Temporal Reasoning with Qualitative and Quantitative Information about Points and Durations*

Rattana Wetprasit and Abdul Sattar
Knowledge Representation and Reasoning Unit
School of Computing and Information Technology
Griffith University, NATHAN, QLD 4111 AUSTRALIA
{rattana,sattar}@cit.gu.edu.au

Abstract

A duration is known as a time distance between two point events. This relationship has recently been formalized as the *point duration network* (PDN) in (Navarrete & Marin 1997). However, only the qualitative information about points and durations was considered. This paper presents an *augmented point duration network* (APDN) to represent both qualitative and quantitative information about point events. We further extend APDN to capture quantitative information about durations. We propose algorithms to solve reasoning tasks such as determining satisfiability of the network, and finding a consistent scenario with minimal domains. Thus, we present an expressively richer framework than the existing ones to handle both qualitative and quantitative information about points as well as durations.

Introduction

Temporal knowledge can be classified into two main categories: qualitative and quantitative (or metric) information. Relationships between events (e.g., Fred arrived at work before John) are considered as a class of qualitative information while numeric distance or an event instance (e.g., Fred took 15-20 minutes to get to work) is considered as quantitative information. Interval algebra (Allen 1983) and point algebra (Vilain & Kautz 1986) are two traditional models to represent and reason with qualitative information when events are considered as intervals and points, respectively. In (Dean & McDermott 1987) and (Dechter, Meiri, & Pearl 1991), two systems for handling metric information between point events were proposed. The integration of qualitative and quantitative information between point and interval events was attempted in (Meiri 1996) and (Kautz & Ladkin 1991).

In (Barber 1993), an object-oriented approach with two types of items: points and durations was introduced. This approach can represent qualitative and quantitative constraints between points and durations but no disjunction of the constraints is allowed. Recently, a point based bi-network to represent qualitative relationships among point events and durations, so called *point duration network* (PDN) has been proposed (Navarrete & Marin 1997). In their framework, a duration represents a time distance between two point events. The basic relations between two durations are: $\{<, >, =\}$, indicating a duration is either shorter, longer or equal to another duration.

In this paper, we examine the frameworks proposed for representing information about points and durations, in particular the recently proposed point duration network framework. Let us consider the example proposed in (Meiri 1996) with additional qualitative information about durations and quantitative information about points concerning Bob's traveling.

Example 1 John, Fred and *Bob* work for a company that has local and main offices in Los Angeles. They usually work at the local office, in which case it takes John less than 20 minutes and Fred 15-20 minutes to get to work. Twice a week John works at the main office, in which case his commute to work takes at least 60 minutes. Today John left home between 7:05-7:10 a.m., and Fred arrived at work between 7:50-7:55 a.m. We also know that Fred and John met at a traffic light on their way to work. *Since Bob lives close to the office, it takes him less time than Fred to go to work and today he leaves home before 7:45 a.m.* □

To the best of our knowledge, none of the existing frameworks can adequately handle this sort of information. We would like to have a system that can sufficiently represent qualitative as well as quantitative information about points and durations. For example, it should be able to deduce that *today Bob arrives at work not later than 8.05 a.m.* We also expect our system to retain the reasoning ability of the existing systems such as deducing that John arrived at the main office after 8.05 a.m., and he arrives at work at least 10 minutes after Fred.

In this paper, we present an augmented point duration network by introducing unary constraints to all point and duration variables. This new framework allows us to:

1. represent and reason with both qualitative and quantitative constraints over point events and durations between points;
2. provide an expressively rich point duration network framework that can handle the disjunction of qualitative and quantitative (metric) constraints (dealing with possible uncertain knowledge); and
3. effectively use the existing techniques for point algebra networks and constraint satisfaction problems to solve the reasoning problems within these extended networks.

Our intuition behind constraining point and duration variables with unary constraints is that the quantitative information about when each point takes place indicates the instance of the corresponding point. The metric information about each pair of points specifies the distance between the two points, which is the instance of the corresponding duration. Therefore, the quantitative temporal information can be naturally represented by constraining the domains of points and durations. Let us consider the above example, if we anchor the beginning of the world, x_0, to 7:00 a.m., other time instances in the above story are represented with respect to x_0. Let us consider Fred's traveling. We denote F_1 and F_2 as the time that Fred leaves home and arrives at work respectively. From the given information that Fred arrives at work between 7:50-7:55 a.m., the domain of F_2 is restricted to the time interval (50,55). The time distance from F_1 to F_2 is also limited to (15,20) by the fact that Fred takes 15-20 minutes to get to work. By the distance property, we can simply infer that the domain of F_1 or the time that Fred leaves home is between (30,40) or 7:30-7:40 a.m.

