Simple Temporal Networks

and extensions

Jan Mrkos

PUI Tutorial
Week 11

VI Recap

Try identify missing (?) components in the equation of the value update (V_n to V_{n+1}):

$$V_n(s) = \max_{?} \sum_{?} ?(?)[?(?) + \gamma ?(?)]$$

Try identify missing (?) components in the equation of the value update (V_n to V_{n+1}):

$$V_n(s) = \max_{?} \sum_{?} ?(?)[?(?) + \gamma ?(?)]$$

$$V_n(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V_{n-1}(s')]$$

VI Recap

Try identify missing (?) components in the equation of the value update (V_n to V_{n+1}):

$$V_n(s) = \max_{?} \sum_{?} ?(?)[?(?) + \gamma ?(?)]$$

$$V_n(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[R(s, a, s') + \gamma V_{n-1}(s')]$$

What is the difference to the Bellman equation?

Outline

- Recap of Simple Temporal Networks
- Simple Temporal Network example

Motivation

Many planners don't work with time explicitly. Given a plan *Simple Temporal Networks* (STNs) can be used to efficiently:

Motivation

Many planners don't work with time explicitly. Given a plan *Simple Temporal Networks* (STNs) can be used to efficiently:

- check time consistency of a plan under time constraints,

Motivation

Many planners don't work with time explicitly. Given a plan *Simple Temporal Networks* (STNs) can be used to efficiently:

- check time consistency of a plan under time constraints,
- if consistent, determine temporal schedule,

Motivation

Many planners don't work with time explicitly. Given a plan *Simple Temporal Networks* (STNs) can be used to efficiently:

- check time consistency of a plan under time constraints,
- if consistent, determine temporal schedule,
- manage real-time execution of a plan and new constraints.

Simple Temporal Networks [1]

Def: Simple Temporal Network

A *Simple Temporal Network* (STN) is a pair $S = (T, C)$ where:

- T is a set of *time-points*, real valued variables
- C a set of constraints of the form:

$$Y - X \leq \delta$$

for $X, Y \in T$ and $\delta \in \mathbb{R}$

[1] Slides based mostly on AIMA, these slides and this example, definition by [Dechter et al., 1991]

Simple Temporal Networks [1]

Def: Simple Temporal Network

A *Simple Temporal Network* (STN) is a pair $S = (T, C)$ where:

- T is a set of *time-points*, real valued **variables**
- C a set of constraints of the form:

$$Y - X \leq \delta$$

for $X, Y \in T$ and $\delta \in \mathbb{R}$

[1] Slides based mostly on AIMA, these slides and this example, definition by [Dechter et al., 1991]

Simple Temporal Networks [1]

Def: Simple Temporal Network

A *Simple Temporal Network* (STN) is a pair $S = (T, C)$ where:

- T is a set of *time-points*, real valued **variables**
- C a set of constraints of the form:

$$Y - X \leq \delta$$

for $X, Y \in T$ and $\delta \in \mathbb{R}$

Q: Can time-points in T be assigned values, so that C is satisfied? (Is STN consistent?)

[1] Slides based mostly on AIMA, these slides and this example, definition by [Dechter et al., 1991]

Simple Temporal Networks [1]

Def: Simple Temporal Network

A *Simple Temporal Network* (STN) is a pair $S = (T, C)$ where:

- T is a set of *time-points*, real valued **variables**
- C a set of constraints of the form:

$$Y - X \leq \delta$$

for $X, Y \in T$ and $\delta \in \mathbb{R}$

Q: Can time-points in T be assigned values, so that C is satisfied? (Is STN consistent?)

We map STNs to graphs (how?)

[1] Slides based mostly on AIMA, these slides and this example, definition by [Dechter et al., 1991]

Simple Temporal Networks [1]

Def: Simple Temporal Network

A *Simple Temporal Network* (STN) is a pair $S = (T, C)$ where:

- T is a set of *time-points*, real valued **variables**
- C a set of constraints of the form:

$$Y - X \leq \delta$$

for $X, Y \in T$ and $\delta \in \mathbb{R}$

Q: Can time-points in T be assigned values, so that C is satisfied? (Is STN consistent?)

