Qualitative and Quantitative Temporal Reasoning with Points and Durations*
(An Extended Abstract)

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

This paper extends the point duration network *(PDN) [3] to represent both qualitative and quantitative information about point events. An algorithm to solve reasoning tasks such as finding a consistent scenario with minimal domains is briefly described. We also propose further extension to capture quantitative information about durations.*

1. Introduction

Recently, a point based bi-network is proposed to represent qualitative relationships between point events and durations—called *point duration network* (PDN) [3]. In this framework, a duration represents a time distance between two point events. The basic relations between two durations are: $\{<, >, =\}$. Let us consider the example proposed in [2] with additional qualitative information about durations and quantitative information about points.

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 usually he leaves home before 7:45 a.m..* □

We would like to deduce that *Bob always 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.

To handle the above problem, we proposed an augmented point duration network by introducing unary constraints over all point and duration variables. Our intuition behind constraining point and duration variables with unary constraints is that the quantitative information about when each point event 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.

2. Definitions

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 the set T ($\{<, >, =\}$).

*A full version of this paper will appear in the Proceedings of the Fifteenth National Conference on Artificial Intelligence 1998 (AAAI-98), Wiscousin [5].

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: multiple-interval, single-interval, and discrete, correspond to the three classes of constraints.

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

2.1. Augmented Point Duration Network

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.

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

- 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.

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

[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 [1].

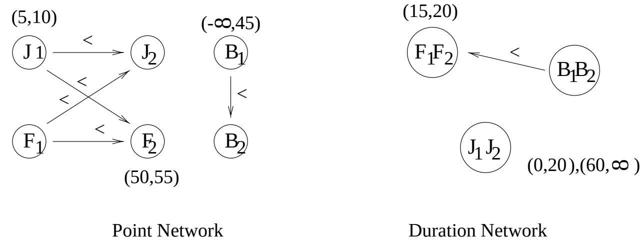


Figure 1. The graphical representation of Example 1

- $Rel(P, D)$ 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.

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) and (F_1, F_2) denote the time that John, Bob and Fed respectively leave home and arrive at the office. By representing the quantitative information as domains of points and durations, all given information can be represented in the APDN as shown in Figure 1.

2.2. 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:

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

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 primitive relation and every quantitative constraint is an element of a single-interval class.

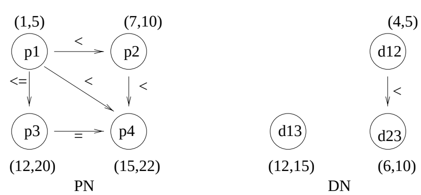
Definition 4 A *consistent scenario of an APDN with minimal domains* is a consistent simple APDN where all constraints are minimal.

3. Reasoning with APDN

We propose an algorithm for finding a consistent scenario of an APDN with minimal domains. This algorithm comprises two functions: find a consistent scenario of APDN, and if the APDN is consistent then compute single-interval minimal 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 Given an APDN of four points: $p_1, ..., p_4$, qualitative and quantitative constraints as shown below:



□

3.1. Computing Consistent Scenarios

Step 1 Identify all equivalent classes of nodes in the point and duration networks independently. This is the same as finding the *strongly connected components* (SCCs) in graph theory and the efficient algorithms by [4] can be applied.

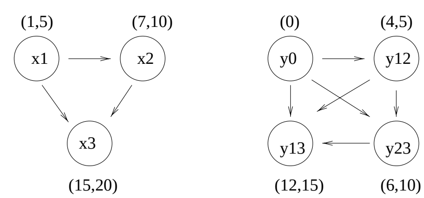
Step 2 From the SCCs in PN and DN, deduce more nodes that can be classified in the SCCs using the following properties:

- 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: From the above example, points p_3 and p_4 are in the same SCC. Therefore, 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 an SCC to another SCC. If any intersection results in empty set, the corresponding network 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. Relation and domain constraints are shown as follows (the direction of the arrow from node i to j indicates i *is less than* j):



Step 4 Find a consistent scenario for PN and DN independently. To identify a consistent scenario of the APDN, $\Sigma^S_{APD} = \langle N^S_P, N^S_D, Rel^S(P, D)\rangle$, 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 $x_1 = 0$, we have $x_2 = x_1 + y_{12} = 1$ and $x_3 = 3$, which are consistent to their atomic relations in the consistent scenario. Therefore, the chosen consistent scenario of PN is consistent to the one for DN.

