On the Partial Observability of Temporal Uncertainty

Michael D. Moffitt

AAAI 2007

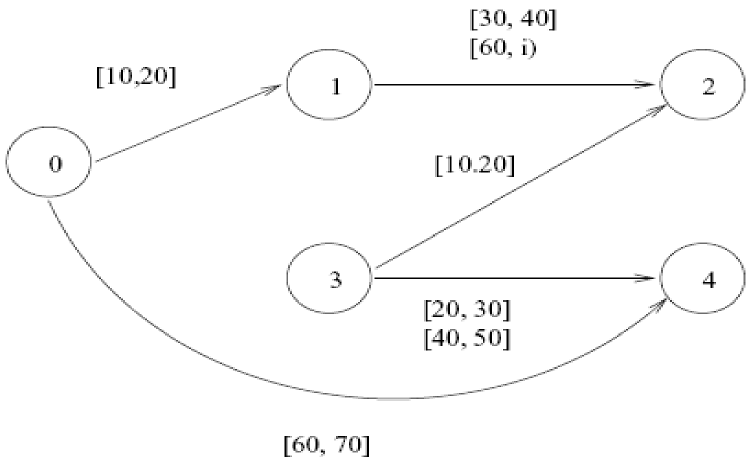
Outline

- Introduction
- Background
- Shared Temporal Causality
- The Partially Observable STPU
- Algorithms for Dynamic Controllability
- Conclusion and Future Work

Introduction

- Uncertainty in constraint-based temporal reasoning: time points are divided into controllable and uncontrollable events.
- Prior studies of uncertainty in Temporal CSPs have required a direct correspondence between observation of an event and its actual execution.
- The author propose an extension to the Simple Temporal Problem with Uncertainty, in which the agent is made aware of only a subset of uncontrollable time points.

Background 1/5
Simple Temporal Problem (STP)



The "Jon and Fred travel to work" Example

- **Definition**: a pair <X, E>, X_i: time point, E_{ij}: constraint
- **Graph-based encoding**
- **Consistency**: no negative cycles
- **Algorithm**: Floyd-Warshall, polynomial time

Background 2/5
The running example

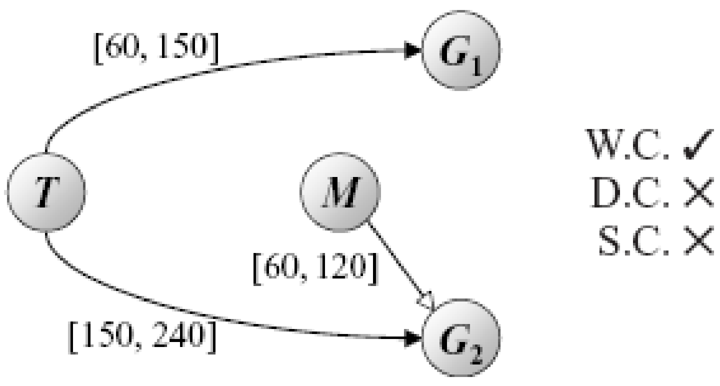


Figure 1: The network corresponding to Example 1.

T: current time

G1: guest 1 arrive

G2: guest 2 arrive

M: medication

Example 1: Mrs. Smith is expecting two family members to visit for dinner. The first guest will arrive in 1 to 2.5 hrs; the second guest will arrive in 2.5 to 4 hrs. Mrs. Smith must take medication 1 to 2 hrs prior to dinner.

Background 3/5

Simple Temporal Problems with Uncertainty (STPU)

- Extension to STP to deal with uncertainty
- Defined by < Xc, Xu, E, C>
- **Xc**: controllable time points, M

  Xu: uncontrollable time points, G1, G2
- **E**: requirement links, M->G2
- **C**: contingent links, E->G1, E->G2

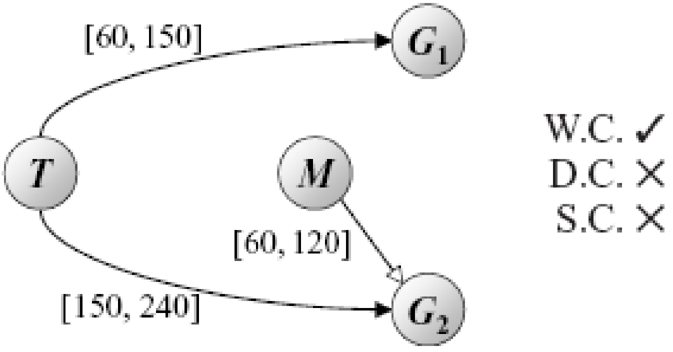


Figure 1: The network corresponding to Example 1.

