Qualitative Temporal Reasoning with Points and Durations

Isabel Navarrete and Roque Marin*
Departamento de Informatica y Sistemas
Universidad de Murcia
30.071-Espinardo, Murcia, Spain
e-mail: inava@dif.um.es

Abstract

We present here a qualitative temporal reasoning system that takes both points and durations as primitive objects and allows relative and indefinite information. We formaly define a *point duration network,* as a structure formed by two point algebra (PA) networks separately but not independently, since ternary constraints are introduced for relating point and duration information. We adapt some of the concepts and reasoning techniques developed for the point algebra networks, such as consistency and minimality. We prove that the problem of determining consistency in a point duration network is NP-complete. A simpler and polynomial-time decision problem is introduced for a restricted kind of point duration networks. Finally we suggest how to determine consistency and find minimal point duration network in the general case.

1 Introduction

Representing and reasoning about temporal knowledge is essential for many areas of Artificial Intelligence. Several constraint-based systems for temporal reasoning have been proposed, mainly concentrated on two kinds of formalisms: qualitative approaches [Allen, 1983; Vilain and Kautz, 1986] and quantitative or metric systems [Dean and McDermott, 1987; Dechter *et* a/., 1991]. Later efforts [Meiri, 1991; Kautz and Ladkin, 1991] have been done on integrating both qualitative and quantitative information between time points and intervals in a single constraint-based computational model for temporal reasoning. Systems supporting duration reasoning have been proposed as well. Allen [1983] has designed a duration reasoning system that allows relative information (e.g. interval I took longer than interval J) and representing uncertainty. Duration information is encoded in a network orthogonal to the interval relationship network, but total consistency of this network is not guaranteed. Temporal constraints on durations are not usually managed in point-based formalims. Barber [1993] presented a duration-based temporal model with metric constraints. But this model is restricted in the sense that no disjunctive qualitative constraints are allowed.

We present here a qualitative temporal reasoning system that takes both points and durations as primitive objects and allows relative and indefinite information. We formaly define a *point duration network,* PDN for short, as a structure formed by two PA networks separately but not independently, since ternary constraints are introduced for relating point and duration information. In section 2 we adapt some of the concepts developed for the point algebra networks, such as consistency and minimality, for the new point-duration model. In section 3 we propose some reasoning task for PD networks and prove that the problem of determining consistency in a PDN is NP-complete. A simpler and polynomial-time decision problem is introduced for a restricted kind of PD networks. Finally we suggest how to determine consistency and find minimal network with an exponential-time algorithm for the general case.

2 Definitions

In this section we review Vilain and Kautz's [1986] point algebra (PA) and PA networks for representing qualitative relations between points. Then we will see how we can augment PA networks with additional variables that represent durations (or elapsed time) between any two points of time, and additional relations expressing relative information concerning durations (e.g. duration between temporal points x_i and x_j is less than duration between points x_k and x_m).

The *point algebra* (PA) is a relation algebra [Tarski, 1941] whose elements are the possible subsets of $T = \{<, >, =\}$, where T is the set of (mutually exclusive)

*This work was funded by the Spanish CICyT through grant TIC95-0604

primitive or basic qualitative temporal relations that can be hold between any two points of time.

A *PA network* is a network of binary relations [Montanari, 1974] where the variables $x_1, \ldots, x_n$ represent time points having the same domain, that may be, for example, the set Q of rational numbers, and the binary relations between variables are of the form R_{ij}, where R_{ij} is a relation of the point algebra that constraints the possible values for variables x_1 and X_j,

2.1 Point Duration Networks (PDN)

We define a *point duration network,* PDN for short, as a struture $\Sigma_{PD} = \langle N_P, N_D, Rel(P,D)\rangle$ formed by two PA networks N_P and N_D and a set of ternary constraints relating points and durations together, where

- N_P is determined by a set $P = \{x_1, \ldots, x_n\}$ of point variables that take values over the rational numbers and a set

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

of binary relations between points of time.

- N_D is given by a set $D = \{d_{ij} \mid 1 \leq i < j \leq n\}$ of duration variables, again over the rationals, and a set

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

of binary relations between durations.