Definitions

Before defining a point duration network, we first review the point algebra as the underlying structure of the proposed framework. Point algebra (PA) considers an event as a time instance, mapping to a rational number on an imaginary time line. The three possible basic relations that can hold between any two points is a set (T) of $\{<, >, =\}$. A PA network is a binary constraint network where the variables represent time points $x_1, ..., x_n$ having the same domain, i.e., the set of rational numbers Q. The binary relation $R_{i,j}$ is a disjunction of the basic point relations in T.

We introduce qualitative and quantitative constraints which will be used in the augmented framework and the further extension.

Definition 1 A *qualitative constraint* between two objects O_i and O_j, in which both objects may be a pair of points or durations, is a disjunction of the form

$$(O_i r_1 O_j) \vee ... \vee (O_i r_k O_j)$$

where each of the r_i's is a basic relation in T.

Definition 2 A *quantitative constraint* is represented by a set of intervals[1]:

$$I = \{I_1, ..., I_k\} = \{[a_1, b_1], ..., [a_k, b_k]\}.$$

- If $a_l \neq b_l$ $(1 \leq l \leq k)$ and $k > 1$ then the constraint is classified as *multiple-interval.*
- If $a_l \neq b_l$ $(1 \leq l \leq k)$ and $k = 1$ then the constraint is classified as *single-interval.*
- If $a_l = b_l$ $(1 \leq l \leq k)$ and $k \geq 1$ then the constraint is classified as *discrete.*

There are two types of quantitative constraints:

1. A *unary constraint* quantitatively restricts the domain of a variable, say O_i, to the given set of intervals. Essentially, it represents the disjunction:

$$(a_1 \leq O_i \leq b_1) \vee ... \vee (a_k \leq O_i \leq b_k).$$

The three types of domains are multiple-interval, single-interval, and discrete, corresponding to the three classes of quantitative constraints.

2. A *binary constraint* represents the metric information between durations (for more detail see the further extension section).

Point Duration Network

The *point duration network* (PDN) was first formulated in (Navarrete & Marin 1997)[2]. A PDN consists of two binary networks: point and duration networks. Domains of point and duration variables are rational numbers, while the binary constraints in both networks are qualitative constraints. For example, $R_{ij,km} = \{\leq\}$, indicates that the duration from point i to j is equal to or shorter than the duration from k to m. The two networks are related by a set of ternary constraints specifying the relationship between points and durations.

Augmented Point Duration Network

We propose to augment the PDN framework with unary domain constraints to enforce the handling of both qualitative and quantitative information about points, and qualitative information about durations.

Definition 3 An *augmented point duration network* (APDN) is a structure $\Sigma_{APD} = \langle N_P, N_D, Rel(P, D)\rangle$, where

- N_P is a network consisting of a set (P) of point variables: $\{x_1, ..., x_n\}$; the domains of points: $\{D_1, ..., D_n\}$, which are restricted by unary constraints; and a set ($Rel(P)$) of binary relations over point variables,

$$Rel(P) = \{R_{i,j} \in 2^T \mid 1 \leq i, j \leq n\}.$$

[1]For simplicity, we assume closed intervals, but the same treatment can be applied to open and semi-open intervals as well. This is similar to a set of intervals for TCSP defined in (Dechter, Meiri, & Pearl 1991).

[2]In (Allen & Kautz 1985), the notion of duration was investigated in terms of one duration being a proportion of another.

- N_D is a network consisting of a set (D) of duration variables: $\{d_{ij} \mid 1 \leq i < j \leq n\}$; the domains of durations: $\{D_{12}, ..., D_{(n-1)n}\}$, which are restricted by unary constraints; and a set $(Rel(D))$ of binary relations over duration variables,

$$Rel(D) = \{R_{ij,km} \in 2^T \mid 1 \leq i, j, k, m \leq n\}.$$

- $Rel(P, D)$ is a set of *ternary* constraints relating points and durations, where $Rel(P, D) = \{\triangle_{ij} \subseteq \mathcal{Q}^3 \mid 1 \leq i, j \leq n\}$ such that $\triangle_{ij} = \{(X_i, X_j, D_{ij}) \in \mathcal{Q}^3 \mid D_{ij} = |X_i - X_j|\}$.

The $Rel(P)$, $Rel(D)$, $Rel(P, D)$, and all unary constraints altogether are referred to as Σ_{APD}-constraints. When a domain of the network is restricted to a simple interval, we call it a *simple domain*.

A duration variable d_{ij} represents time elapsed between points x_i and x_j in an absolute value form, i.e., only d_{ij} $(i < j)$ is represented (not d_{ji}). The set of ternary constraints, $Rel(P, D)$, specifies instances of points and durations which are related to each other by the distance property $d_{ij} = |x_i - x_j|$.

