Transforming Timed Influence Nets into Time Sliced Bayesian Networks

Sajjad Haider
Abbas K. Zaidi

The 2004 Command and Control Research and Technology Symposium
June 15 – 17, 2004
San Diego, CA

System Architectures Lab
George Mason University

Outline

- Introduction of Time Sliced Bayesian Networks and Timed Influence Nets
- Need for the transformation algorithm.
- Explanation of the transformation algorithm
- Application of the algorithm
- Conclusions and future research direction

Advantages of Probabilistic Belief Networks

64 probability values are required to represent the joint distribution of 6 binary state variables, i.e., $2^6 = 64$

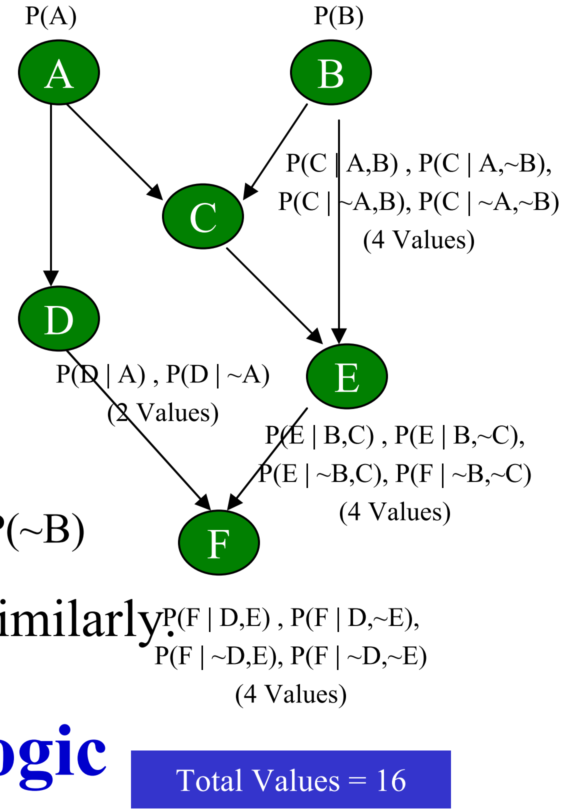
Probabilistic Network representations can reduce this number significantly

The joint distribution is computed as

P(A,B,C,D,E,F) = P(F | D,E)P(D | A)P(E | B,C)P(C | A,B)P(A)P(B)

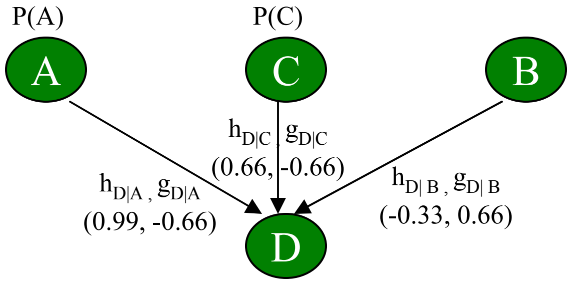
P(A,B,~C,~D, E,F) = P(F | ~D,E)P(~D | A)P(E | ~B,C)P(C | A,~B)P(A)P(~B)

Probabilities for other 62 combinations can be found out similarly.



CAusal STrength (CAST) Logic

- Inputs have ranges from –1 to 1.
- $h_{D|A}$ is analogous (but not equal) to P(D | A) while $g_{D|A}$ is analogous (but not equal) to P(D | ~A).