We map STNs to graphs:

- Variables $\rightarrow$ nodes
- Constraints $\rightarrow$ edges

[1] Slides based mostly on AIMA, these slides and this example, definition by [Dechter et al., 1991]

Example

I have a plan for getting to the PDV exam:

- Take a train from Kolín to Prague-Libeň
- Walk from Prague-Libeň to Vysočanská (Yellow - B line)
- Take metro from Vysočanská to Karlovo náměstí

Example

I have a plan for getting to the PDV exam:

- Take a train from Kolín to Prague-Libeň
- Walk from Prague-Libeň to Vysočanská (Yellow - B line)
- Take metro from Vysočanská to Karlovo náměstí

Regardless whether there are other possible (better) plans, we want to check whether this one is consistent (i.e. feasible) under following constraints:

Example

I have a plan for getting to the PDV exam:

- Take a train from Kolín to Prague-Libeň
- Walk from Prague-Libeň to Vysočanská (Yellow - B line)
- Take metro from Vysočanská to Karlovo náměstí

Regardless whether there are other possible (better) plans, we want to check whether this one is consistent (i.e. feasible) under following constraints:

- A friend drops me off at the at the station in Kolin at 8:00.
- Train ride takes *at least* 50 minutes, I might have to wait for the train.
- Walking takes 10 to 20 minutes, depending whether I run or walk.
- Ride on the metro takes *at most* 20 minutes, the metro runs every minute.
- I have to be at the PDV exam by 9:30.

Example

$$O \text{ –Train– } X_1 \text{ – (walk) – } X_2 \text{ –Metro– } X_3$$

Example

$$O \text{ –Train– } X_1 \text{ – (walk) – } X_2 \text{ –Metro– } X_3$$

Given the constraints, we have $S = (T, C)$:

Example

$$O \text{ –Train– } X_1 \text{ – (walk) – } X_2 \text{ –Metro– } X_3$$

Given the constraints, we have $S = (T, C)$:
(We introduce a special reference variable (node), $O = 0$ as a starting point.)

$$T = \{O, X_1, X_2, X_3\}, \text{where } O \text{ maps to 8:00}$$

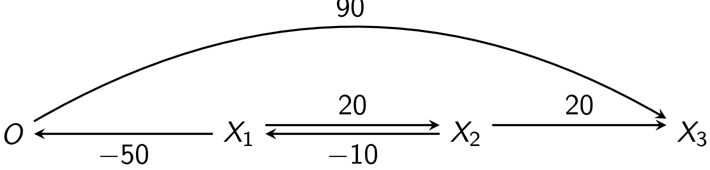
Example

$$O \text{ –Train– } X_1 \text{ – (walk) – } X_2 \text{ –Metro– } X_3$$

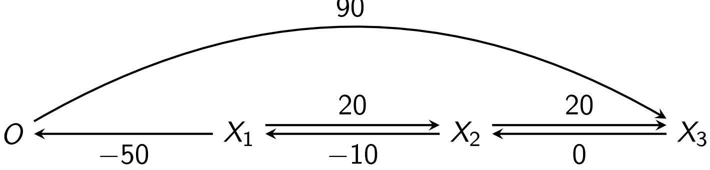
Given the constraints, we have $S = (T, C)$:
(We introduce a special reference variable (node), $O = 0$ as a starting point.)

