

# Reasoning with Sequences of Point Events\*

## (An Extended Abstract)

R. Wetprasit and A. Sattar

School of Comp. and Infor. Tech.  
Griffith University  
Nathan, Brisbane, 4111  
AUSTRALIA

L. Khatib

Computer Science Program  
Florida Institute of Technology  
150 W. University Blvd.,  
Melbourne, Fl. 32901, USA

### Abstract

We propose to model recurring events as multi-point events by extending Vilain and Kautz's point algebra [7]. We then propose an exact algorithm (based on van Beek's exact algorithm) for finding feasible relations for multi-point event networks. The complexity of our method is compared with previously known results both for recurring and non-recurring events. We identify the special cases for which our multi-point based algorithm can find exact solution. Finally, we summarise our paper with brief discussion on ongoing and future research.

## 1 Introduction

In this paper, we consider the events that are sequences of time points as *multi-point events* by extending Vilain and Kautz's *point algebra*. Each multi-point event could be a finite sequence of recurring instantaneous actions, or a finite set of the beginning or ending of interval events that occur repeatedly. It is straightforward to transform a multi-point event network into corresponding point algebra network. We can then apply the existing PA algorithms to the reasoning tasks in questions. Intuitively, we believe that reasoning with multi-point event networks should have better performance than the traditional PA network because computation on implicit relations of the same multi-point event could be omitted.

Even though, the non-convex interval model [3] can represent and reason about the recurring interval events, obtaining an exact solution for interval-based problem is a hard problem which requires exponential time algorithms [1, 3]. However, if the numbers of subintervals are known and we restrict the relations between subintervals to pointizable relations (SIA) [6], by translating to multi-point event networks, our algorithm computes the exact solutions for the original non-convex interval networks in polynomial time.

\*A full version of this paper will appear in Proceedings of Canadian AI Conference 1996 (CSCSI-96).

## 2 Multi-Point Events and Their Relations

A *multi-point event* (MPE) is a collection of points, when each point represents the related subevents. A MPE is in *normal form* if all points are totally ordered, or the relation between the  $i$ th point to the  $i + 1$ th point of the MPE is ' $<$ '.

Given two MPEs:  $I$  of size  $n$  and  $J$  of size  $m$ ,  $R(I, J)$  is the *multi-point relation* (MPR) between MPE  $I$  and  $J$ . An element  $R(I_i, J_j)$  is a disjunction of point relation defined in *point algebra* [7] representing the relationship between the  $i$ th point of  $I$  and the  $j$ th point of  $J$ . For computational purpose,  $R(I, J)$  is represented by using a  $A_{n \times m}$  matrix relation, when the rows of matrix represent the points of  $I$  and the columns represent the points of  $J$ .

Solving reasoning tasks for multi-point events in our framework is based on constraint satisfaction techniques. A binary constraint network of  $k$  MPEs, each contains  $n$  point subevents, consists of  $k$  nodes where each node represents an individual MPE. The domain of each node is a set of real numbers each representing a point subevent:  $D_i = \{a_1, a_2, \dots, a_n : a_j < a_{j+1}\}$ . The labels on the arcs are matrix relations representing constraints between two nodes. An **instantiation** of a node is a  $k$ -tuple  $(x_1, x_2, \dots, x_k)$  when  $x_i \in D_i$ . The **minimal label** or the matrix of feasible relations between two MPEs is the matrix in which each element is consisting of all and only the feasible relations. A **scenario** is a set of atomic relations between pairs of MPEs. Each atomic relation corresponds to a matrix label for each arc.

## 3 Reasoning with MPE Networks

The path-consistency checking for MPE network is done in two levels: between two MPEs and three MPEs. The *canonical form* is defined to ensure path-consistency between three points being in two different MPEs. The path-consistency between three MPEs obtains from the *composition operations* over two matrix relations (defined in Section 3.2).