System Architectures Lab
George Mason University

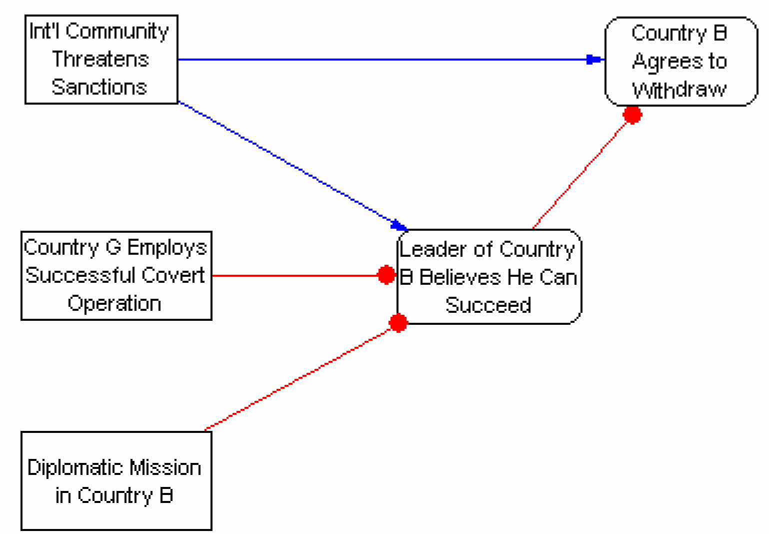
Influence Nets

Probabilistic Belief Networks that use CAST Logic for model specification are termed as Influence Nets.

The current implementation of Influence Nets assume that the parents of a node are marginally independent.

Root Nodes　　　Non-root Nodes

Root Nodes typically represents actionable events



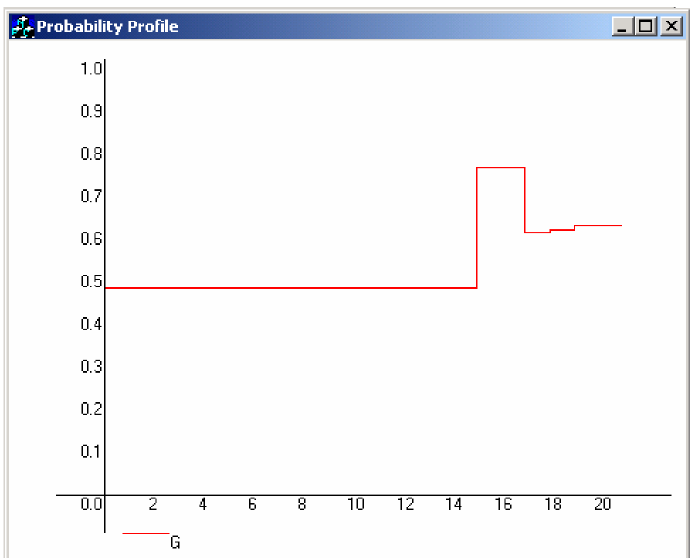
Positive Impact

Negative Impact

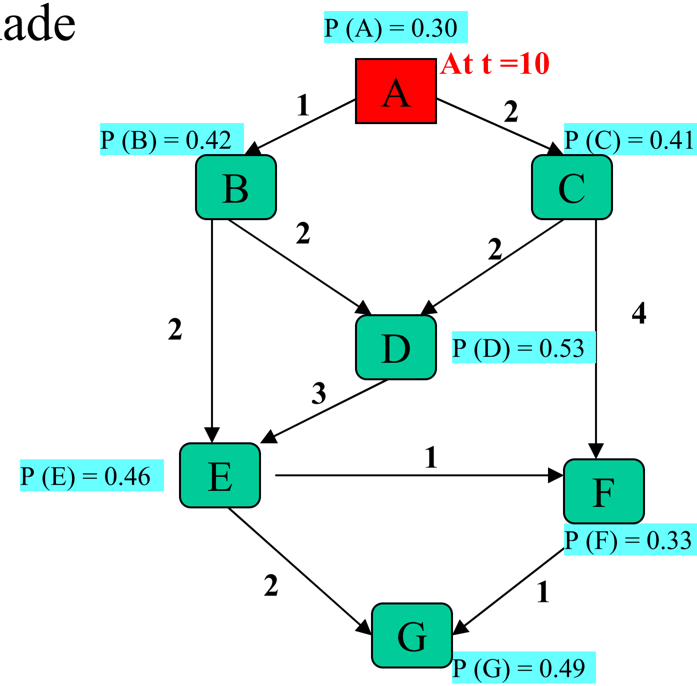
Timed Influence Nets

Timed Influence Nets have following additional parameters

- A time delay is associated with each arc.
- A time delay is associated with each node.
- Each actionable event is assigned time stamp(s) at which the decision(s) regarding the state of that action is(are) made



