# Temporal Constraints Satisfaction Problems (TCSP)

Dra. Mª Dolores Rodríguez Moreno

Universidad de Alcalá
ISG

# Objectives

| **Specific Objectives** |
|---|
| • Main representation in TCSP<br>• Main techniques |
| **Source** |
| • Dechter, Meiri, and Pearl (1991). Simple Temporal Networks. Artificial Intelligence 49:61–95.<br>• Dana Nau's slides for Automated Planning. Licensed under License https://creativecommons.org/licenses/by-nc-sa/2.0/<br>• Brian William. (2002). Slides. Temporal Planning, Scheduling and Execution |

# Outline

- **Motivation**
- Definition
- Simple Temporal Network (STN)
- Properties of STN
- A Completed Plan Forms an STN
- Conclusions
- ToDo Example

# Motivation example

- I would like to solve this problem:
  - John goes to work by car, which takes 30–40 minutes
  - Fred goes to work in a carpool, which takes 40–50 minutes
  - Today, John left home between 7:10 and 7:20
  - Fred arrived at work between 8:00 and 8:10
  - We also know that John arrived at work about 10-20 mn after Fred left home
- What are the times that Fred left home?

# Outline

- Motivation
- **Definition**
- Simple Temporal Network (STN)
- Properties of STN
- A Completed Plan Forms an STN
- Conclusions
- ToDo Example

# Definition

- Involves a set of variables: $X_{i\ \ldots}\ X_n$
- Each variable represents a time point (TP)
- Each constraint represented by a set of intervals: $\{I_1, \ldots, I_n\} = \{[a_1, b_1], \ldots. [a_n, b_n]\}$
  - $T_i = (a_i \leq X_i \leq b_i)$ or . . . or $(a_i \leq X_i \leq b_i)$ → unary
  - $T_{ij} = (a_1 \leq X_i - X_j \leq b_1)$ or ... or $(a_n \leq X_i - X_j \leq b_n)$ → binary
- a constraint $c_{i \longrightarrow j}$ between two time points $x_i$ and $x_i$ can be expressed as a union of intervals
$$x_j - x_i \in I_{ij1} \cup \cdots \cup I_{ijn} = [a_{ij1}, b_{ij1}] \cup \cdots \cup [a_{ijn}, b_{ijn}]$$
- Use Simple Temporal Networks representation

# Outline

- Motivation
- Definition
- **Simple Temporal Network (STN)**
- Properties of STN
- A Completed Plan Forms an STN
- Conclusions
- ToDo Example

# Simple Temporal Network (I)

- Formally:
  - A STN S is a pair (T, C) where T is a set {***to***, t1, …, tn} of TP variables
  - C is a finite set of binary constraints on those variables, each constraint having the form $t_j - t_i \leq \delta$, for some real number $\delta$
  - The "variable" ***to*** → represents an arbitrary, fixed reference point on the time-line
  - The constraints in C are called the explicit constraints in S
- Implementation: A directed graph where vertices are TPs and every arc represents a single temporal constraint defined on two TPs