Illustration: (Continued from Example 1) J_1, J_2, B_1, B_2 denote the time that John and Bob respectively leave home and arrive at the office. All given information can be represented in the APDN as shown in Figure 1. The beginning of the story is anchored at 7:00 a.m. Therefore, the information that John left home between 7:05-7:10 a.m. is represented as a single-interval domain of point J_1, i.e., (5,10), and the same treatment is applied for the time that Fred arrived at work (F_2), and Bob left home (B_1). The time that John takes in reaching, either his local office (less than 20 minutes), or the main office (at least 60 minutes), is represented by the multiple-interval domain of the duration node J_1J_2. The information that Fred takes between 15-20 minutes to go to work is represented as the domain of the duration node F_1F_2. The qualitative relation that *Bob takes less time than Fred to go to work* is specified by the constraint between durations B_1B_2 and F_1F_2. The incomplete qualitative information that *Fred and John met at a traffic light on their way to work* can be interpreted as either a *start, started by, during, contain, finish, finished-by, overlapped, overlapped-by, or equal* relationship between the two events of Fred and John going to work. This can be represented by a conjunction of relations between endpoints of the two intervals as in the point network of Figure 1.

Consistency and Minimality

Given an APDN, $\Sigma_{APD} = \langle N_P, N_D, Rel(P, D)\rangle$ with n point variables $(x_1, ..., x_n)$, and the domains of points $(D_1, ..., D_n)$. An assignment of all variables in N_P is the n-tuple of the form:

$$A_P = ((x_1, X_1), ..., (x_n, X_n)), \qquad X_i \in D_i.$$

Similarly, the assignment of all $\frac{n(n-1)}{2}$ duration variables in N_D $(d_{12}, ..., d_{(n-1)n})$ with the domain constraints $(D_{12}, ..., D_{(n-1)n})$ is the tuple of the form:

$$A_D = ((d_{12}, Y_{12}), ..., (d_{(n-1)n}, Y_{(n-1)n})), \qquad Y_{ij} \in D_{ij}.$$

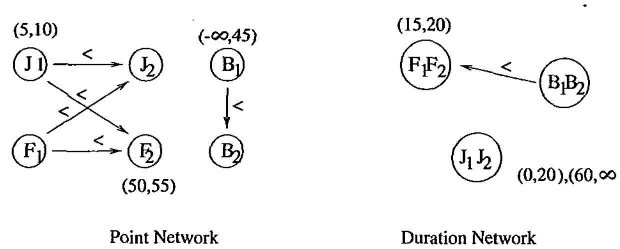


Figure 1: The graphical representation of Example 1

A pair $A(A_P, A_D)$ is *a solution* for the APDN iff it satisfies all the Σ_{APD}-constraints. An APDN is *consistent* iff there is a solution.

A value X_i is a feasible value for a variable x_i if there exists a solution in which $x_i = X_i$. The set of all feasible values of a variable is called the *minimal domain.* We can also say the same for the minimal domain of a duration variable.

A *simple APDN*, $\Sigma^S_{APD} = \langle N^S_P, N^S_D, Rel^S(P, D)\rangle$, is an APDN such that every qualitative constraint is a basic relation and every quantitative constraint is an element of the single-interval class.

Definition 4 A *consistent scenario of an APDN with minimal domains* is a consistent simple APDN where all constraints are minimal.

In next section, we present an algorithm to find a consistent scenario of APDN with minimal domains.

Reasoning with APDN

The useful reasoning tasks for an APDN are determining:

1. satisfiability of the network;
2. a consistent scenario with minimal domains; and
3. the minimal APDN.

The satisfiability problem is a special case of finding a consistent scenario of an APDN with minimal domains (or the minimal APDN). However, computing consistent scenarios of PDNs is an NP-complete problem (Navarrete & Marin 1997). Similarly, we cannot expect better computational complexity in the case of APDNs, as stated in the following theorem:

Theorem 5 *Deciding the satisfiability of an augmented point duration network (APDN) is an NP-complete problem.*

This theorem can be proved by showing that the APDN framework is an integration of the PDN framework and the *temporal constraint satisfaction problem* (TCSP) introduced in (Dechter, Meiri, & Pearl 1991). A TCSP network is a constraint network of point events with unary (domain) and binary quantitative constraints. Each constraint is represented by a set of intervals I defined in Definition 2.

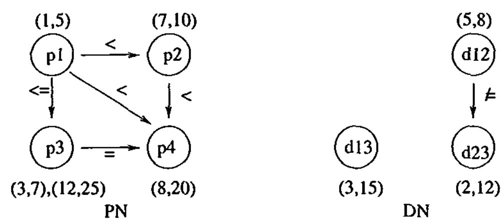
We propose an algorithm for finding a consistent scenario of an APDN with minimal simple domains. To do so, instead of starting from the simple case, as proposed for PDN and TCSP, we allow non-atomic relations in point and duration networks with multiple-interval domains as input. Our goal is to prune the unnecessary search space before decomposing the main problem into several simple problems. This algorithm comprises two functions: CSAPDN and DomainMinimize.

Algorithm *CSAPDN_MinD*
Input: An APDN
Output: A consistent scenario of APDN with minimal simple domains
If CSAPDN(Σ_{APD},Σ^S_{APD}) then
if DomainMinimize(Σ^S_{APD}) then
the APDN is consistent.