0	0.488
15	0.771
17	0.616
18	0.624
19	0.632



A: t = 10 B: t = 11 C: t = 12 D: t = 13, 14 E: t = 13, 16, 17

F: t = 14, 16, 17, 18 G: t = 15, 17, 18, 19

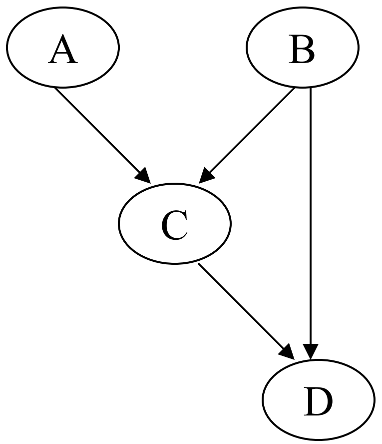
Time Sliced Bayesian Network

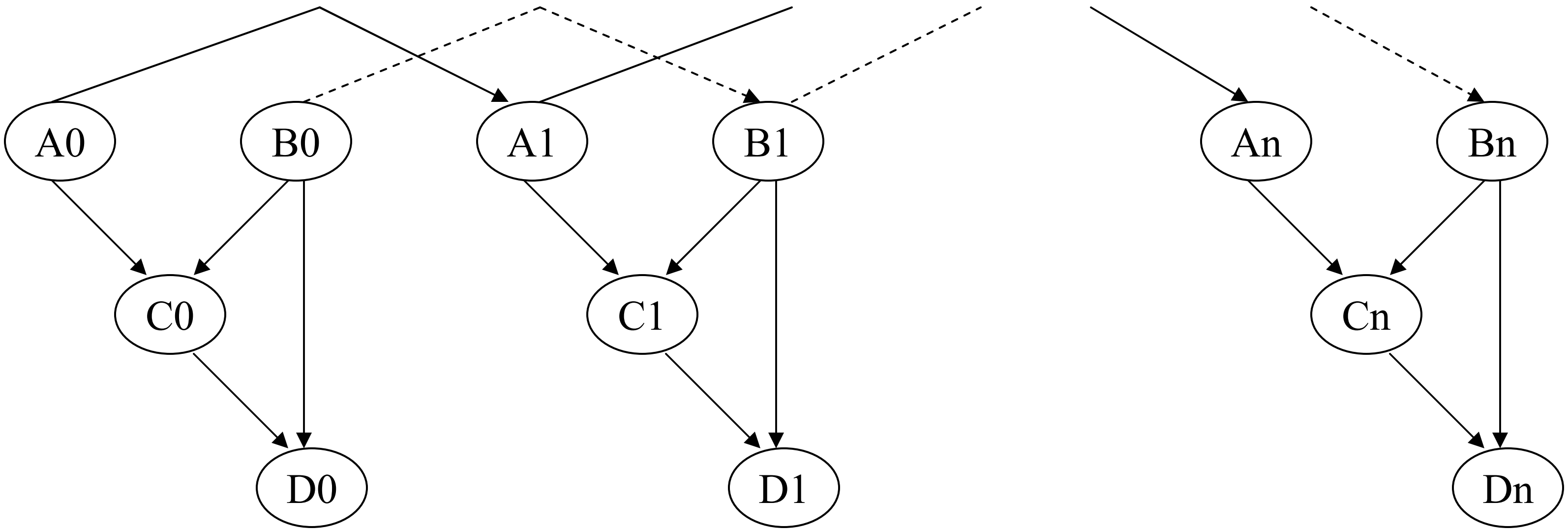
The text book approach of building TSBN is

Specify a static Bayesian Network

Specify the temporal dependencies that exist between the nodes in different time slices.

Unroll the model for a given number of time slices.

