Temporal Constraint Networks

Search & Optimization

Esteban Lanter

University of Basel

December 04th, 2014

Table of Contents

Introduction

Temporal Constraint Networks: Reasoning about time

- Points in time
- Time Intervals
- Relative locations of points or intervals

Introduction

3 major models:

- Point Algebra
- Interval Algebra
- Quantitative Temporal Networks

Tradeoff: Expressiveness vs. Tractability

Qualitative Networks

Reasoning on qualitative temporal statements

- No quantitative statements about time
- Qualitative statements about the relative location of temporal elements
- Interval Algebra: Temporal objects are intervals
- Point Algebra: Temporal objects are time points

Both approaches are closely related

Qualitative Networks

Example

John was not in the room when I touched the switch to turn on the light, but John was in the room later when the light went out.

What do we want to reason about?

- Is the temporal statement consistent?
- If yes, what kind of possible scenarios along the time line can we construct?

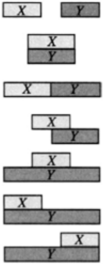
Interval Algebra

Qualitative statements regarding the relative locations of paired intervals

- 7 atomic relations between intervals (13 including inverses)

Relation	Symbol	Inverse
X before Y	b	bi
X equal Y	=	=
X meets Y	m	mi
X overlaps Y	o	oi
X during Y	d	di
X starts Y	s	si
X finishes Y	f	fi

Example



- The relative location of the Intervals I and J can be specified by a relation set $\{r_1, \ldots, r_k\} \rightarrow I\{r_1, \ldots, r_k\}J \rightarrow (Ir_1J) \vee \ldots \vee (Ir_kJ)$

Interval Algebra - Example

Example

John was not in the room when I touched the switch to turn on the light, but John was in the room later when the light went out.

Representing this knowledge within the IA framework:

- [*Switch*] as time of touching the switch, [*Light*] as time that light was on, [*Room*] as time John was in the room
- [*Switch*] *overlaps* or *meets* [*Light*]: $Switch\{o, m\}Light$
- [*Switch*] is either *before*, *meets*, *met by*, or *after* [*Room*]: $Switch\{b, m, mi, a\}Room$
- [*Light*] *overlaps*, *starts*, or is *during* [*Room*]: $Light\{o, s, d\}Room$.

Interval Algebra Network

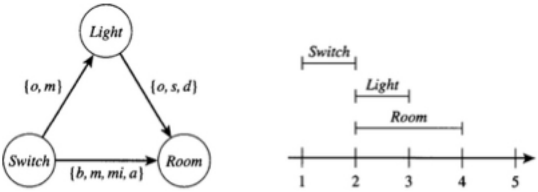
Knowledge can be represented as a constraint network:

- Variables representing temporal intervals: $\{x_1, \ldots, x_n\}$
- Domain: Ordered pairs of real numbers representing beginning & end point of corresponding interval: $D_i = (a, b)$
- Binary constraints between pairs of intervals given as IA relations: $C_{ij} \subseteq \{b, m, o, s, d, f, bi, mi, oi, si, di, fi, =\}$
  $\rightarrow$ 8191 possible constraints
- Solution: Assignment of pair of numbers to each variable so that no constraint is violated

Unique to IA representation of TCP: Constraints are given as enumerated atomic relationships, and not as explicit relations over variable domains.

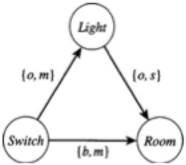
Interval Algebra Network - Example

Graph Representation and one possible solution



- Solution corresponds to feasible relations: (*Switch m Light*), (*Light s Room*), (*Switch m Room*)
- IA reasoning is mainly interested in finding the consistent *qualitative* arrangement of the intervals
- Deciding consistency and finding solutions requires search and inference algorithms

The Minimal IA Network



IA networks are binary → Minimal Networks

- Most explicit representation of all feasible relations between all pairs of intervals
- It is tighter but equivalent as it has the same solution set
- Search algorithms still necessary to find solutions - but search space is smaller
- How to generate the minimal network? → **Path-Consistency**

Path-Consistency in IA Networks

Path consistency requires inference → in IA: Composition

- The *composition* of 2 basic relations r' & r'' can be deduced using a transitivity table

	b	*s*	*d*	*o*	*m*
b	*b*	*b*	*bomds*	*b*	*b*
s	*b*	*s*	*d*	*bom*	*b*
d	*b*	*d*	*d*	*bomds*	*b*
o	*b*	*o*	*ods*	*bom*	*b*
m	*b*	*m*	*ods*	*b*	*b*