The CSAPDN finds a consistent scenario of APDN, if succeed then DomainMinimize finds minimal simple domains with respect to the consistent scenario.

To demonstrate the functioning of our algorithm, the following simple example will be used through out this section:

Example 2 An APDN of four points $(p_1, ..., p_4)$ with qualitative and quantitative constraints as shown below:



□

Computing Consistent Scenarios

The core structure of CSAPDN is motivated by the algorithm CSPAN for finding a solution for PA networks (Van Beek 1992). Here, we need to consider the relationship between points and durations and also their domains.

The main description of CSAPDN is as follows:

Step 1 Since all nodes in the same class (if instantiated, they will have '=' relationship) must have the same domain, our first task is to identify all equivalent classes of nodes in the point and duration networks individually. This is the same as finding the *strongly connected components* (SCCs) in graphs for which the efficient algorithms by (Tarjan 1972) can be applied[3].

Illustration: In PN, points p_3 and p_4 are in the same SCC, while none of durations can be classified into the same class. Points p_1, p_3 and p_4 are not in the same class as '=' $\notin R_{1,4}$.

Step 2 From the SCCs in PN and DN, deduce more nodes that can be classified into same SCCs using the following properties:

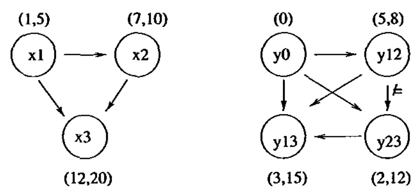
[3]Nodes i and j belong to the same SCC if there exists a path from i to j and a path from j to i which contain only the edges labeled with '$\leq$' or '=', e.g., $i \leq j = i$.

- If two points i and j are in the same SCC then it implies: **1)** For every point $k \neq i, j$, it must be $d_{ik} = |k - i| = |k - j| = d_{jk}$ (d_{ik} and d_{jk} must be in the same SCC); and **2)** $d_{ij} = 0$ (d_{ij} must be in the same SCC as the null duration d_0).
- If two durations, ik and jk ($i \neq j$) are in the same SCC then points i and j must be in the same SCC. The same statement can also be made for ki and kj.

Illustration: Since points p_3 and p_4 are in the same SCC, then a) duration d_{34} has null distance; b) the distances from point p_1 to p_3 and p_4 are equal, or d_{13} and d_{14} are in the same SCC. This also applies to d_{23} and d_{24}. In the duration network, no further node can be grouped into SCC.

Step 3 Condense PN and DN by collapsing each SCC into a single node. The domain of each new node will be the intersection of all domains in the SCC, and the relation between a pair of the new nodes will be the intersection of the relations from nodes in one SCC to another SCC. Thus, the original APDN is reduced to the point and duration networks such that all nodes are in different SCCs. If any intersection results in empty set, the corresponding constraint is inconsistent.

Illustration: The point network then is reduced to three nodes: x_1, x_2, and x_3, while x_3 includes points p_3 and p_4. The reduced duration network consists of four nodes: y_0 with the null duration d_{34}, and y_{12}, y_{23} and y_{13} with the durations (d_{12}), (d_{23}, d_{24}) and (d_{13}, d_{14}) respectively. Relations and domain constraints are shown as follows (the direction of the arrow from node i to j indicates i *is less than* j, if not labeled otherwise). The interval $(3, 7)$ on the domain of p_3, after intersection with domain of p_4, becomes empty set.



Step 4 Find a consistent scenario for the APDN, in other words, a pair of consistent scenarios of reduced networks for PN and DN that satisfy each other. This step is much like the function Exist_Solution proposed in (Navarrete & Marin 1997). We first find a consistent scenario for PN and DN independently. Then each duration in the consistent scenario of DN is instantiated with an integer d corresponding to the ordering of the durations. Using the distance property $x_j = x_i + d_{ij}$, the value of each point in PN is then calculated. If all relations between the point values in PN are satisfied then a consistent scenario for the APDN is found.

Illustration: We choose consistent scenarios of PN and DN as: $x_1 < x_2 < x_3$, and $y_0 < y_{12} < y_{23} < y_{13}$, $y_0 < y_{23}$, $y_0 < y_{13}$, $y_{12} < y_{13}$. Then without considering the domains of all nodes, we sequentially assign integers to all durations: $y_0 = 0$, $y_{12} = 1$, $y_{23} = 2$ and $y_{13} = 3$. By the distance property and the initial value

C	QUAN(C)
$<$	$(0, \infty)$
$\leq$	$[0, \infty)$
$=$	$[0]$
$>$	$(-\infty, 0)$
$\geq$	$(-\infty, 0]$
$\neq$	$(-\infty, 0), (0, \infty)$
$?$	$(-\infty, \infty)$

Table 1: The QUAN translation

$x_1 = 0$, we have $x_2 = x_1 + y_{12} = 1$ and $x_3 = 3$, which are consistent with their atomic relations in the consistent scenario. Therefore, the chosen consistent scenario of PN is consistent with the one for DN.