- $Rel(P,D) = \{\Delta_{ij} \subseteq \mathbb{Q}^3 \mid 1 \leq i,j \leq n\}$ such that

$$\Delta_{ij} = \{(X_i, X_j, D_{ij}) \in \mathbb{Q}^3 \mid D_{ij} = |X_i - X_j|\}$$

We refer to $Rel(P), Rel(D)$ and $Rel(P,D)$ altogether as Σ_{PD}*-constraints.*

We can represent indefinite information both in the N_P and N_D networks, since the relations R_{ij} and $R_{ij,km}$ are allowed to be a disjunction of primitive relations in T. Each duration variable d_{ij} represents time elapsed between two temporal points X_i and X_j. It does not supposes anything about the relative position of points X_i and X_j. This information is encoded in the N_P network. In order to properly compare the magnitude of separation between points, we consider durations must take non-negative values and so we use the euclidean distance to model durations between points. Consequently, equations $d_{ij} = |x_i - X_j|$ impose ternary constraints that show the influence of points over durations and viceversa. Since $d_{ij} = |x_i - x_j| = |X_j - x_i| = d_{ji}$ we need only one duration variable d_{ij} $(i < j)$ for representing time elapsed between any two points x_i, and X_j. Hence, variable d_{ji} would be redundant and it is not considered as part of durations set D.

As a consequence of ternary constraints $Rel(P, D)$, the PA networks N_P and N_D are not independent of each other and thus we cannot solve them as independent binary constraint satisfaction problems (CSP). Note that we use the term network for a Σ_{PD} structure although this is not a network in a strict sense, since Σ_{PD} does not describe a binary CSP. Alternatively, we could also consider the problem represented by Σ_{PD} as a general (nonbinary) and continuous domain CSP whose variables are $V = P \cup D$, the domain of each variable is Q, and $C = Rel(P) \cup Rel(D) \cup Rel(P,D)$ is the set of binary and ternary constraints. We prefer presenting the problem as two PA networks N_P and N_D separately but not independently since it clearer expresses the different meaning of point and duration variables and constraints, and also offers the possibility of borrowing some of the representation and reasoning techniques developed in the study of point algebra and binary networks. We follow an idea suggested by Allen [1983] when he proposes a duration reasoning system whose duration information is encoded in a network orthogonal to the interval relationship network.

2.2 Consistent and Minimal PDN

Given a PD network $\Sigma_{PD} = \langle N_P, N_D, Rel(P,D)\rangle$ with n point variables, a n-tuple of the form

$$C_P = (\langle x_1, X_1\rangle, \ldots, \langle x_n, X_n\rangle)$$

that denote the assignment $X_1, \ldots, X_n$ to $x_1, \ldots x_n$ respectively, is a *n-compound label* [Tsang, 1993] for point variables. Similarly, a *d-compound label* for duration variables, where $d = \frac{n \times (n-1)}{2}$ is a d-tuple which assigns d rationals values to d duration variables, i.e.,

$$C_D = (\langle d_{12}, D_{12}\rangle, \ldots, \langle d_{(n-1)n}, D_{(n-1)n}\rangle)$$

A pair $C = (C_P, C_D)$ is a *consistent instantation* of the PD network Σ_{PD} if an only if the assignments to point and duration variables given by C_P and C_D satisfy all the Σ_{PD}-constraints, what means that,

$$\begin{array}{l} \forall X_i, X_j : (X_i, X_j) \in R_{ij} \\ \forall D_{ij}, D_{km} : (D_{ij}, D_{km}) \in R_{ij,km} \\ \forall X_i, X_j, D_{ij} : (X_i, X_j, D_{ij}) \in \Delta_{ij} \end{array} \qquad (1)$$

The $(n+d)$-tuple

$$S = (X_1, \ldots, X_n, D_{12}, \ldots, D_{(n-1)n})$$

satisfiying conditions (1) is a *solution tuple* of the network Σ_{PD}.

A PDN is *consistent* if at least one solution tuple exists. Otherwise the network is *inconsistent.* A consistent PD network Σ_{PD} represents a $(n+d)$-ary relation Σ_{PD}-$\rho \subseteq \overbrace{\mathbb{Q} \times \ldots \times \mathbb{Q}}^{(n+d)}$ defined by the set of all solution tuples of Σ_{PD}.