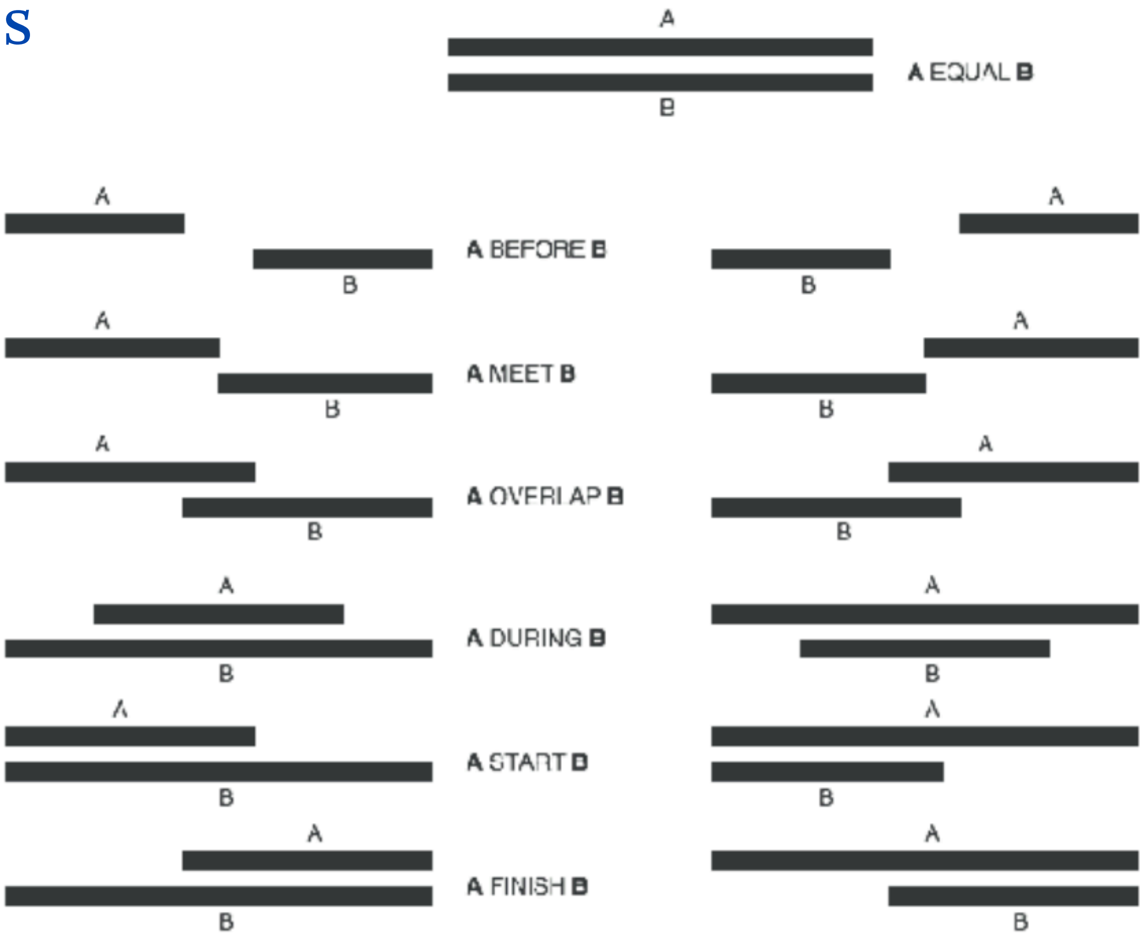
# Simple Temporal Network (II)

- A solution to S=(T, C) is a complete set of variable assignments that satisfies all the constraints in C
- Sufficient to represent most Allen relations
  - Simple metric constraints
- Allows to represent a set of TPs Xi at which events occur
  - Unary constraints

    $(a_0 \leq X_i \leq b_0)$ or $(a_1 \leq X_i \leq b_1)$ or . . .
  - Binary constraints

    $(a_0 \leq X_j - X_i \leq b_0)$ or $(a_1 \leq X_j - X_i \leq b_1)$ or . . .

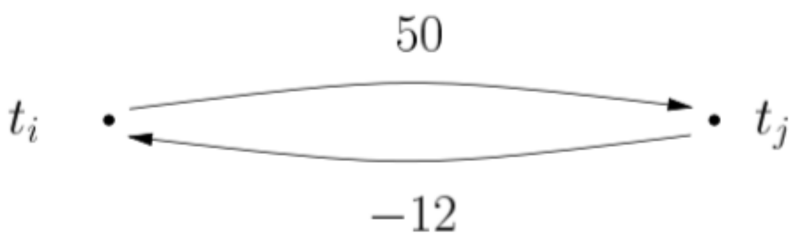
# Allen primitives




Source: Flavio S. Corrêa da Silva

# Simple Temporal Network (III)

- A temporal constraint tj – ti ≤ $\delta$ in an STN is represented in the corresponding graph by a directed edge from ti to tj with weight (or length) $\delta$
- For example: tj – ti ≤ 50 and ti – tj ≤ -12