Computing Minimal Domains

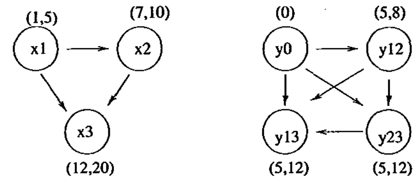
The subfunction DomainMinimize takes the consistent scenario, $\Sigma^S_{APD} = \langle N^S_P, N^S_D, Rel^S(P, D)\rangle$, from the subfunction CSAPDN with (possible multiple-interval) domains as an input, and returns the minimal simple domains with respect to N^S_P and N^S_D[4]. A general description of this subfunction is as follows:

Step 1 Find the domains that are consistent with the qualitative relations of N^S_P and N^S_D individually by applying arc-consistency. In (Meiri 1996), the domains of a nonempty arc- and path-consistent CPA network (the PA network without '$\neq$' relation) over multiple-interval domains are shown to be minimal. A consistent scenario is certainly k-consistent ($k \leq n$)[5], and all qualitative labels are non-disjunctive constraints (no '$\neq$' relation). Therefore, acquiring an arc-consistent network at this step will result in all minimal domains of the corresponding network. The main operation of the arc-consistency algorithm REVISE$((i, j))$ makes arc (i, j) consistent by tightening the domain of node i according to the domain of node j and the qualitative constraint between i and j:

$$D_i := D_i \otimes (D_j - QUAN(C_{i,j}))$$

Function QUAN(C) transforms qualitative temporal constraints to quantitative constraints (Meiri 1996) (shown in Table 1).

Illustration: The consistent scenarios of N^S_P from the previous steps is already arc-consistent. Enforcing arc-consistency on N^S_D results in the updating of two domains: D_{23} and D_{13}. Consider arc (d_{23}, d_{12}), function REVISE$((d_{23}, d_{12}))$ is expressed as: $D_{23} := D_{23} \otimes (D_{12} - QUAN(C_{23,12})) = (2, 12) \otimes ((5, 8) - (-\infty, 0)) = (2, 12) \otimes (5, \infty) = (5, 12)$. The domain of d_{23} has been updated, and thus all related arcs to d_{23} ((d_{12}, d_{23}), (d_{13}, d_{23}) and (d_0, d_{23})) are added into queue. Then after the propagation terminates, the arc-consistent N^S_P and N^S_D are as shown below:



Step 2 Minimize the domains with respect to both N^S_P and N^S_D. As mentioned in relation to the proof of Theorem 5, the quantitative information represented in our framework is equivalent to the metric constraints represented in TCSP. A multiple-interval domain of a point p_i in N^S_P is equivalent to the unary constraint of a point in TCSP. This constraint is represented as the binary constraint between points p_i and p_0 (the beginning of the world) in TCSP, and implies the disjunction:

$$(a_1 \leq p_i - p_0 \leq b_1) \vee ... \vee (a_k \leq p_i - p_0 \leq b_k)$$

where $a_1, ..., a_k$ and $b_1, ..., b_k$ are endpoints of multiple intervals defined in Definition 2. The multiple-interval domain of a duration d_{ij} in N^S_D is equivalent to the metric constraint between points p_i and p_j in TCSP. This constraint represents the disjunction:

$$(a_1 \leq p_j - p_i \leq b_1) \vee ... \vee (a_k \leq p_j - p_i \leq b_k).$$

A special polynomial case of TCSP when all constraints are single intervals is called the *simple temporal problem* (STP) (Dechter, Meiri, & Pearl 1991). The minimal STP network can be found by applying Floyd-Warshall's All-Pairs-Shortest-Paths algorithm. (Dechter, Meiri, & Pearl 1991) also showed that applying the path-consistency algorithm (Mackworth 1977) to an STP network is identical to applying Floyd-Warshall's All-Pairs-Shortest-Paths algorithm. Here, when all unary constraints of the APDN are restricted to single intervals (simple domains), they form an equivalent STP network. Therefore, enforcing global path-consistency over all simple domains of N^S_P and N^S_D concurrently, results in all minimal simple domains of the consistent scenario of the APDN.

Looked at in the context of STP, the path-consistency conditions of the quantitative constraints in Σ^S_{APD} can be expressed as follows:

$$D_i = D_j - D_{ij} \quad (1)$$

$$D_j = D_i + D_{ij} \quad (2)$$

$$D_{ij} = D_j - D_i \quad (3)$$

$$D_{ij} = D_{ik} - D_{jk} \quad (4)$$

where $1 \leq i < j \leq n, 1 \leq k \leq n, k \neq i, j$. If the domain of a point is constrained (by Condition 1 or 2), we need to consider all other related domains (Conditions 1, 2 and 3). If the domain of a duration node is updated (by Condition 3 or 4), we examine Conditions 1, 2 and 4. We introduce three queues: QPD, QPP and QDD.