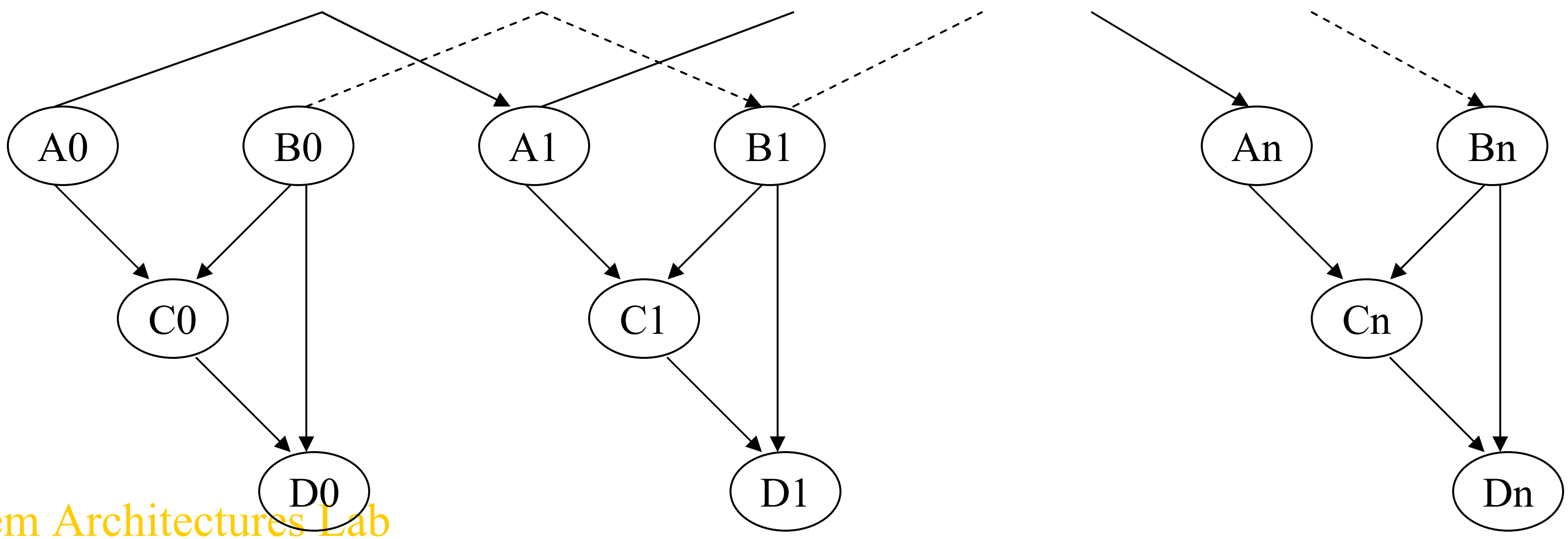




Time Sliced Bayesian Network (Cont..)

A TSBN is a Bayesian Network where

- Nodes are indexed by time
- Time is integer valued and begins at zero
- The local distribution for a variable can depend on
  - any variable that precedes it in time
  - variables at the same time that are prior to it in node ordering



Motivation

Advantages of TINs

- Simpler knowledge elicitation
- Courses of Actions (COA) specification is easier
- Model Reading is easier

Disadvantages of TINs

- Limited capability for execution monitoring

Advantages of TSBNs

- Variety of algorithms available for execution monitoring

Disadvantages of TSBNs

- Knowledge elicitation is intractable
- Model Reading is not easy

Pre-Processing of TIN

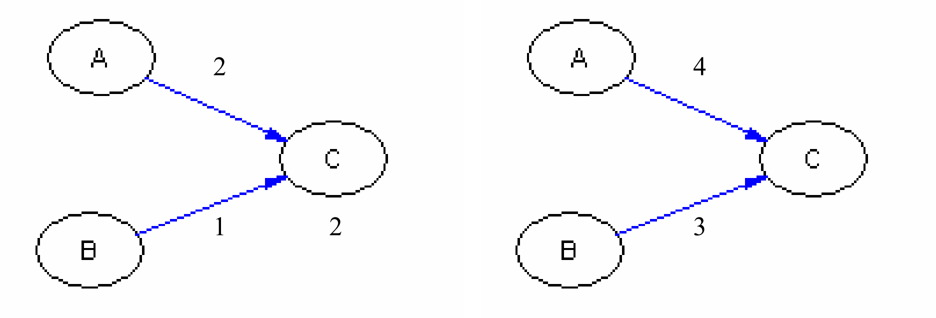
Let TIN = (V, E, P, D_V, D_E, A) where

V: Set of Nodes E: Set of Edges P: Joint Distribution of Random Variables

D_V: Delay Associated with Nodes D_E: Delay Associated with Arcs

A: Input Scenario

Before executing the transformation algorithm the delays on the nodes are reassigned to the arcs



TIN = (V, E, P, D_V, D_E, A) → TIN = (V, E, P, D, A)

The Transformation Algorithm

Given TIN = (**V**, **E**, **P**, **D**, **A**)

1. Find the maximum path length between the root nodes and target nodes, i.e.,

M = $\max_{i,j}[P_{i,j}]$ where

$P_{i,j}$: path between nodes i and j such that i, k $\in$ V and $\neg\exists$ (k, i) $\in$ E