Semanticaly, a consistent instantation of a network Σ_{PD} is a description of a world where we can map points to a time line, in such a way, we not only preserve the relative position amongst points but also the relative magnitude of separation between them.

Example 1 *Let Σ_{PD} be a PDN with three point variables, such that $x_1 < x_2$, $x_2 < x_3$ and $d_{12} > d_{23}$ (anyother variables are unconstrained). One consistent instantation to this network may be $C = (C_P, C_D)$, where*

$$C_P = (\langle x_1, 3\rangle, \langle x_2, 6\rangle, \langle x_3, 7\rangle)$$
$$C_D = (\langle d_{12}, 3\rangle, \langle d_{13}, 4\rangle, \langle d_{23}, 1\rangle)$$

A PD network $\Sigma^S_{PD} = \langle N^S_P, N^S_D, Rel^S(P, D)\rangle$ so that every constraint from $Rel(P)$ and $Rel(D)$ is a primitive relation, is called a *simple PDN*.

A simple PD network $\Sigma^S_{PD} = \langle N^S_P, N^S_D, Rel^S(P, D)\rangle$ is a *consistent scenario* of $\Sigma_{PD} = \langle N_P, N_D, Rel(P, D)\rangle$ if an only if the following conditions hold,

1. $P^S = P$, and $D^S = D$
2. $\forall R_{ij} \in Rel(P),\ R^S_{ij} \in Rel^S(P) : R^S_{ij} \subseteq R_{ij}$
3. $\forall R_{ij,km} \in Rel(D),\ R^S_{ij,km} \in Rel^S(D) : R^S_{ij,km} \subseteq R^S_{ij,km}$
4. Σ^S_{PD} is consistent

One consistent instantation $C = (C_P, C_D)$ of a PD network Σ_{PD} determines a consistent scenario Σ^S_{PD}. Indeed, for every two values X_i, X_j in C_P, just one primitive relation R^S_{ij} from T is satisfied, since primitive relations are mutually exclusive; so it is the case that $x_i\ R^S_{ij}\ x_j$ and $R^S_{ij} \subseteq R_{ij}$ and hence we take R^S_{ij} as the new constraint between variables x_i and x_j in $Rel^S(P)$ of Σ^S_{PN}. In the same way, we take $R^S_{ij,km}$ as the primitive relation of $Rel^S(D)$ that is satisfied by the assignment D_{ij}, D_{km} to variables d_{ij}, d_{km} in C_D.

Two PD networks $\Sigma^1_{PD} = \langle N^1_P, N^1_D, Rel^1(P, D)\rangle$ and $\Sigma^2_{PD} = \langle N^2_P, N^2_D, Rel^2(P, D)\rangle$ with the same variables are *equivalent* iff they have the same solutions, what means that Σ^1_{PD}-ρ = Σ^2_{PD}-ρ. Following Montanari's work [1974] on binary CSP, we can define a partial ordering among the equivalence class of all PD networks representing the same $(n + d)$-ary relation. The ordering relation, affecting only binary constraints, is defined as follows,

$$\Sigma^1_{PD} \subseteq \Sigma^2_{PD} \text{ iff } R^1_{ij} \subseteq R^2_{ij} \text{ and } R^1_{ij,km} \subseteq R^2_{ij,km}$$

for every $1 \leq i, j, k, m \leq n$. As well, we can define the *intersection* of two equivalent PD networks Σ^1_{PD} and Σ^2_{PD} as a new equivalent PD network $\Sigma_{PD} = \Sigma^1_{PD} \cap \Sigma^2_{PD}$ whose binary relations are given by,

$$R_{ij} = R^1_{ij} \cap R^2_{ij} \text{ and } R_{ij,km} = R^1_{ij,km} \cap R^2_{ij,km}$$

for every $1 \leq i, j, k, m \leq n$.