[4]For simplicity, we will refer to a component in the reduced network of PN as a point node, and a component in the reduced network of DN as a duration node.

[5]As noted in (Meiri 1996), the notation of k-consistency here is slightly different from the orginal definition (Freuder 1978) since at this stage, we consider the consistency as per infinite domains.

Elements in QPD represent the domain indices in Conditions 1, and 2, while QPP and QDD are for Conditions 3 and 4 respectively. The propagation of the four conditions over the domains of APDN is repeated until all domains are stable, or become empty indicating inconsistency.

While propagating such conditions, we maintain the consistency between each pair of domains and the atomic relations N_D^S by calling arc-consistency algorithm. Our modified version of arc-consistency algorithm determines only the arcs related to the updated domain. The algorithm also returns the indices of all affected nodes for further propagation.

When the propagation terminates, the resulting network is a consistent scenario of the APDN with minimal simple domains.

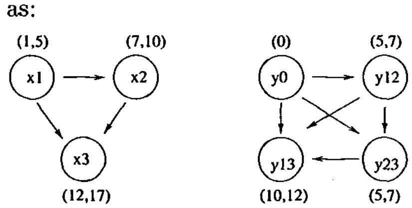
Illustration: For this example, initially QPD and QPP each consist of three elements: (1,2),(1,3) and (2,3). QDD has three elements: (1,3,2),(1,2,3) and (2,1,3).

With an element (1,3) in QPD, the domain of d_3 is updated as follows: $D_3 \otimes (D_1 + D_{13}) = (12,20) \otimes ((1,5) + (5,12)) = (12,20) \otimes (6,17) = (12,17)$. Then the domains of all nodes connecting to p_3 in N_P^S are computed by arc-consistency algorithm, and elements $(1,3), (2,3)$ are added in QPD, and $(3,1), (3,2)$ in QPP. None of other elements in QPD updates other domains at this stage.

When consider QPP, the element (1,3) updates the domain of d_{13} as follows: $D_{13} \otimes (D_3 - D_1) = (5,12) \otimes ((12,17) - (1,5)) = (5,12) \otimes (7,16) = (7,12)$. Then checking arc-consistency on the domains of all nodes connecting to d_{13} in N_D^S does not affect other domains. The elements $(1,3)$ are added in QPD, and $(1,3,2), (2,1,3)$ in QDD. The domain of d_{23} is also updated to $(5,10)$ by the condition $D_{23} \otimes (D_3 - D_2)$.

The domain of d_{12} is updated by the element (1,3,2) from QDD as follows: $D_{12} \otimes (D_{13} - D_{23}) = (5,8) \otimes ((7,12) - (5,10)) = (5,8) \otimes (-3,7) = (5,7)$. Applying arc-consistency to N_D^S results in no change to other domains of the network. Then the elements $(1,2)$ are added in QPD, and $(1,2,3), (3,1,2)$ in QDD.

When the algorithm terminates, a consistent scenario with minimal simple domains of the given APDN is returned as:



Our algorithm can handle the general problem without the restriction of atomic labels. However, if such restriction is applied, our algorithm solves the desired problem in polynomial time.

Theorem 6 *Finding a consistent scenario with minimal domains of a simple APDN is solvable in polynomial time, $O(nd^2)$, where n, d are the numbers of points and durations respectively.*

In (Navarrete & Marin 1997), deciding the satisfiability of point duration networks when only qualitative information is considered has the complexity of $O(d^2)$.

The minimal APDN can be computed by repeatedly finding all consistent scenarios with their corresponding minimal simple domains. Then taking union over all consistent scenarios and minimal domains. Of course, this task requires exponential time in worst case. It is worth noting here that a consistent scenario can have several sets of minimal simple domains depending upon the number of multiple intervals given. However, consistent scenarios can be found efficiently as our algorithm prunes much of the search space using both qualitative and quantitative constraints.

Further Extensions to APDN

The augmented point duration network framework proposed earlier in this paper allows only qualitative constraints labeling arcs in the duration networks. As a result, the metric constraints between durations cannot be handled. In this section, we propose a further extension to the APDN to address the problem when quantitative information between durations is allowed e.g., *Bob takes 30-45 minutes less than Fred to go to work.*

The proposed extension, a *fully quantified point duration network*, is an APDN, as defined earlier, except that the binary constraints between durations in the duration network are *quantitative*. The binary constraint, $C_{ij,km} \in I$, restricts the permissible values for the time differences between durations ij and km; it represents the disjunction:

$$(a_1 \leq d_{km} - d_{ij} \leq b_1) \vee \ldots \vee (a_k \leq d_{km} - d_{ij} \leq b_k).$$

This can be one of the three constraint classes in Definition 2.