2. Construct a TSBN (**V1, E1, P1**) where

V1: $\forall\, v \in$ **V** add v_i to **V1** where i = 0, 1, …, M

= $\{v_i \mid v \in$ **V**, i = 0, 1, …., M$\}$

E1 = $\{(x_i, y_j) \mid$ i = max (0, j – D(x, y)); x, y $\in$ **V** and i, j = 0, 1, …., M$\}$

P1: **P** when indices are ignored

For example, $P(y_i \mid x_i) = P(y \mid x)$ when x, y $\in$ **V** and x_i, $y_j \in$ **V1**.

This step draws the nodes in the TSBN for M time slices. The connections are drawn between the non-root nodes and their parents. The following step is required once an input scenario is determined.

3. Let S = maximum time stamp associated with the root nodes as provided by the input scenario:

(Add S additional time slices in the TSBN obtained in the previous step by following the procedure outlined in Step 2.

(The resultant network is the modified TSBN (**V1, E1, P1**) where

V1 = $\{v_i \mid v \in$ **V**, i = 0, 1, …., M+S$\}$

E1 = $\{(x_i, y_j) \mid$ i = max (0, j – D(x, y)); x, y $\in$ **V** and i, j = 0, 1, …., M+S $\}$

P1: **P** when indices are ignored

(Let **R1** = Set of Root Nodes where **R1** $\subset$ **V1**. $\forall\, r \in$ **R1** connect r_{t-1} to r_t where t = 1, 2, …, M+S, unless t is the time at which the variable is set to a state.

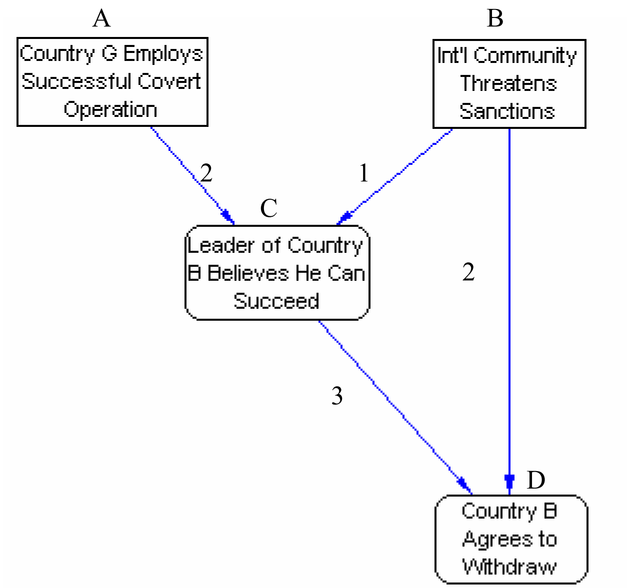
Step 1 Of the Algorithm

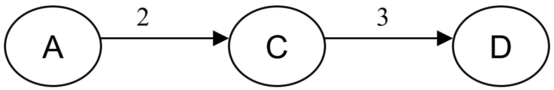
Given TIN = (**V, E, P, D, A**)

1. Find the maximum path length between the root nodes and target nodes, i.e.,

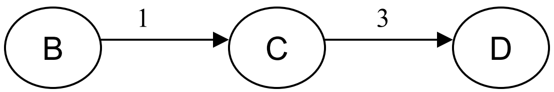
$M = \max_{i,j}[P_{i,j}]$ where

$P_{i,j}$: path between nodes i and j such that i, k $\in$ V and $\neg\exists$ (k, i) $\in$ E