Given a PDN Σ_{PD}, there exists a unique PDN, Σ^M_{PD}, equivalent to Σ_{PD} which is minimal with respect to $\subseteq$ (the uniqueness is guaranteed because equivalent networks are closed under intersection, proofs can be found in [Montanari, 1974]). Σ^M_{PD} is the *minimal point duration network* representing Σ_{PD}-ρ and the binary relations in Σ^M_{PD} are called the *minimal relations.* Each binary and primitive relation in Σ^M_{PD} is *feasible* [Van Beek, 1992], i.e, we can find a consistent instantation of Σ^M_{PD} which satisfies the given relation.

3 Reasoning Task with PD Networks

Given a PDN, some of the reasoning task we can think about are:

- Determining consistency.
- Find the minimal relation between two point or duration variables.
- Find the minimal network equivalent to a given one.

Van Beek [1992] gives exact algorithms for these problems in the context of PA networks. But, as we suggested previously in section 2, these algorithms are not suitable for PD networks. We can see that with a very simple example.

Example 2 *Let Σ_{PD} be a PDN with three point variables, such that $x_1 < x_2$, $x_2 < x_3$ and $d_{13} < d_{12}$ (anyother variables are unconstrained). There is no consistent instantation because it must be $d_{13} = d_{12} + d_{23}$, but since $d_{23} \geq 0$ it is not possible that $d_{13} < d_{12}$. So Σ_{PD} is inconsistent although Np and ND, considered as independent PA networks, are consistent.*

3.1 Consistency in PND

From the above PDN reasoning task, the main one is to determine consistency since we can find a polynomial transformation from the latest tasks to the first one. This would be useful if we could check for consistency in polynomial time. But, unfortunately, this is not possible, as we show in the next theorem. Let CONS_PDN be the decision problem of determining if a given PDN is consistent or not.

Theorem 1 *CONS.PDN is NP-complete.*

Proof: We follow the general procedure described by Garey and Johnson [1979] for devising an NP-completeness proof for a decision problem. First we show CONSJPDN belongs to the class NP. This is easy, since for a YES instance of the problem, a nondeterministic Turing machine needs only to guess a consistent instantation and check in polynomial time that the assignments satisfy all the Σ_{PD}-**constraints.** In a second stage we must find a polynomial reduction of known NP-complete problem to CONS.PDN. We use GRAPH

COLORING for this purpose. An instance of this problem is a graph $G = (V, E)$ and an integer k, and the question is: is there a mapping $f : V \longrightarrow \{1, 2, \ldots, k\}$ such that $(v, w) \in E$ implies $f(v) \neq f(w)$?

Given an undirected graph $G = (V, E)$ with $|V| = n$ and an integer k we show how to construct a PD network $\Sigma_{PD} = \langle N_P, N_D, Rel(P, D)\rangle$ such that Σ_{PD} has a consistent instantation if an only if there is a coloring of G using k colors. The set of point variables is

$$P = \{t_1, \ldots, t_k, t_{k+1}\} \cup \{x_{k+2}, \ldots, x_{k+n+1}\}$$

where each $x_{k+j}, (j > 1)$ correspond to a vertex of V. The set of duration variables is

$$D = \{d_{ij} \mid 1 \leq i < j \leq k+n+1\}$$

We impose two kinds of constraints between point variables. First, $t_i < t_{i+1}, \forall 1 \leq i \leq k$. Second, for each vertex $v \in V$ associated with variable x_{k+j} we require that

$$\begin{array}{l} t_1 < x_{k+j} \\ t_i \neq x_{k+j}, \forall 2 \leq i \leq k \\ x_{k+j} < t_{k+1} \end{array} \qquad (2)$$

Our intention is that $]t_i, t_{i+1}[$ is associated with color i and just with constraints (2) a vertex v can be mapped to any of the k colors. Now we must avoid two vertices connected by and edge being mapped to the same color. That is why we introduce the following constraints on duration variables. For each edge $(v, w) \in E$ with associated variables x_{k+j} and x_{k+m} we require that