### 3.1 Canonical Form

A matrix relation  $A_{n \times m}$  is said to be in the canonical form if the path-consistency conditions [5] among their neighbours as following are satisfied (more detail of the algorithm for converting a matrix into the canonical form is in [3]):

1.  $A_{i,j-1} \subseteq A_{i,j} \circ > (\forall 1 \leq i \leq n, 1 < j \leq m)$
2.  $A_{i+1,j} \subseteq > \circ A_{i,j} (\forall 1 \leq i < n, 1 \leq j \leq m)$
3.  $A_{i,j+1} \subseteq A_{i,j} \circ < (\forall 1 \leq i \leq n, 1 \leq j < m)$
4.  $A_{i-1,j} \subseteq < \circ A_{i,j} (\forall 1 < i \leq n, 1 \leq j \leq m)$

These conditions could be used to approximate the path-consistency of the MPE network. In our algorithm for finding feasible relations, we first transform the given matrices into canonical form. This is to reduce the elements in the matrices and also maintain the path-consistency between two MPEs.

**Lemma 1** *Given a MPE network with  $k$  MPEs ( $k \geq 1$ ), each contains  $n$  points ( $n \geq 2$ ). The MPE network has a corresponding PA network.*

**Theorem 2** *Suppose a MPE network has two nodes  $I$  and  $J$ . Let  $A$  be the matrix relation between  $I$  and  $J$ . If  $A$  is in canonical form, then the corresponding PA network is path-consistent.*

### 3.2 Matrix Relations Operations

The operations on two matrix relations, which are necessary in solving reasoning tasks, are defined by adopting the operations on constraint matrices from [4]. Table 1 defines the result of matrix operations ( $\oplus$ ,  $\otimes$ ,  $\hat{\cdot}$ ,  $\odot$ , and  $\circ$ ) on two MPRs in terms of basic operations in point algebra. The symbols  $+$ ,  $\times$ ,  $-$ ,  $\circ$ , and  $\sim$  correspond to union, intersection, complement, composition and inverse operators defined in PA, respectively.

|              |  |
|--------------|--|
| Union        | $C = A \oplus B$ iff $C_{i,j} = A_{i,j} + B_{i,j}$   |
| Intersection | $C = A \otimes B$ iff $C_{i,j} = A_{i,j} \times B_{i,j}$   |
| Complement   | $C = \hat{A}$ iff $C_{i,j} = \sim A_{i,j}$   |
| Inverse      | $C = \hat{A}$ iff $C_{i,j} = \sim A_{j,i}$   |
| Composition  | $C = A \odot B$ iff<br>$C_{i,j} = \prod_{k=1}^r (A_{i,k} \circ B_{k,j})$<br>$= A_{i,1} \circ B_{1,j} \times A_{i,2} \circ B_{2,j}$<br>$\times \dots \times A_{i,r} \circ B_{r,j}$<br>$\forall i, j (1 \leq i \leq n, 1 \leq j \leq m)$ |

Table 1: The Multi-Point Event Operations

### 3.3 The Algorithm

The algorithm to solve the minimal labels problem for MPE network is based on van Beek's exact algorithm for PA network [6]. He has shown that the path-consistency algorithm alone is not sufficient for PA network by pointing out a counter-example consisting

of four vertices called a *forbidden subgraph*. We define the forbidden subgraph for MPE network in terms of points in MPEs as follow:

**Definition 3 (A forbidden subgraph)** *Given a MPE subgraph of any four nodes:  $V, W, S$ , and  $T$ . Let  $V$  be  $(V_1 < \dots < V_m)$ ,  $W$  be  $(W_1 < \dots < W_n)$ ,  $S$  be  $(S_1 < \dots < S_o)$ , and  $T$  be  $(T_1 < \dots < T_p)$ , where  $m, n, o$  and  $p$  are the numbers of points in  $V, W, S$  and  $T$  respectively. The subgraph is called a forbidden MPE subgraph, if there exist  $v, w, s$ , and  $t$ , which are any valid indices of  $V, W, S$ , and  $T$ , such that the following conditions are satisfied:*

$$\begin{aligned} R(V_v, W_w) &= \neq, \\ R(V_v, S_s) &= \geq, \\ R(W_w, S_s) &= \leq, \\ R(V_v, T_t) &= <, \\ R(W_w, T_t) &= <, \\ R(S_s, T_t) &= \leq. \end{aligned}$$