Background 4/5
Controllability of STPU

- **Weak Controllability**: for every possible projection, there must exist a consistent solution.
- **Strong Controllability**: there exists a single consistent solution that satisfies every possible projection.
- **Dynamic Controllability**: there exists a strategy that depends on the outcomes of only those uncontrollable events occurred in the past.

Dynamic controllability is computable in $O(N^4)$-time. (Morris 2006)

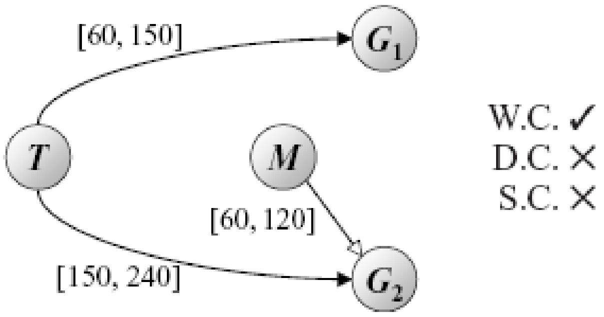


Figure 1: The network corresponding to Example 1.

Shared Temporal Causality 1/3

- **Example 2**: Uncertainty was due to unknown traffic conditions. Considering the routes both guests take, we can expect the second guest arrive between 1.5 to 2.5 hrs after the first.

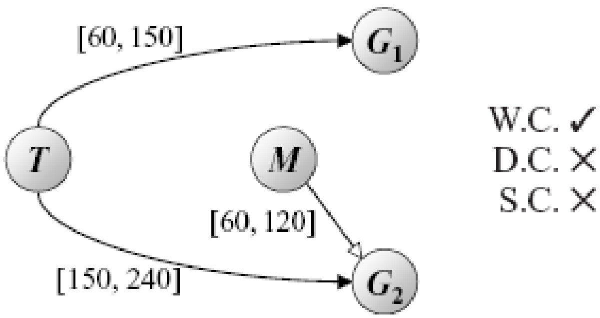


Figure 1: The network corresponding to Example 1.

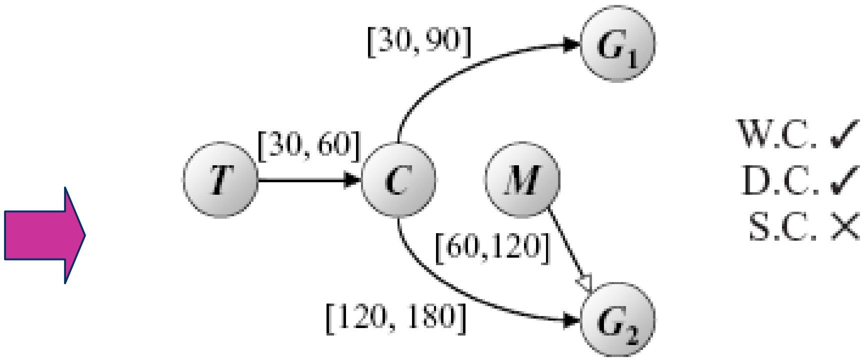


Figure 2: A common causal process has been factored into the contingent link $T \Rightarrow C$.

T→C: a common contingent process

Shared Temporal Causality 2/3

- By analysis of the subnetwork (C, M, G2), we can infer a lower and upper bound on C->M: [ub(G2-C)-ub(G2-M), lb(G2-C)-lb(G2-M)]=[60,60]
- So Example 2 is dynamically controllable with the strategy M=C+60.
- A subtle fault: Our plan cannot depend on C, since C's corresponding causal process is hidden from view.

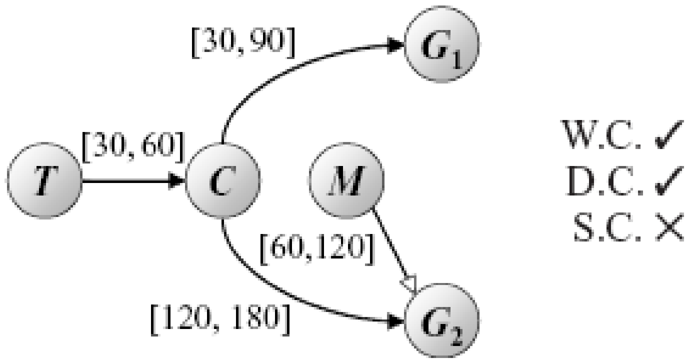


Figure 2: A common causal process has been factored into the contingent link $T \Rightarrow C$.