$$T = \{O, X_1, X_2, X_3\}, \text{where } O \text{ maps to 8:00}$$

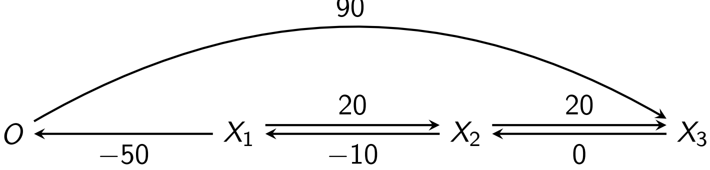
$$C = \begin{cases} O - X_1 \leq -50 & \text{train} \\ X_2 - X_1 \leq 20 & \text{walk} \\ X_1 - X_2 \leq -10 & \text{walk} \\ X_3 - X_2 \leq 20 & \text{metro} \\ X_3 - O \leq 90 & \text{exam start} \end{cases}$$



$$C = \begin{cases} O - X_1 \leq -50 & \text{train} \\ X_2 - X_1 \leq 20 & \text{walk} \\ X_1 - X_2 \leq -10 & \text{walk} \\ X_3 - X_2 \leq 20 & \text{metro} \\ X_3 - O \leq 90 & \text{exam start} \\ & \end{cases}$$

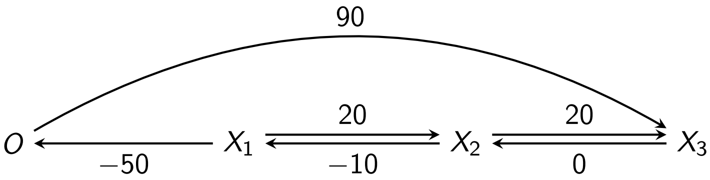


$$C = \begin{cases} O - X_1 \leq -50 & \text{train} \\ X_2 - X_1 \leq 20 & \text{walk} \\ X_1 - X_2 \leq -10 & \text{walk} \\ X_3 - X_2 \leq 20 & \text{metro} \\ X_3 - O \leq 90 & \text{exam start} \\ X_2 - X_3 \leq 0 & \text{metro} \end{cases}$$

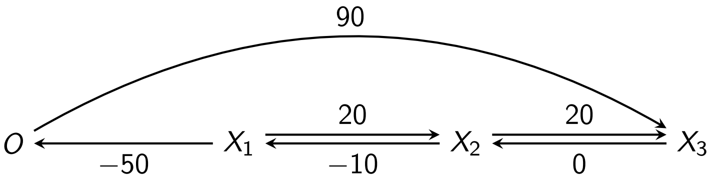


$$C = \begin{cases} O - X_1 \leq -50 & \text{train} \\ X_2 - X_1 \leq 20 & \text{walk} \\ X_1 - X_2 \leq -10 & \text{walk} \\ X_3 - X_2 \leq 20 & \text{metro} \\ X_3 - O \leq 90 & \text{exam start} \\ X_2 - X_3 \leq 0 & \text{metro} \end{cases}$$

Note: Right-to-left arrows are (+) upper bounds, left-to-right are (-) lower bounds

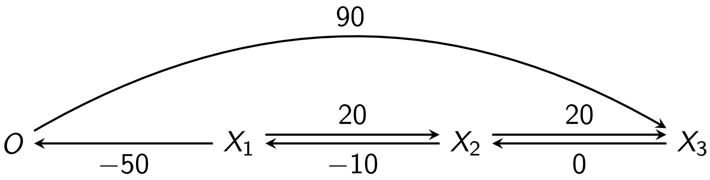


Explicit constraints generate other, implicit constraints:



Explicit constraints generate other, implicit constraints:

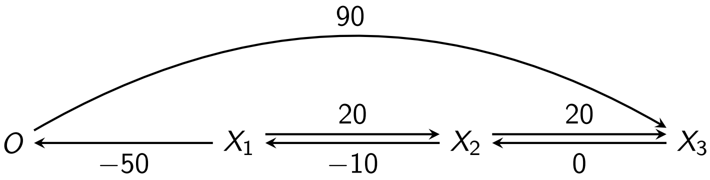
- Sum constraints $\rightarrow$ **paths** in graph (e.g., $X_3 - X_1 \leq 40$)



Explicit constraints generate other, implicit constraints:

- Sum constraints → **paths** in graph (e.g., $X_3 - X_1 \leq 40$)
- **Stronger** constraints → **shorter** paths

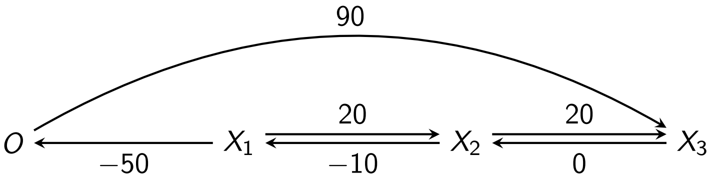
Example



Using the explicit constraints, we can calculate shortest path lengths between all combinations of nodes:

D	O	X_1	$X2$	X_3
O				90
X_1	-50		20	
X_2		-10		20
X_3			0	

Example

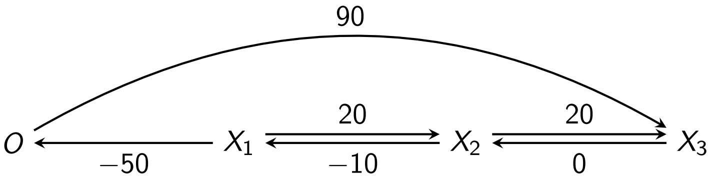


Using the explicit constraints, we can calculate shortest path lengths between all combinations of nodes:

D	O	X_1	$X2$	X_3
O				90
X_1	-50		20	
X_2		-10		20
X_3			0	

(e.g. by using Floyd-Warshall in more complex cases)

Example



Using the explicit constraints, we can calculate shortest path lengths between all combinations of nodes:

D	O	X_1	$X2$	X_3
O	0	80	90	90
X_1	-50	0	20	40
X_2	-60	-10	0	20
X_3	-60	-10	0	0

(e.g. by using Floyd-Warshall in more complex cases)

Consistency?

Questions:

- Q: Was the plan consistent?

Consistency?

Questions:

- Q: Was the plan consistent?
- A: Yes.

Consistency?

Questions:

- Q: Was the plan consistent?
- A: Yes.
- Q: When would it be inconsistent?

Consistency?

Questions:

- Q: Was the plan consistent?
- A: Yes.
- Q: When would it be inconsistent?
- A: e.g. if I wanted to wake up at 9:00

Consistency?

Questions:

- Q: Was the plan consistent?
- A: Yes.
- Q: When would it be inconsistent?
- A: e.g. if I wanted to wake up at 9:00
- Q: How would we know it was infeasible?

Consistency?

Questions:

- Q: Was the plan consistent?
- A: Yes.
- Q: When would it be inconsistent?
- A: e.g. if I wanted to wake up at 9:00
- Q: How would we know it was infeasible?

Consistency?

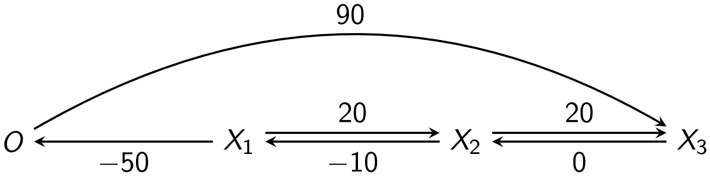
Questions:

- Q: Was the plan consistent?
- A: Yes.
- Q: When would it be inconsistent?
- A: e.g. if I wanted to wake up at 9:00
- Q: How would we know it was infeasible?

Thm: "Fundamental Theorem" of STNs

STN consistent $\iff$ Distance matrix has zeros on diagonal $\iff$ graph has no negative cycles

Summary



- Solution is an assignment of values to timepoints (nodes) that satisfies given constraints.
- If such solution exists, it is consistent.
- Consistency can be checked by checking the distance matrix.

Thank you for participating in the tutorials :-)

Please fill out the feedback form →

https://forms.gle/gQHP1uA3CLajtcdr8