# Simple Temporal Network (IV)

- It is useful to have a TP, called Z/X0, whose value is fixed at 0
- Binary constraints involving Z are equivalent to unary constraints:

  $X - Z \leq 7 \quad \Longleftrightarrow \quad X \leq 7$

  $Z - X \leq -3 \quad \Longleftrightarrow \quad X \geq 3$

# Motivation example

- John goes to work by car, which takes 30–40 minutes
- Fred goes to work in a carpool, which takes 40–50 minutes
- John left home between 7:10 and 7:20
- Fred arrived at work between 7:50 and 8:10
- John arrived at work after than Fred left home, but not more than 20 minutes later than he left

# Motivation example: representation

- X1 represents John going to work
- X2 represents Fred going to work
- X1 and X2 represent John leaving home and arriving at work, respectively
- X3 and X4 denote the same events for Fred
- We also need a temporal reference point to be able to refer to absolute time points; this is denoted by x0 will stand for seven o'clock this morning

# Example: constraints representation

- $T = \{X_0, X_1, X_2, X_3, X_4\}$,

$X_0/Z =$ 7am

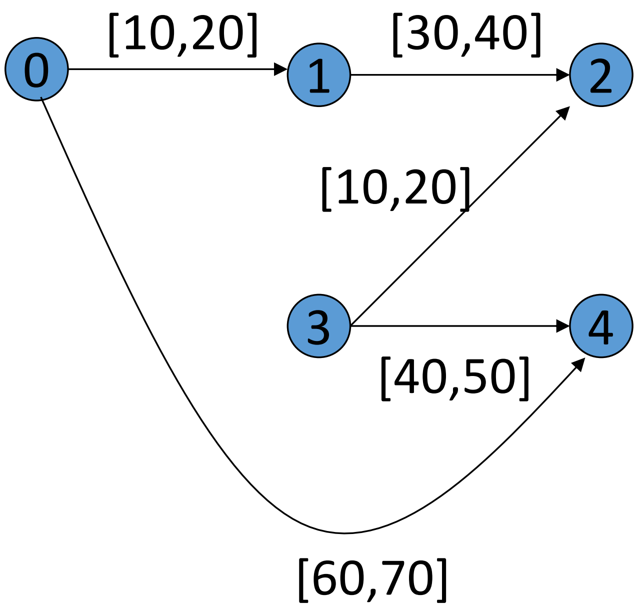
- Time units = minutes
- C=
  - $10 \leq X_1 - X_0 \leq 20$ (John leaving between 7:10 and 7:20)
  - $30 \leq X_2 - X_1 \leq 40$ (John duration 30-40mn)
  - $60 \leq X_4 - X_0 \leq 70$ (Fred arrives between 8:00 and 8:10)
  - $40 \leq X_4 - X_3 \leq 50$ (Fred duration 40-50mn)
  - $10 \leq X_3 - X_2 \leq 20$ (John arrived work about 10-20 mn after Fred left home)