$$d_{i(i+1)} < d_{(k+j)(k+m)}, \forall 1 \leq i \leq k.$$

The above is clearly a polynomial transformation and finally we must show the equivalence of both problems. If there exists a consistent instantation of Σ_{PD}, contraints on points and durations forces two points v, w with $(v, w) \in E$ not to be assigned values on the same interval $]t_i, t_{i+1}[, \forall 1 \leq i \leq k$. So it must be possible to map vertex v and w to different colors. Conversely, if the answer to the instance of GRAPH COLORING problem is yes, we can find a consistent instantation of Σ_{PD}. For example,we take $t_1 = 1$ and $t_i = t_{i-1} + 3, \forall 1 < i \leq k+1$. Now ,$\forall 1 \leq i \leq k$ we take $d_{i(i+1)} = 3$. For the rest of point variables, corresponding to vertices of the graph we calculate the appropriate assignments in the following way. For vertices mapped to the same color we can assign the same value to its corresponding point variables. Suppose now we have $(v, w) \in E$. Then if x_{k+j} corresponds to v and x_{k+m} corresponds to w, then we can find apropriate values (even integer ones) from different "color intervals" such that $d_{(k+j)(k+m)} > 3$ as we require. Finally we compute the remaining assignments for duration variables calculating the distance between implicated points. ■

3.2 Consistency in Simple PDN

CONS-PDN so belongs to NP. But, can we really find a deterministic algorithm that solves this problem?, or what is the same, is CONSJPDN decidible?. To answer this question we propose a new decision problem CONSJSIMPLE.PDN: given a simple PDN, is the network consistent? We now show this problem can be solved in polynomial time.

Theorem 2 *CONSSIMPLE.PDN £ P.*

Proof: We prove this showing a polynomial time algorithm that for a given simple PD network Σ^S_{PD} returns YES when Σ^S_{PD} is consistent and returns NO when Σ^S_{PD} is inconsistent. The algorithm basicaly finds a consistent instantation of Σ^S_{PD} if the network is consistent and thus returns YES, or shows that no such instantation exists and returns NO. We associate a precedence graph [Meiri and Pearl, 1990] $G_{<,P} = (V_P, E_P)$ to the PA network *Np.* The set of nodes *Vp* are labeled with the indices of point variables and for every two points $x_i, x_j \in P$,

- if $x_i < x_j$ we add the arc $i \rightarrow j$ to E_P
- if $x_i > x_j$ we add the arc $j \rightarrow i$ to E_P and
- we add both $i \rightarrow j$, $j \rightarrow i$ to E_P just in the case $x_i = x_j$.

Now we associate a precedence graph to the PA network N^S_D. We take $V_D = \{ij \mid 1 \leq i < j \leq n\} \cup \{d_0\}$. Each node *ij* is associated with duration *dij* and the special node *do* represents the null duration. The set *ED* is calculated looking at duration constraints $Rel^S(D)$, in a similar way we have done with *Ep.* And for every duration variable, it must be $d_{ij} \geq d_0$, by distance properties. So we include in *ED* an arc *do* -> *ij* for each *dij* in *D.* In what follows it is requiered that $x_i R_{ij} x_j \Leftrightarrow x_j R_{ij}^{-1} x_i$, where ~^{I} is the inverse PA operation and always x_i, = x_i,. Similar assumptions are made for relations between durations.

We show in figure 1 a function for determinig consistency in a simple PDN, named ConsJSimple and use two auxiliary functions Same_SCC(SCC_P, i, j) that return true if vertices i and j are in the same strongly connected component $G_{<,P}$ Same_SCC(SCC_D, ij, km) t do the same with vertices **ij**, km of $G_{<,D}$. A third function ExistJSolution is used to find a consistent instantation if possible and returns true, otherwise returns false. For simplicity we just show the algorithms for ConsJSimple and ExistJSolution. Lines 1 to 3 just check if the PA network N^S_P is consistent and we adapt here Meiri and Pearl's consistency algorithm [1990]. We can say the same with lines 8 to 10 with respect to the PA network N^S_D. The idea is that if two vertices are in the same SCC this forces correspondig variables to be equal, otherwise Σ^S_{PD} would not satisfy *Rel(P)* or *Rel(D)* and thus the PDN would be inconsistent.

```
1. SCC_P := strongly connected components in G_{<,P};
2. for each i,j ∈ V_P such that Same_SCC(SCC_P, i, j)
   do begin
3.   if R_ij ≠ " = " then return (NO);
4.   for each (k ∈ V_P) and (k ≠ i,j) do
5.     if R_ik,jk ≠ " = " then return (NO);
6.   E_D := E_D ∪ {ij → d_0}
7. end;
8. SCC_D := strongly connected components in G_{<,D};
9. for each ij, km ∈ V_D such that
   Same_SCC(SCC_D, ij, km) do
10.  if R_ij,km ≠ " = " then return(NO);
11. for each ij ∈ V_D such that Same_SCC(SCC_D, ij, d_0)
    do begin
12.  if R_ij ≠ " = " then return (NO);
13.  for each (k ∈ V_P) and (k ≠ i,j) do
14.    if R_ik,jk ≠ " = " then return (NO);
15. end;
16. if Exist_Solution(Σ^S_PD, SCC_P, SCC_D) then return
    (YES);
17. else return (NO);
```