When representing the binary constraints between durations:

- if the qualitative relation between a pair of durations $\{<, >, =\}$ is given, we transform the relation into metric constraint using Table 1;
- if both qualitative and quantitative information is given, the metric constraint is the intersection of the given quantitative constraint and the transformation of the qualitative constraint from Table 1; and
- if there is no binary constraint between a pair of durations provided, we allow the infinite range $(-\infty, \infty)$ label.

Example 3 Combining the information from Example 1 and *Bob takes 30-45 minutes less than Fred to go to work*, the fully quantified point duration network representing this information is shown in Figure 2. □

A *solution* and the *consistency* of a fully quantified point duration network can be defined as in APDN.

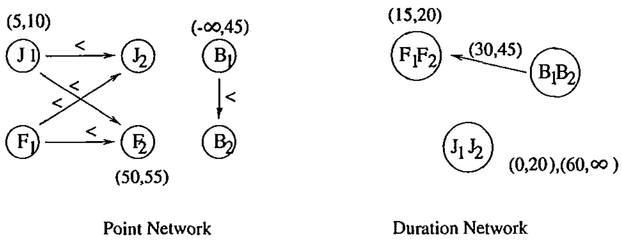


Figure 2: The graphical representation of the fully quantified problem

The main strategies for solving APDN introduced earlier are still useful for solving the consistency problem and finding a consistent scenario with minimal simple domains in a fully quantified point duration network. Here, we describe how the techniques proposed for APDN can be adapted to address the above problems.

Computing consistent scenarios

For APDN, after identifying the networks of all strongly connected components, we find a consistent scenario of the point network that is consistent to a consistent scenario of the duration network. Here, since all constraints in DN are quantitative constraints, we find a consistent scenario of PN that is consistent with a minimal simple DN (the DN with all constraints being single intervals and minimal). A minimal simple DN can be computed by applying the techniques for finding the minimal STP as proposed in (Dechter, Meiri, & Pearl 1991). We introduce a null duration d_0, and associate a directed edge-weighted graph, called a *distance graph*. The single-interval domain $[a, b]$ of duration d_{ij} represents a binary constraint:

$$a \leq d_{ij} - d_0 \leq b.$$

The metric constraint $[a, b]$ constrains the time difference between durations d_{ij} and d_{km} as follows:

$$a \leq d_{km} - d_{ij} \leq b.$$

Then applying *All-Pairs-Shortest-Paths* algorithm to the distance graph where arcs are weighted with the permissible values $[a, b]$. This results in the minimal distances between all pairs of nodes, thus the minimal simple duration network.

The consistency between a consistent scenario of PN and the minimal simple DN is determined by using a similar method as for APDN. The value of a point is $x_j = x_i + d_{ij}$ where d_{ij} in this case is the upper bound of the minimal domain of duration d_{ij}[6]. A consistent scenario of PN is consistent with the minimal simple DN if the atomic labels of the scenario are consistent with the point values computed by using the above equation.

[6]In (Dechter, Meiri, & Pearl 1991), the authors proved that for a minimal STP of n nodes with minimal domains, if d_{0i} and d_{i0} denote the upper and lower bounds of node i, there are two special solutions to the STP which are the tuples: $(d_{01}, ..., d_{0n})$ and $(-d_{10}, ..., -d_{n0})$. Here we can choose either one.

Computing minimal domains

Since the domains of the minimal simple DN are already minimal, we initially apply the arc-consistency algorithm only to the consistent scenario of PN (N_P^S). Then minimizing the quantitative constraints with respect to both N_P^S and N_D^S becomes almost the same as in APDN (Step 2 of the subfunction DomainMinimize). The propagation of the four constraints (Conditions 1,2, 3 and 4) specifying the global path-consistency conditions of both N_P^S and N_D^S, and the treatment of N_P^S remain the same as for APDN. However, instead of performing arc-consistency on N_D^S when a duration domain is updated, we call the All-Pairs-Shortest-Paths algorithm to maintain the consistency of all constraints. After all constraints are stable, the resulting network is a consistent scenario of a fully quantified point duration network with minimal simple domains.

Theorem 7 *Finding a consistent scenario with minimal domains of a simple fully quantified point duration network is solvable in polynomial time, $O(d^3)$, where d is the number of durations.*

Since relations between durations can be translated into binary quantitative constraints, this results in the following theorem:

Theorem 8 *An augmented point duration network (APDN) is a special case of a fully quantified point duration network.*

Discussion

This section discusses some additional advantages of APDN and fully quantified PDN over the existing frameworks, in particular Meiri's qualitative and quantitative model (Meiri 1996).