The infeasible relation in the forbidden MPE subgraph is  $R(S_s, T_t)$ , which is ' $\leq$ '. This relation causes inconsistency among those four nodes. However, if we don't allow ' $=$ ' between  $S_s$  and  $T_t$ , the subgraph becomes 4-consistent [2].

If  $S$  is identical to  $T$ , the relation between  $S_s$  and  $T_t$  must be ' $<$ ' because all MPEs are in normal form (either ' $<$ ' or ' $>$ ' allowed between points in MPE). Thus, we don't need to take this case into consideration. Similarly, we can show that other pairs of MPE nodes in the forbidden subgraph cannot be identical. Therefore, the set of four MPE nodes considered as a forbidden subgraph are mutually different.

The following algorithm using the canonical form matrices and the matrix operations to solve the minimal label problem in polynomial time. We found that we do not need to check the path-consistency again after performing the procedure FIND.SUBGRAPHS\_MPE since the network is still path-consistent.

#### Algorithm FEASIBLE\_MPE

**Input:** A MPE network represented as a matrix  $C$  where entry  $C_{IJ}$  is the label on the arc from nodes  $I$  to  $J$ . Each  $C_{IJ}$  is a matrix relation  $R[I, J]$ , where an entry of  $R[I, J]$  is an internal relation between points in MPEs  $I$  and  $J$

**Output:** The set of feasible relations for  $C_{IJ}$ ,  $I, J = 1, 2, \dots, k$

```
begin
  PATH_CONSISTENCY_MPE
  FIND_SUBGRAPHS_MPE
end
```

#### Procedure PATH\_CONSISTENCY\_MPE

```
begin
  For each matrix relation  $R[I, J]$  do
    Canonical_Conv( $R[I, J]$ )
   $Q := \{(I, K, J) \mid 1 \leq I < J \leq k, 1 \leq K \leq k, K \neq I, J\}$ 
  While  $Q$  is not empty do
    select and delete a path  $(I, K, J)$  from  $Q$ 
```

```

Temp := R[I, J]  $\otimes$  (R[I, K]  $\odot$  R[K, J])
If (Temp  $\neq$  R[I, J]) then
    Canonical_Conv(Temp)
    R[I, J] := Temp
    R[J, I] := Temp (inverse of Temp)
    Q := Q  $\cup$  RELATED_PATHS(I, J)
end

Procedure RELATED_PATHS(I, J)
    Return {(I, J, K), (K, I, J) |  $1 \leq K \leq k$ ,
        K  $\neq$  I, K  $\neq$  J}

Procedure FIND_SUBGRAPHS_MPE
begin
    For each matrix relation such that
        R[V, W]v,w = ' $\neq$ ' ( $1 \leq V < W \leq k$ )
        and ( $1 \leq v, w \leq n$ ) do
        Initialize P1, P2, P, Q1, Q2, Q to empty set
        For each MPE K ( $1 \leq K \leq k$ , K  $\neq$  V, W) do
            P1 = P1  $\cup$  adj_MPEs( $\geq$ , Vv, K)
            P2 = P2  $\cup$  adj_MPEs( $\geq$ , Ww, K)
            Q1 = Q1  $\cup$  adj_MPEs( $\leq$ , Vv, K)
            Q2 = Q2  $\cup$  adj_MPEs( $\leq$ , Ww, K)
        P = P1  $\cap$  P2, Q = Q1  $\cap$  Q2
        For each Ss  $\in$  P do
            For each Tt  $\in$  Q do
                If S  $\neq$  T then
                    R[S, T]s,t := '<'
                    R[T, S]t,s := '>'
end