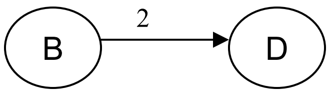




Path Length: 5



Path Length: 4



Path Length: 2

Step 2 Of the Algorithm

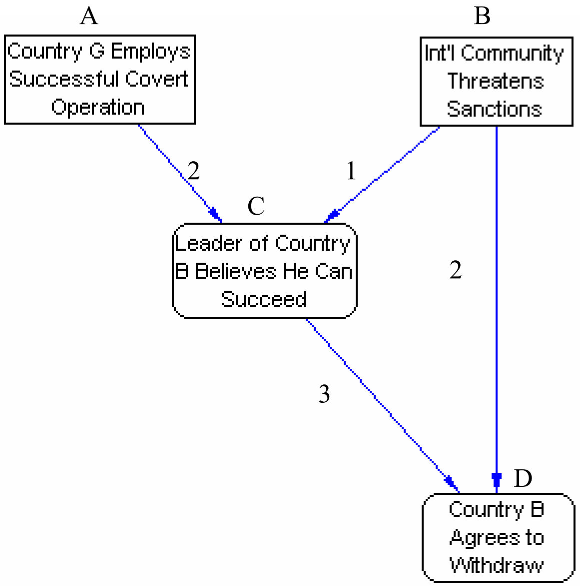
2. Construct a TSBN (**V1, E1, P1**) where

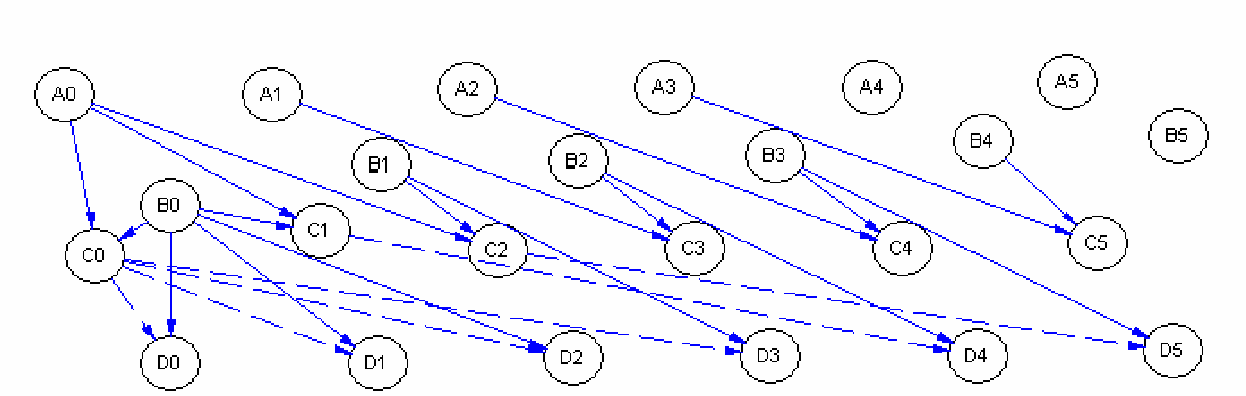
V1: $\forall\ \nu \in$ **V** add ν_i to **V1** where i = 0, 1, …, M

= $\{\nu_i \mid \nu \in$ **V**, i = 0, 1, …., M$\}$

E1 = $\{(x_i, y_j) \mid i = \max(0, j - D(x, y));\ x, y \in$ **V** and i, j = 0, 1, …., M$\}$

P1: **P** when indices are ignored





Step 3 Of the Algorithm

3. Let S = maximum time stamp associated with the root nodes as provided by the input scenario:

(a) Add S additional time slices in the TSBN obtained in the previous step by following the procedure outlined in Step 2.

(b) The resultant network is the modified TSBN (**V1, E1, P1**) where

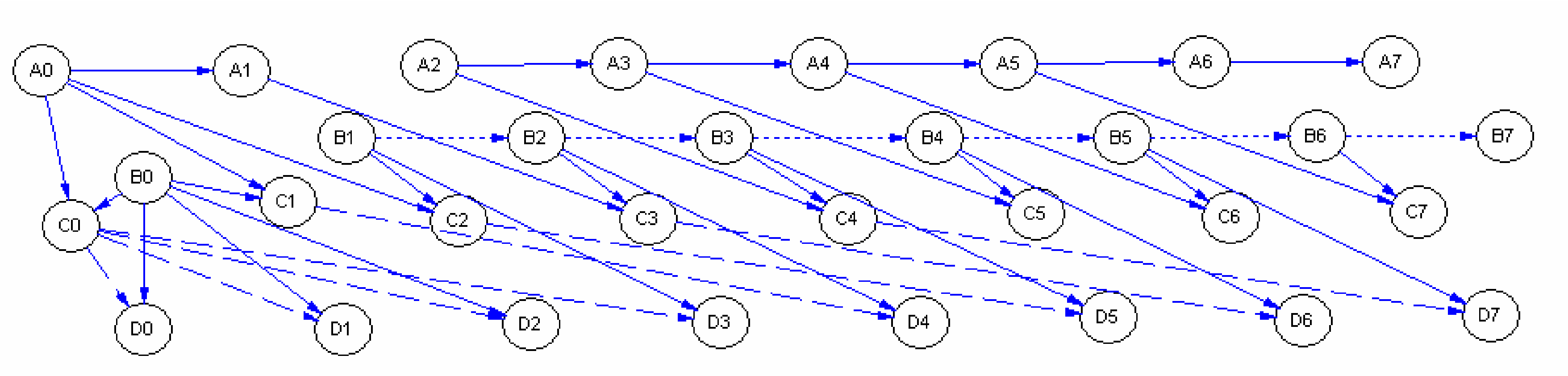
V1 = $\{v_i \mid v \in$ **V**, i = 0, 1, …., M+S$\}$