The unary constraints in APDN and fully quantified PDN can be viewed as the metric constraints between pairs of points in TCSP. Let us consider our path-consistency algorithms in the context of TCSP. At the global domain minimizing step (corresponding to the propagating of Conditions 1, 2, 3, and 4), the algorithms determine the consistency of all possible length-two paths. If any arc (corresponding to a domain in APDN/fully quantified PDN) is updated, the algorithm ensures the arc-consistency of the corresponding PN or DN network by calling the arc-consistency algorithm. Generally, performing arc-consistency checks the consistency between domains of all pairs of nodes with respect to the constraints between each two corresponding nodes. Here, each domain of APDN specifies all permissible distances between two points as in TCSP. This means the arc-consistency algorithm checks the consistency between any two distances, or four points. Therefore, consistency between four points (cf. TCSP) is implicitly maintained in APDN/fully quantified PDN framework by ensuring path-consistency. Hence, the path-consistency in APDN/fully quantified PDN parallels 4-consistency in TCSP.

In (Meiri 1996), Allen's interval algebra relations, Vilain & Kautz's point algebra relations and quantitative information between points are represented in a hybrid network. Nodes of the network are either intervals or points. Arcs connecting nodes representing points are labeled with quantitative constraints, otherwise they are labeled with qualitative relations. Conceptually, this network combines the TCSP and Allen's interval algebra frameworks. However, this framework is not capable of capturing the qualitative and quantitative information about durations, such as the information about Bob, given in Example 1. This is crucial as it is a general type of information which arises in various applications. A technical example pointed out in (Kautz & Ladkin 1991) is the problem of translating metric constraints to Allen's interval relations as follows[7]:

Example 4 Given metric information about the durations of two intervals I and J: $3 < (I_2 - I_1) < \infty$ and $-\infty < (J_2 - J_1) < 2$, I_1, I_2 and J_1, J_2 denote the starting and ending points of intervals I and J respectively. The reasoning system should be able to infer that *I has longer duration than J*, thus *I cannot be during J*. □

In the APDN framework, the above information is represented as domain constraints of duration nodes I_1I_2 and J_1J_2. The proposed path-consistency algorithm will infer the single constraint that *"duration I_1I_2 is longer than J_1J_2"*. Hence, APDN can further infer *"I cannot be during J"*. Meiri's framework cannot represent and reason about the relation lengths of different durations. Therefore, it would be unable to conclude this final inference without first achieving 4-consistency.

Conclusion

Precisely, the contributions of this paper are:

- An augmented point duration network (APDN) which adequately handles both qualitative and quantitative information about point events.
- A further extension of the APDN framework to capture quantitative information about durations.
- The algorithms for finding a consistent scenario with minimal domains for both APDN and the extended frameworks.
- Identifying simple cases for which the time complexities of the proposed algorithms are polynomial.

Like other point-based approaches which are restricted to conjunctions of binary relations, the point duration network based model cannot handle the disjointedness of interval algebra (e.g., Interval A is either before or after interval B). Our ongoing research is to investigate an APDN-based framework that can deal with full interval relations. One possibility is combining the APDN framework with the *generalized multi-point event* framework (GMPE) (Wetprasit, Sattar, & Khatib 1997). GMPE represents the disjunction of interval relations by using disjunction of matrices of relations between interval endpoints.

[7](Kautz & Ladkin 1991) proved that to compute the strongest set of basic interval relations, metric constraints between four points (extreme points of two intervals) must be considered at a time.

Acknowledgments

We thank Peter van Beek for his valuable comments which improved the preliminary version of this paper. Also, thanks to Michael Maher and the anonymous reviewers for their helpful comments.

References

Allen, J., and Kautz, H. 1985. A model of naive temporal reasoning. In J.Hobbs, and R.Moore., eds., *Formal Theories of the Commonsense World.* Ablex. 251–268.

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

Barber, F. 1993. A metric time-point and duration-based temporal model. *SIGART Bulletin* 4(3):30–49.

Dean, T., and McDermott, D. 1987. Temporal data base management. *Artificial Intelligence* 32:1–55.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Freuder, E. 1978. Synthesizing constraint expressions. *Communications of the ACM* 21(11):958–966.

Kautz, H., and Ladkin, P. 1991. Integrating metric and qualitative temporal reasoning. In *Proceedings of AAAI-91*, 241–246.

Mackworth, A. 1977. Consistency in networks of relations. *Artificial Intelligence* 8:99–118.

Meiri, I. 1996. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence* 87:343–385.

Navarrete, I., and Marin, R. 1997. Qualitative temporal reasoning with points and durations. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1454–1459.

Tarjan, R. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1:146–160.

Van Beek, P. 1992. Reasoning about qualitative temporal information. *Artificial Intelligence* 58:297–326.

Vilain, M., and Kautz, H. 1986. Constraint propagation algorithms for temporal reasoning. In *Proceedings of AAAI-86*, 377–382. San Mateo: Morgan Kaufman.

Wetprasit, R.; Sattar, A.; and Khatib, L. 1997. A generalised framework for reasoning with multi-point events. In *Lecture Notes in Computer Science 1345; Advances in Computing Science, Proceedings of the third Asian Computing Science Conference (ASIAN'97)*, 121–135. Kathmandu, Nepal: Springer.