```

Our algorithm for computing feasible relations consists of two main tasks: checking path-consistency between three MPEs throughout the network (procedure PATH\_CONSISTENCY\_MPE), and eliminating the infeasible relations from the forbidden subgraphs (procedure FIND\_SUBGRAPHS\_MPE). We maintain the consistency between two MPEs by calling procedure Canonical\_Conv (detailed in [3]). This procedure transforms the matrix relations into canonical form before inserting to the database. Function *adj\_MPEs*( $\geq$ , V<sub>v</sub>, K) in procedure FIND\_SUBGRAPHS\_MPE returns the set of elements K<sub>k</sub>, in which R(V<sub>v</sub>, K<sub>k</sub>) = ' $\geq$ ' by checking the relations between V<sub>v</sub> and K<sub>i</sub>, ( $1 \leq i \leq n$ ). Here are some technical results:

**Lemma 4** Changing the label R(S<sub>s</sub>, T<sub>t</sub>) of the forbidden subgraph defined in Definition 3 will not lead to path inconsistency.

**Theorem 5** The closure of a MPE network, calculated by the algorithm FEASIBLE\_MPE, corresponds to a path consistent PA network.

**Theorem 6** The algorithm FEASIBLE\_MPE correctly finds the minimal labels for all internal relations in the MPE network.

**Theorem 7** The algorithm FEASIBLE\_MPE, for k MPE nodes in the constraint network and each node contains at most n points, has a time complexity of  $O(\max(n^5 k^3, mn^2 k^2))$ , where there are m  $\neq$ -relations.

## 4 Conclusion

The main contribution of this framework is an extension of point-based representation to reason with the recurring events that are considered as collections of point subevents. The algorithm we proposed correctly finds all feasible relations in the MPE network. The complexity of this algorithm is  $O(\max(n^5 k^3, mn^2 k^2))$ , where k is the number of MPEs with maximum n points and m is the number of ' $\neq$ ' internal relations in the network. The complexity for finding the same solutions for non-recurring PA network with the same data (nk point events) is  $O(\max(n^3 k^3, mn^2 k^2))$ , where m is also the number of ' $\neq$ ' relations in the network [6].

For non-convex interval network, the minimal labels of the network can be approximately achieved in  $O(n^5 k^3)$  complexity algorithm, or  $O(n^3 k^3)$  when considering only the affected relations [3]. However, if we restrict the internal relations between subintervals to be pointizable interval relations (SIA) [6] and transform into MPE network, our algorithm yields the exact solutions. The complexity of our algorithm is proportional to the number of ' $\neq$ ' edges, the worst case would seldom occur in the real world domain.

Currently, we are working on improving complexity of the algorithm we presented in this paper by avoiding the redundant computation of the unconstrained relations, and on constructing the algorithm for finding a consistent scenario for the MPE network. We are implementing our algorithms and comparing the performance to other non-multi-point event approaches.

## References

- [1] J. Allen, "Maintaining Knowledge about Temporal Intervals," *Communication of the ACM*, Vol. 26:11, pp. 832-843, 1983.
- [2] E.C. Freuder, "A Sufficient Condition for Backtrack-Free Search," *Journal of ACM*. Vol. 29, pp. 24-32, 1982.
- [3] L. al-Khatib, *Reasoning with Non-Convex Time Intervals*. PhD dissertation, Florida Institute of Technology, Melbourne, Florida. 1994.
- [4] P. Ladkin and R.D. Maddux, "On Binary Constraint Problems," *Journal of ACM*. Vol. 41:3, pp. 435-409, 1994.
- [5] A.K. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence*. Vol. 8, pp. 99-118, 1977.
- [6] P. van Beek, *Exact and Approximate Reasoning about Qualitative Temporal Relations*. Technical Report TR-90-29, University of Alberta, Edmonton, Alberta, Canada. 1990.
- [7] M. Vilain and H. Kautz, "Constraint Propagation Algorithms for Temporal Reasoning," *Proceedings of AAAI-86*, San Mateo; Morgan Kaufman. 1986.