments. *Proc. 4th Int. Conf. on Intelligent Autonomous Systems IAS-4*, 95–101.

Becker, S. (1995). Hybrid temporal and non-temporal knowledge for scheduling. *Proc. of the TIME-95*, 29–40.

Chu, W.W., & Ngai, H.P. (1993). Embedding temporal constraint propagation in machine sequencing for job shop scheduling. *AI EDAM 7(1)*, 37–52.

Dean, T., & Kambhampati, S. (1995). Planning and scheduling. *CRC Handbook of Computer Science and Engineering*, 1–38.

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraints networks. *Artificial Intelligence 49*, 61–95.

Freksa, C. (1992). Temporal reasoning based on semi-intervals. *Artificial Intelligence 54*, 199–227.

Galton, A. (1990). A critical examination of Allen's theory of action and time. *Artificial Intelligence 42*, 159–188.

Ghallab, M. & Laruelle, H. (1994). Representation and control in IxTeT, a Temporal Planner. *Proc. AIPS-94*, 61–67.

Golumbic, M.C., & Shamir, R. (1993). Complexity and algorithms for reasoning about time: A graph-theoretic approach. *J ACM 40*, 1108–1133.

Jaffar, J., & Lassez, J.L. (1987). From unification to constraints. *Japan LP Conf.*

Jaffar, J., Maher, M.J., Stuckey, P., & Yap, R. (1992). *Projecting CLP(R) constraints*. IBM Research Report.

Kautz, H.A., & Ladkin, P.B. (1991). Integrating metric and qualitative temporal reasoning. *Working Notes AAAI Spring Symposium Series*, 67–78.

Knight, B., & Ma, J. (1994). Time representation: A taxonomy of temporal models. *Artificial Intelligence Review 7*, 401–419.

Ma, J., & Knight, B. (1994). A general temporal theory. *The Computer Journal 37(2)*, 114–123.

Ma, J., Knight, B., & Petridis, M. (1994). A revised theory of action and time based on intervals and points. *The Computer Journal 37(10)*, 847–857.

Mackworth, A.K. (1992). The logic of constraint satisfaction. *Artificial Intelligence 58(1–3)*, 3–20.

Meiri, I. (1991). Combining qualitative and quantitative constraints in temporal reasoning. *Proc. of the AAAI-91*, 260–267.

Nebel, B., & Burckert, H.J. (1995). Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *Journal of the Association for Computing Machinery 42(1)*, 43–66.

Rit, J.F. (1986). Propagating temporal constraints for scheduling. *Proc. of the AAAI-86*, 383–388.

Shoham, Y. (1987). Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence 33(1)*, 89–104.

Trudel, A. (1993). A formal specification of Allen's processes. *Proc. Second Symposium on Logical Formalization of Commonsense Reasoning*, 208–216.

Tsang, E. (1993). *Foundations of Constraint Satisfaction*. Academic Press, San Diego, CA.

van Beek, P. (1992). Reasoning about qualitative temporal information. *Artificial Intelligence 58*, 297–326.

van Beek, P., & Cohen, R. (1990). Exact and approximate reasoning about temporal relations. *Comput. Intell. 6*, 132–144.

van Beek, P., & Manchak, D.W. (1996). The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research 4*, 1–18.

Vidal, T., & Ghallab, M. (1995). Constraint-based temporal management in planning: The IxTeT approach. Report No. LAAS N.95313.

Vilain, M., & Kautz, H. (1986). Constraint propagation algorithms for temporal reasoning. *Proc. AAAI-86*, 377–382.

Zweben, M., & Fox, M.S. (1994). *Intelligent scheduling*. Morgan Kaufmann, San Mateo, CA.

Silvana Badaloni is a Research Associate at the Department of Electronics and Computer Science (DEI) of the Faculty of Engineering, University of Padova. She has a degree in Physics and an M.Sc. in Artificial Intelligence. Until 1984, her field of research was plasma physics. Her current field of investigation includes temporal reasoning, nonmotonic reasoning, and planning. She has studied applications of temporal reasoning to problems of scheduling and planning as regards robots. In 1998, she chaired the Sixth Conference of the Italian Association of Artificial Intelligence, AI*IA98.

https://doi.org/10.1017/S0890060499132037 Published online by Cambridge University Press