Path-Consistency in IA Networks - Composition

Example

2 basic relations:

- I meets K: $r' = I\{m\}K$
- K is during J: $r'' = K\{d\}J$

- Composition $r' \otimes r'' = m \otimes d = \{o, d, s\}$

Path-Consistency in IA Networks - Composition

IA has multiple relations in a constraint → composite relations

- Composition of 2 composite relations:

$$R' \otimes R'' = \{r' \otimes r'' | r' \in R', r'' \in R''\}$$

Example

- $\{b, d, o\} \otimes \{s, o\} = \{b, d, o, m, s\}$

Path-Consistency in IA Networks - Relaxation

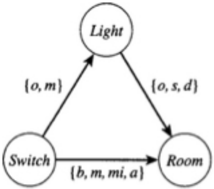
Path consistency through Relaxation

- For three given variables x_i, x_j, and x_k, the equivalent path-consistent subset can be achieved by applying the relaxation algorithm:

$$C_{ij} \leftarrow \; C_{ij} \oplus (C_{ik} \otimes C_{kj})$$

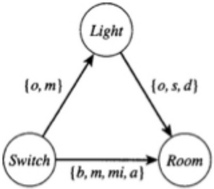
- Repeated application of the relaxation algorithm can convert a the whole network into its equivalent path-consistent form.

Path Consistency in IA - Example I



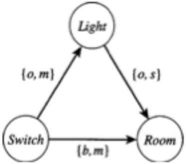
- Apply $C_{SR} \leftarrow C_{SR} \oplus (C_{SL} \otimes C_{LR})$
- $C_{SL} \otimes C_{LR} = \{o, m\} \otimes \{o, s, d\} = \{b, o, m, d, s\} = C_{SR}{}'$
- $C_{SR} = \{b, m, mi, a\}$
- $C_{SR} \leftarrow C_{SR} \cap C_{SR}{}' = \{b, m\}$

Path Consistency in IA - Example II



- Apply $C_{LR} \leftarrow C_{LR} \oplus (C_{LS} \otimes C_{SR})$
- $C_{LS} \otimes C_{SR} = \{oi, mi\} \otimes \{b, m\} = \{b, m, o, fi, di, si\} = {C_{LR}}'$
- $C_{LR} = \{o, s, d\}$
- $C_{LR} \leftarrow C_{LR} \cap {C_{LR}}' = \{o, s\}$

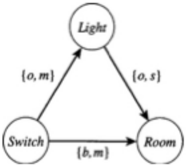
Path Consistency in IA - Example III



- No changes after further application of relaxation
- The resulting Network is now path-consistent
- If relaxation results in empty constraint → inconsistency discovered

But IA networks are generally incomplete

- Minimal networks are not guaranteed to be found
- Minimal networks not guaranteed to be globally consistent → backtracking may still be necessary



- The relation set (*Switch o Light*), (*Light s Room*), (*Switch m Room*) is inconsistent

Point Algebra

Time points and not intervals

- less expressive $\rightarrow$ more tractable
- Transitivity table:

	<	=	>
<	<	<	?
=	<	=	>
>	?	=	>

- In Point Algebra, consistency can be decided in $O(n^2)$
- Path consistency can always generate a minimal network in $O(n^4)$ ($O(n^3)$ for special subsets)
- Point Algebra preferred framework if problem can be stated (translation IA $\rightarrow$ PA sometimes possible)

Quantitative Temporal Networks

QTN vs IA and PA

- IA and PA formulate temporal knowledge in form of *qualitative* statements on the relative locations of time variables
- **Quantitative temporal networks work with explicit metrical information on time**

Quantitative Temporal Networks

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

What do we want to know?

- Is the information in the story consistent?
- Is it possible that John took the bus and Fred used the carpool?
- What are the possible times at which Fred left home?

The Temporal Constraint Satisfaction Problem

Defining the TCSP

- A set of variables $\{x_1, \ldots, x_n\}$ with continuous domains
  - Each variable represents a time point. A time point is the beginning or ending of an event or a neutral point in time
- A set of constraints $\{I_1, \ldots, I_k\} = \{[a_1, b_1], \ldots, [a_k, b_k]\}$
  - each constraint is represented by a set of intervals
  - each interval is defined by a pair of time points
  - Unary constraint: $(a_1 \leq x_i \leq b_1) \vee \ldots \vee (a_k \leq x_i \leq b_k)$
  - Binary constraint: $(a_1 \leq x_j - x_i \leq b_1) \vee \ldots \vee (a_k \leq x_j - x_i \leq b_k)$

Building the TCSP network

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

Identify intervals!