E1 = $\{(x_i, y_j) \mid i = \max(0, j - D(x, y)); x, y \in$ **V** and i, j = 0, 1, …., M+S $\}$

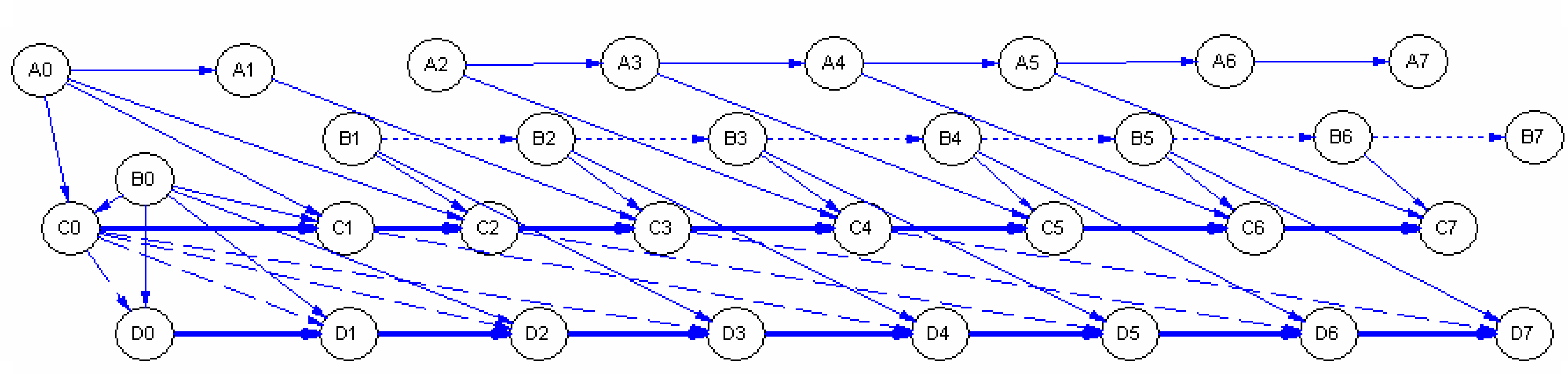
P1: **P** when indices are ignored

(c) Let **R1** = Set of Root Nodes where **R1** $\subset$ **V1**. $\forall r \in$ **R1** connect r_{t-1} to r_t where t = 1, 2, …, M+S, unless t is the time at which the variable is set to a state.

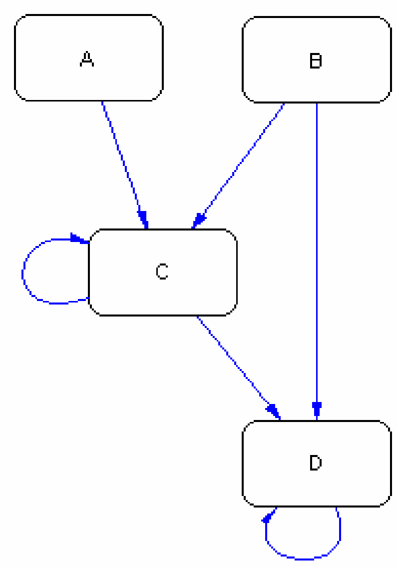
Let A is taken at time 2 while B is taken at time 1