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

3.2. Computing Minimal Domains

The minimal domains of the consistent scenario resulting from the previous process are acquired by the following steps[2]:

Step 1 Apply arc-consistency algorithm to N_P^S and N_D^S individually. The main operation of the arc-consistency algorithm REVISE((i,j)) can be expressed as:

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

Function QUAN(C) transforms qualitative temporal constraints to quantitative constraints [2] (shown in Table 1).

Illustration: Both the consistent scenarios of PN and DN from the previous steps are already arc-consistent. However, if we consider arc (x_1, x_2), function REVISE((x_1, x_2)) is expressed as: $D_1 := D_1 \otimes (D_2 - QUAN(C_{1,2})) := (1,5) \otimes ((7,10) - (0,\infty)) := (1,5) \otimes (-\infty, 10) := (1,5)$.

Step 2 Minimize the domains with respect to both N_P^S and N_D^S by propagating all the quantitative constraints which can be expressed as follows:

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

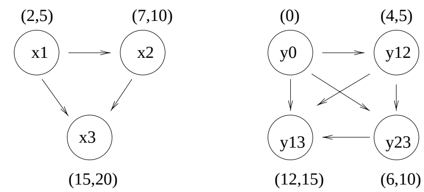
$$D_j = D_i + D_{ij} \quad (2)$$

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

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

where $1 \leq i < j \leq n, 1 \leq k \leq n, k \neq i, j$. If the domain of a point, D_i, is constrained (by Equation 1 or 2), we need to consider all other related domains (Equation 2 and 3, where $1 \leq i < j \leq n$). If the domain of a duration node, D_{ij}, is updated (by Equation 3 or 4), we examine Equations 1, 2 and 4, where $1 \leq k \leq n, k \neq i, j$. We introduce three queues: QPD, QPP and QDD. Elements in QPD represent the domain indices in Equations 1, and 2, while QPP and QDD are for Equations 3 and 4 respectively. The propagation is repeated until all domains are stable or become empty, indicating inconsistency. While propagating, the consistency of the domains among the nodes in the same network needs to be maintained.

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). Considering the element (1,2) of QPD: $Temp = D_1 \otimes (D_2 - D_{12}) = (1,5) \otimes ((7,10) - (4,5)) = (2,5)$. This tightens the domain of x_1, but the network is still arc-consistent. The elements ((2,1),(3,1)) and ((1,2),(1,3)) are added to QPD and QPP respectively. When the algorithm terminates, no further domains are updated. A consistent scenario of the given APDN with minimal domains is shown below:



Theorem 5 *Finding a consistent scenario of APDN with minimal domains for a simple APDN is solvable in polynomial time, $O(nd^2)$, where n, d are the numbers of points and durations respectively.*

4. Further Extensions to APDN

In this section, we propose to further extend the APDN to address the problem when quantitative information between durations is allowed. A *fully quantified point duration network* is an APDN, as defined earlier, except the binary constraints between durations in the duration network are *quantitative constraints*. ($C_{ij,km} \in I$). This restricts the permissible values for the time differences between durations ij and km.

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.

Finding a *consistent scenario* of a fully quantified point duration network with minimal domains can be achieved by similar method as for APDN. However, as all constraints in the duration network are quantitative, we apply the All-Pairs-Shortest-Paths algorithm to minimize all those single-intervals constraining arcs and nodes of the duration network (detailed in the full paper). Then minimizing all domains with respect to both point and duration networks is also based on the propagating of those equations proposed in the previous section.

[2]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.

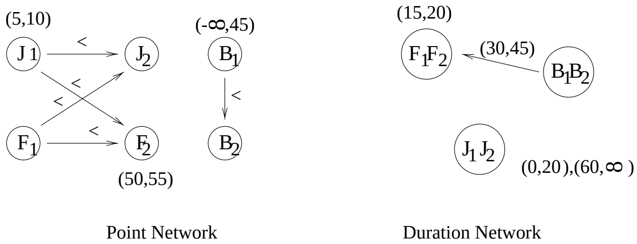


Figure 2. The graphical representation of the fully quantified problem

5. 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.

References

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

[2] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence*, 87, 1996.

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

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

[5] R. Wetprasit and A. Sattar. Temporal reasoning with qualitative and quantitative information about points and durations. In *Proceeding of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press / The MIT Press, 1998. To appear.