Shared Temporal Causality 3/3

- Another Hidden Temporal Causality example:

  Deep Space One (DS1) spacecraft controlled by the New Millennium Remote Agent (NUMA)—one of the earliest applications of temporal reasoning with uncertainty.

  (Muscettola, 1998)

- Need to seek a relaxation to the STPU that can accommodate partial observability.

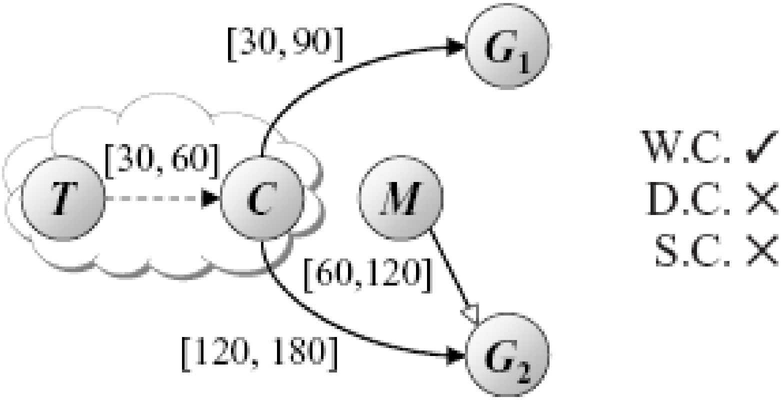
Partially Observable STPU 1/2

- Extension to the STPU to deal with partial observablility.
- Defined as <Xc, Xo, Xu, E, Co, Cu>
- **Xc**: controllable

  Xo: observable uncontrollable

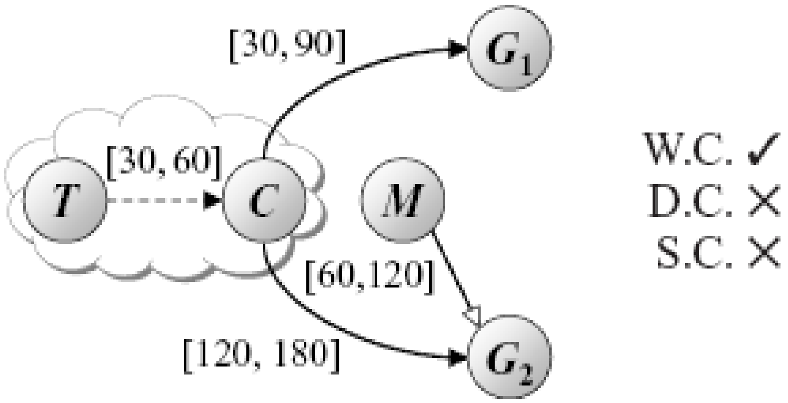
  Xu: unobservable uncontrollable
- **E**: requirement links
- **Co**:observable contingent links

  Cu:unobservable contingent links



Partially Observable STPU 2/2

A Partially Observable STPU is Dynamically Controllable if there exists a strategy that depends on the outcomes of only those uncontrollable, observable events that have occurred in the past.



Algorithms for Dynamic Controllability
- Original Reduction Rules

- The labeled distance graph characterized for STPU.
- Tightening of edges is achieved by a set of reduction rules.
- A strongly polynomial-time algorithm for dynamic controllability is obtained by repeatedly applying these rules until a certain cutoff is reached.

(Upper-Case Reduction)
$$A \xleftarrow{B:x} C \xleftarrow{y} D \text{ adds } A \xleftarrow{B:(x+y)} D$$

(Lower-Case Reduction) If $x < 0$,
$$A \xleftarrow{x} C \xleftarrow{c:y} D \text{ adds } A \xleftarrow{x+y} D$$

(Cross-Case Reduction) If $x < 0$, $B \neq C$,
$$A \xleftarrow{B:x} C \xleftarrow{c:y} D \text{ adds } A \xleftarrow{B:(x+y)} D$$

(No-Case Reduction)
$$A \xleftarrow{x} C \xleftarrow{y} D \text{ adds } A \xleftarrow{x+y} D$$

Algorithms for Dynamic Controllability
- Augmented Reduction Rules 1/3

Must-Precede($A \xleftarrow{x} C$) // *for STPU*

1. If ($x < 0$) *return true;*
2. *return false;*

Figure 5: Returns *true* if C must execute after A (*false* otherwise).