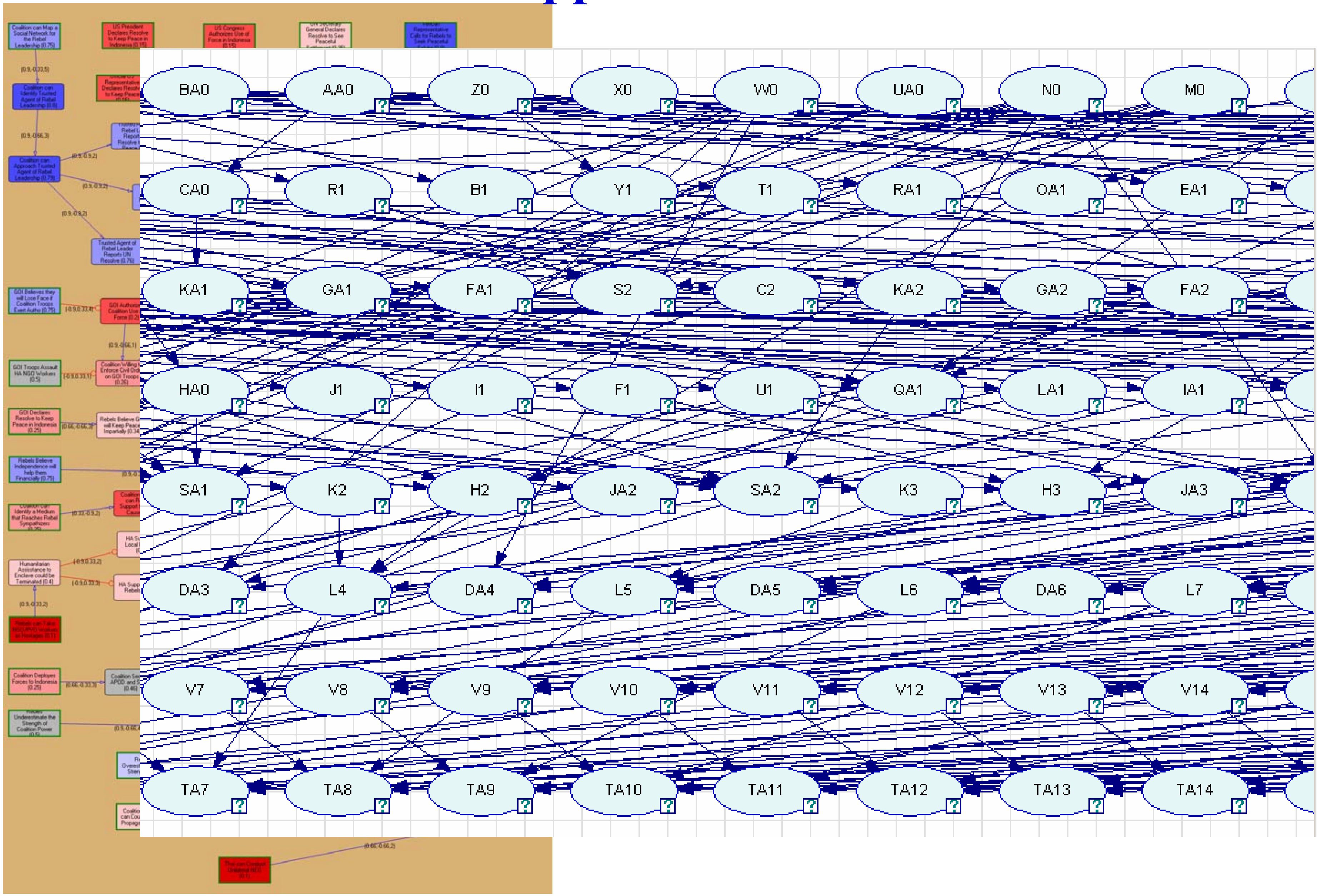
Temporal Dependencies Among Non-Root Nodes



A self-loop can be added to the Influence Net that represents the dependency among a non-root node (or represents memory)



Application



Conclusions

- An algorithm is presented that transforms TINs into TSBNs.
- TINs are easier to use for
  - ❑Knowledge elicitation
  - ❑COA specification
  - ❑Understanding model
- TSBNs are good for belief updating.
- The idea is to use
  - ❑TINs as a front end tool for model building and course of action selection.
  - ❑Real-Time execution monitoring is accomplished by using the TSBN as a back end tool.