Figure 1: Function Cons_Simple $(\Sigma^S_{PD}, G_{<,P}, G_{<,D})$

If N^S_P and N^S_D are, we can say "independently consistent", we must finally prove if there is a consistent instantation that satisfies not only *Rel*(P) or *Rel*(D), but also relations in *Rel*(P,D). Lines 4 to 6 check two metric properties that must obey any two points variables that are equal. First if $x_i = x_j$ obviously for every other different point x_k it must be $d_{ik} = |x_i - x_k| = |x_j - x_k| = d_{jk}$. And second if $x_i = x_j$ then $d_{ij} = 0$ and so d_{ij} must be forced to be in the same SCC that null duration d_0. And conversely, if $d_{ij} = 0$ then $x_i = x_j$ and $d_{ik} = d_{jk}$ (lines 11 to 14).

Next we show the code for Exist_Solution in figure 2. By the construction of $G_{<,P}$ and $G_{<,D}$ and since all the relations between points and durations variables are primitive, the topological orders we calculate after finding the strongly connected components (lines 1,2), are always unique. Because of this, we can assign values to duration variables arbitrarily but according to the constraints in *Rel*(D) (line 4). Then we calculate values for point variables such that for every x_i, x_j, d_{ij} it is the case that $d_{ij} = |x_i - x_j|$ (line 7), so *Rel*(P,D) are satisfied. Notice there is no matter which x_j is taken in line 7,

```
1. Let D_0 < D_1 < ··· < D_m be the topological order
   from SCC_D;
2. Let P_1 < P_2 < ··· < P_q be the topological order
   from SCC_P;
3. for k = 0 to m do
4.   for each ij in D_k let d_ij := k;
5. for each i in P_1 let x_i := 1;
6. for k = 2 to q do
7.   for each i in P_k let x_i := x_j + d_ij /* such that j
     is in P_{k-1} */
8. for every x_i, x_j ∈ P do
9.   if not x_i R_ij x_j then return false;
10. return true;
```

Figure 2: Function Exist-Solution$(\Sigma^S_{PD}, SCC_P, SCC_D)$

due to the check we do in line 5 of ConsJSimple. Indeed, it must be $d_{ij} = d_{ij'}$ for each $j, j' \in P_{k-1}$. Finally we must prove if the instantiation we have calculated for point variables satisfies constraints in *Rel(P)* (lines 8 and 9). If the function returns true then a consistent instantiation has been found. Otherwise, no solution exists when the function returns false. Although the function Exist .Solution only return true or false we could easily modify the code to return also the solution tuple if it required.

Hence, Cons-Simple correctly check for consistency in a simple PDN and this is done in polynomial time. In fact, Cons-Simple is $O(d^2)$, where *d* is the number of duration variables. This time is due mainly to the cost of computing SCCs$G_{<,D}$ which $O(|V_D|+|E_D|)$ fwe use Tarjan's algorithm [1972]. Since $G_{<,D}$ is a complete graph $O(|V_D|+|E_D|) = O(d^2)$. The topological order in line 1 can be computed with a depth-first search on the directed acyclic graph with SCCD as the set of vertices, with cost $O(d^2)$ in the worst case. ■

Once we know how to determine consistency in a simple PDN, we could devise an algorithm for the same task with a general PDN. We have to examine each simple PDN extracted from the general one and apply ConsJSimple until one consistent scenario is found. Of course this algorithm is exponential in the worst case, when the network is inconsistent. Another exponential-time algorithm could be developed to find the minimal network using the result of the next theorem.