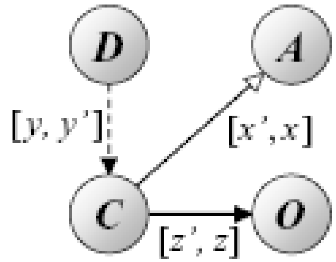
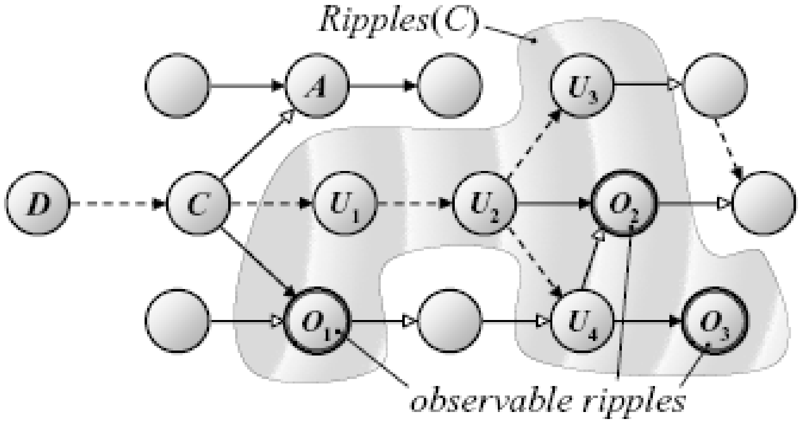


Figure 6

- O makes C sufficiently Observable to A by two conditions:
  1) O must be sufficiently punctual (z<=x): it must be always possible for A to occur during or before O.
  2) O must be sufficiently informative (z-z’<=x-x’): the width of the interval on C->O must be no greater than the width of interval on C->A.
- Theorem: If C is sufficiently observable to A via O, the subnetwork is locally controllable. (i.e., we can dynamically determine a value for A following O that satisfies the requirement link C->A)

Algorithms for Dynamic Controllability
- Augmented Reduction Rules 2/3



- Definition : Given an unobservable, uncontrollable event C, we define Ripples(C) as the set of uncontrollables that lie at the conclusion of a contingent link that begins with either C or another unobservable uncontrollable in Ripples(C).

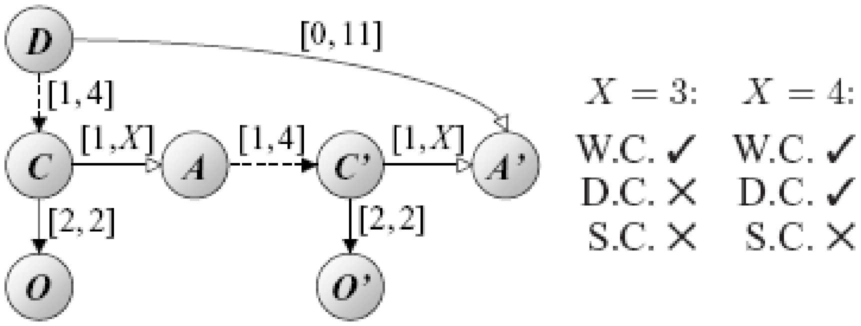
Algorithms for Dynamic Controllability
- Augmented Reduction Rules 3/3

Must-Precede-Observation($A \overset{x}{\leftarrow} C$) // *for POSTPU*

1. If ($x < 0$) *return true;*
2. If ($C \in X_U$ and $A \notin Ripples(C)$)
3.   For each observable uncontrollable $O \in Ripples(C)$
4.   If ($x < z$) *next* O;
5.   If ($x - x' < z - z'$) *next* O;
6.   *return false;*
7.   *return true;*
8. *return false;*

- Replace the function Must-Precede()
- Line 2: If C is an unobservable uncontrollable and event A will not be subsequently scheduled by nature,
- Line 3-5: examine each observation point O in Ripples(C) to check if it is sufficiently punctual and sufficiently informative.
- Line 6,7: If both conditions are met, then sufficient observability is ensured.

Algorithms for Dynamic Controllability
- Incompleteness and an Open Problem



- A network where sufficient observability does not necessarily guarantee global dynamic controllability.

- When X=3: if both C and C' occurs as late as possible, waiting for observables require A' to execute no earlier than 4+2+4+2=12 units after D, violating upper bound of 11.

Conclusion

- A new formalism—Partially Observable STPU, extends the STPU to include unobservable events.
- A re-characterization of its levels of controllability
- A sound extension to the reduction rules for maintaining the labeled distance graph.

Future Work

- Resolve the complexity class of dynamic controllability
- Extending the linear cutoff algorithm to be applied to POSTPU.