- John is traveling to work
  - John leaves home x_1 and John arrives at work x_2
  - Interval $[x_1, x_2]$
- Fred is traveling to work
  - Fred leaves home x_3 and Fred arrives at work x_4
  - Interval $[x_3, x_4]$

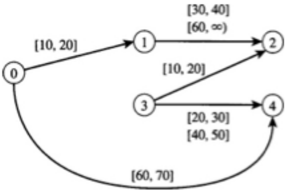
Building the TCSP network

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

The directed constraint graph

- x_o is the beginning of the world
- All time quantities are relative to x_0
- Assign $x_0 = 7{:}00$ A.M.
- $10 \leq x_1 - x_0 \leq 20$
- $20 \leq x_4 - x_3 \leq 30$ or $40 \leq x_4 - x_3 \leq 50$



Properties I

The TCSP is a binary network → usual rules apply!

- Value v is *feasible* for variable x_i if there exists a solution in which $x_i = v$
- Set of all feasible values for a domain is *minimal domain*
- Set of all feasible values for $x_i - x_j$ is a *minimal constraint*
- A network is minimal iff its domains and constraints are minimal
- If every consistent assignment of values to variables can be extended to a solution → *binary decomposable*

Properties II

Inference techniques:

- Union
- Intersection
- Composition

Using these, we can find out:

- Is the Network consistent?
- If the network is consistent, what are possible legal scenarios? When can x_i occur? What are legal relations between x_i and x_j?

These are all NP-hard to solve! **But there is a special class of temporal Problems that can be processed in polynomial time**

STP - The Simple temporal Problem

The STP is a TCSP in which all constraints specify a single interval

- Each edge $i \to j$ is labeled by a single interval $[a_{ij}, b_{ij}]$ that represents the constraint $a_{ij} \leq x_j - x_i \leq b_{ij}$
- Expressed a pair of inequalities: $x_j - x_i \leq b_{ij}$ and $x_i - x_j \leq -a_{ij}$

Convenient representation: Distance Graph

- Not the same as directed constraint graph
- Each edge $i \to j$ labeled by weight a_{ij} representing $x_i - x_j \leq -a_{ij}$

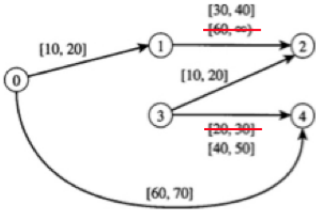
This network can be solved by a shortest-path algorithm!

STP - Example

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

- Only singular intervals → we (must) assume that John used a car and Fred used a carpool.
- The other possibilities (John uses bus and Fred uses car) are disregarded (for now)
- $T_{12} = \{(30, 40)\}$
- $T_{34} = \{(40, 50)\}$

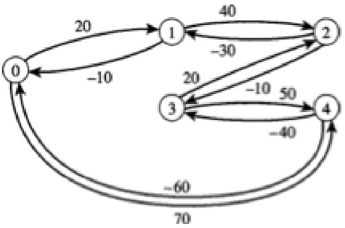


STP - Example

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

- Only singular intervals → we (must) assume that John used a car and Fred used a carpool.
- The other possibilities (John uses bus and Fred uses car) are disregarded (for now)
- $T_{12} = \{(30, 40)\}$
- $T_{34} = \{(40, 50)\}$



STP - Observations

Each path from point i to point j induces a constraint on the distance $x_j - x_i$

- $x_j - x_i \leq \sum_{j=1}^{k} a_{i_{j-1},i_j}$

There can be multiple paths from i to j which induce multiple constraints

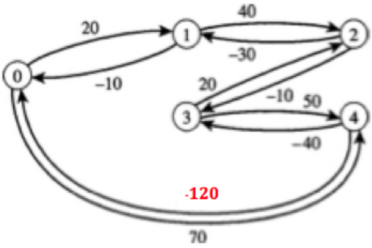
- $x_4 - x_1 = a_{01} + a_{12} + a_{23} + a_{34} = 20 + 40 - 10 + 50 = 100$
- $x_4 - x_1 = a_{10} + a_{04} = -10 + 70 = 60$

The intersection of all these constraints is given by the shortest path d_{ij} between i and j

- $x_j - x_i \leq d_{ij}$
- $min(x_4 - x_1) = 60 \rightarrow d_{14} = 60$

STP - Consistency

The STP is consistent iff there are no negative cycles



- Cycle: $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_0$
- Sum of inequalities:
  $a_{01} + a_{12} + a_{23} + a_{34} + a_{40} \equiv x_0 - x_0 \leq -20 \rightarrow \textit{inconsistent}!$