Theorem 3 *Given* $\Sigma_{PD} = \langle N_P, N_D, Rel(P,D)\rangle$, *the network* $\Sigma^T_{PD} = \langle N^T_P, N^T_D, Rel^T(P,D)\rangle$ *with the same* *variables and binary relations given by,*

$$R^T_{ij} = \bigcup R^S_{ij}$$
$$R^T_{ij,km} = \bigcup R^S_{ij,km}$$

where the union is over all the consistent scenarios Σ^S_{PD} *of* Σ_{PD}*, is the minimal network equivalent to* Σ_{PD}

We omit the proof since a similar one can be found in [Dechter *et* al., 1991]. This theorem shows we can obtain the minimal network Σ^M_{PD} and minimal binary relations by generating all the simple PDN from Σ_{PD}, checking for consistency with Cons-Simple and taking the union of primitive and feasible relations.

4 Conclusion

We have presented a qualitative temporal reasoning system that takes both points and durations as primitive objects and allows relative and indefinite information. We have formally defined a PDN as a structure formed by two interconnected PA networks. This allows us to borrow and adapt some of the concepts developed for the point algebra networks, such as consistency and minimality. We have proved that the problem of determining consistency in a PDN is NP-complete and a simpler and polynomial-time decision problem for a restricted kind of PD networks has been introduced which is useful for cheking consistency and finding minimal network in the general case.

Despite the intractability of reasoning tasks with general PD networks, we think these tasks may be useful in several areas such as scheduling and planning systems. The analysis of such systems requires the ability to specify and prove relations between critical states or actions and their durations. So, several strategies may be adopted to put this PD reasoning model to work in practical systems. One may be, for instance, to reach minimality both in *Np* and *ND* independently, and consecuently, accept its incompleteness. Or better, find how to restrict the information in a PDN so that we can obtain polynomial-time reasoning algorithms. The restricted model, however, may be expressive enough to work well in practice. Actually we are working in this direction and have found how restricting just the relations between points may lead to polinomial-time problems. We are also investigating how to integrate qualitative and metric information between points and durations and the possible aplications of these new models.

References

[Allen, 1983] J.F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM,* 26(II):832-843, 1983.

[Barber, 1993] F.A. Barber. A metric time-point and duration-based temporal model. *SIGART Bulletin,* 3:30-49, 1993.

[Dean and McDermott, 1987] T.L. Dean and D.V McDermott. Temporal data base management. *Art. Intel!.,* 32:1-55, 1987.

[Dechter *et* a/., 1991] R. Dechter, I. Meiri and J. Pearl. Temporal constraint networks. *Art. IntelL,* 49:61-95, 1991.

[Garey and Johnson, 1979] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, CA, 1979.

[Kautz and Ladkin, 1991] H. Kautz and P.B . Ladkin. Integrating metric and qualitative temporal reasoning. In *Proceedings of* AAAI-91, 1991.

[Meiri, 1991] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning.In *Proceedings of* AAAI- 91, pages 260-267, 1991.

[Meiri and Pearl, 1990] I. Meiri and J. Pearl. Faster constraint satisfaction algorithms for temporal reasoning. Tech. Rept. R-151, Computer Science Departament, University of California, Los Angeles, CA, 1990.

[Montanari, 1974] U. Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Information Science,* 7:95-132, 1974.

[Tarjan, 1972] R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.,* 1:745-750, 1972.

[Tarski, 194l] A. Tarski. On the calculus of relations. *J Symbolic Logic,* 6:73-89, 1941.

[Tsang, 1993] E. Tsang. *Fundattons of Constraint Satisfaction.* Academic Press, San Diego, CA, 1993.

[Van Beek, 1992] P. Van Beek. Reasoning about qualitative temporal information. *Art. Intell,* 58:297-326, 1992.

[Vilain and Kautz, 1986] M. Vilain and H.Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of* AAAI-86, pages 377-382, 1986.