STP - Consistency

If the STP is consistent

- $d_{0j} \leq d_{0i} + a_{ij}$ will always be satisfied

$$\rightarrow d_{0j} - d_{0i} \leq a_{ij}$$

- $d_{01}, d_{02}, \ldots, d_{0n}$ will always satisfy all inequality constraints a_{ij} on the STP edges
- Hence, one of the solution sets of an STP has the form:

$$(x_1 = d_{01}, x_2 = d_{02} \ldots x_n = d_{0n})$$

- Together with its negative reverse we have 2 solution sets:

$$S_1 = (d_{01}, \ldots, d_{0n}), S_2 = (-d_{10}, \ldots, -d_{n0})$$

- Upper/lower bounds in the sense that these assign the latest or earliest possible times

The D-Graph

Distances can be specified by a complete and directed d-graph

	0	*1*	*2*	*3*	*4*
0	0	20	50	30	70
1	−10	0	40	20	60
2	−40	−30	0	−10	30
3	−20	−10	20	0	50
4	−60	−50	−20	−40	0

- This is a more explicit representation of the STP
- Any consistent STP is backtrack-free (decomposable) relative to the constraints in its d-graph.

The Minimal STP Network

Domains and constraints characterized by the D-Graph can easily be converted into a minimal representation

- Minimal network $M_i ij = \{[-d_{ji}, d_{ij}]\}$
- Feasible values for variable $X_i = [d_{i0}, d_{0i}]$

	0	1	2	3	4
0	[0]	[10, 20]	[40, 50]	[20, 30]	[60, 70]
1	[−20, −10]	[0]	[30, 40]	[10, 20]	[50, 60]
2	[−50, −40]	[−40, −30]	[0]	[−20, −10]	[20, 30]
3	[−30, −20]	[−20, −10]	[10, 20]	[0]	[40, 50]
4	[−70, −60]	[−60, −50]	[−20, −30]	[−50, −40]	[0]

The minimal network gives us bounds for all feasible values for any potential temporal relationship between the variables

Constructing the D-Graph

The Floyd-Warshall Algorithm

ALL-PAIRS-SHORTEST-PATHS

Input: A distance graph $G = (V, E)$ with weights a_{ij} for $(i, j) \in E$.

Output: A d-graph.

1. **for** $i := 1$ **to** n **do** $d_{ii} \leftarrow 0$
2. **for** $i, j := 1$ **to** n **do** $d_{ij} \leftarrow a_{ij}$
3. **for** $k := 1$ **to** n **do**
4. **for** $i, j := 1$ **to** n **do**
5. $d_{ij} \leftarrow min\,(d_{ij}, d_{ik} + d_{kj})$

- Applying the Floyd-Warshall algorithm is equivalent to imposing Path-Consistency on an STP
- Inconsistency is detected by examining sign of the diagonal elements of the D-Graph
- Runtime $O(n^3)$

Assembling the solution

- Start with $x_0 = 0$
- Assign to each variable any value that satisfies the D-Graph constraints relative to its previous assignments
- Due to no backtracking computational burden remains mainly on constructing the D-Graph ($O(n^3)$)

The general TCSP

As opposed to the STP, the general TCSP has multiple intervals per edge

- General approach: Decompose TCSP into multiple STP's - Permute through selections with only 1 Interval per constraint
- 1 particular selection of Intervals: *labeling*
- Minimal TCSP network: Union over all individual minimal networks of all possible labelings T

The general TCSP-Example

Example

John travels to work either by car (30-40 minutes) or by bus (at least 60 minutes). Fred travels to work either by car (20-30 minutes) or in a carpool (40-50 minutes). Today John left home between 7:10 and 7:20 A.M., and Fred arrived at work between 8:00 and 8:10 A.M. We also know that John arrived at work 10-20 minutes after Fred left home.

4 labelings - 4 Individual STP's

- John: car, Fred: car
- John: car, Fred, carpool
- John: bus, Fred: car
- John: bus, Fred: carpool

Solving the general TCSP

Straightforward:

- Brute Force enumeration through all labelings - $O(n^3 k^e)$

Sophisticated:

- Meta-CSP - Variables are TCSP edges, Domains are possible intervals
- Backtracking algorithm assigns intervals to edges
- Backtrack if negative cycle is encountered on current STP

Preprocessing TCSP's:

- Enforce path consistency or directional path consistency on TCSP