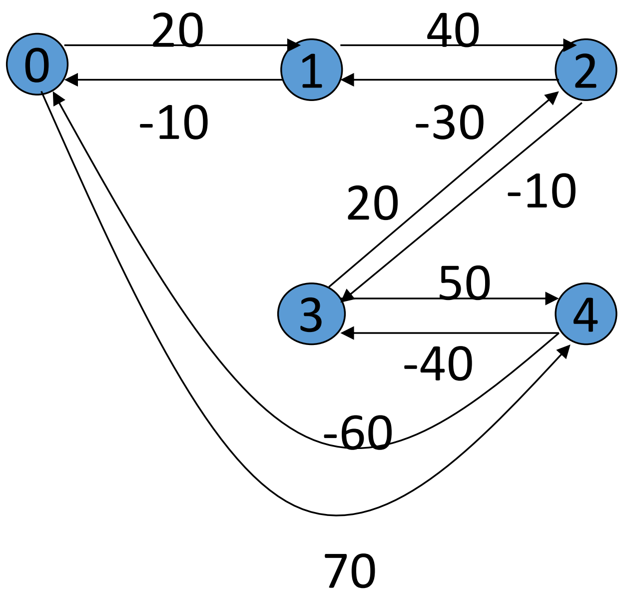
# Example: d-graph representation

$T_{ij} = (a_{ij} \leq X_j - X_i \leq b_{ij})$

$X_j - X_i \leq b_{ij}$

$X_i - X_j \leq - a_{ij}$



# Outline

- Motivation
- Definition
- Simple Temporal Network (STN)
- **Properties of STN**
- A Completed Plan Forms an STN
- Conclusions
- ToDo Example

# Properties of STN (I)

- **Temporal distance** from tj to ti in an STN is the length of the shortest path from tj to ti in the distance graph G (or negative infinity if no such path exists)
- **Distance Matrix** for an STN S = (T,C) is a matrix D each entry of which equals the temporal distance between the corresponding pair of time-points in T

$$-\mathcal{D}(t_j, t_i) \;\leq\; t_j - t_i \;\leq\; \mathcal{D}(t_i, t_j)$$

- The distance Matrix for a given STN may be computed from scratch in polynomial time

# Properties of STN (II)

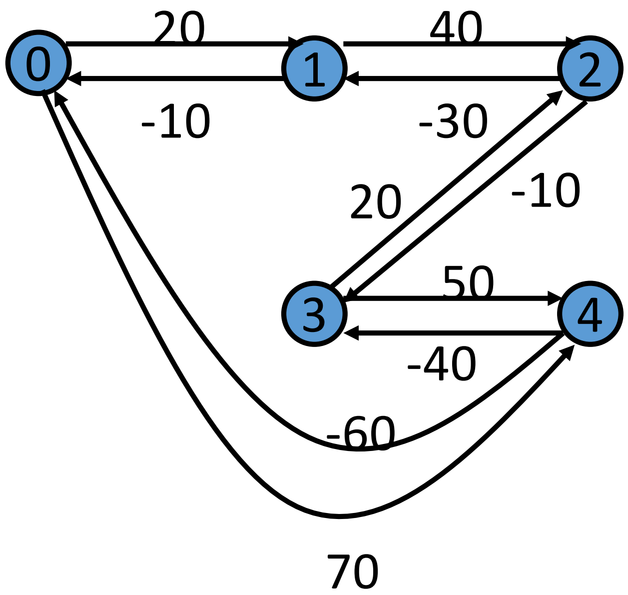
- An STN S is **consistent** if and only if its corresponding distance graph G has no negative cycles (i.e., if and only if the path length around any loop is non-negative)
- An STN S is **consistent** if and only if all of the diagonal entries in its distance matrix are zero
- Explicit constraints can be combined to form implicit constraints:

  $tj - ti \leq 30$ and $tk - tj \leq 40 \rightarrow tk - ti \leq 70$

# Shortest Paths of Gd

| | 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 |

d-graph



# STN Minimum Network

| | 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 |

d-graph

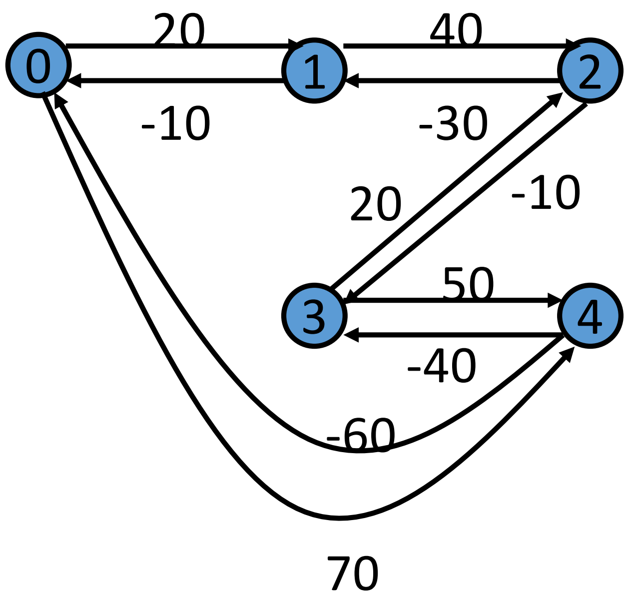
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | [0] | [10,20] | [40,50] | [20,30] | [60,70] |
| **1** | [-20,-10] | [0] | [30,40] | [10,20] | [50,60] |
| **2** | [-50,-40] | [-40,-30] | [0] | [-20,-10] | [20,30] |
| **3** | [-30,-20] | [-20,-10] | [10,20] | [0] | [40,50] |
| **4** | [-70,-60] | [-60,-50] | [-30,-20] | [-50,-40] | [0] |

STN minimum network

# Test Consistency: No Negative Cycles

| | 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 |

d-graph

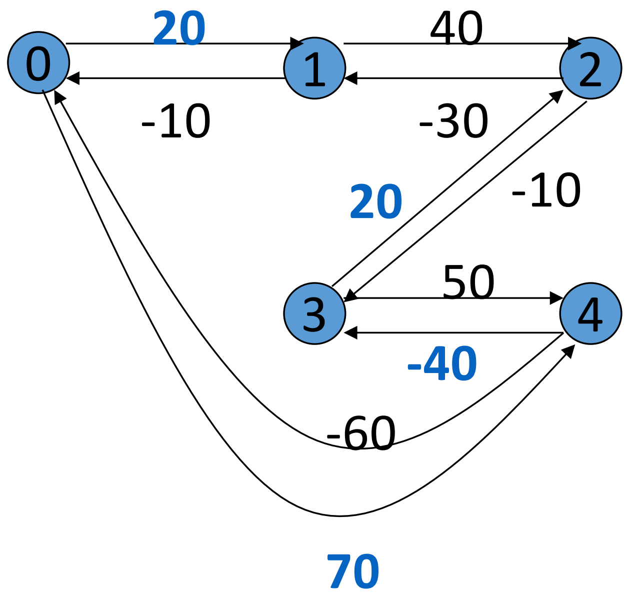


# Latest Solution

Node 0 is the reference.

| | 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 |

d-graph

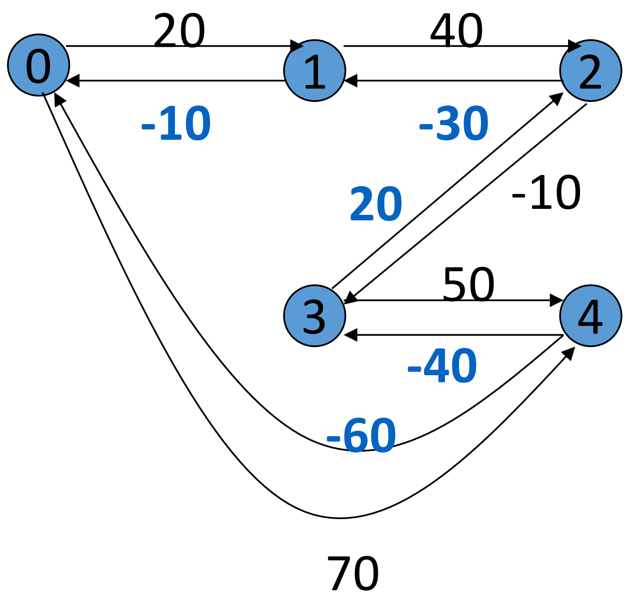


# Earliest Solution

Node 0 is the reference.

| | 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 |

d-graph



# Feasible Values

| | 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 |

d-graph

- $X_1$ in [10, 20]
- $X_2$ in [40, 50]
- $X_3$ in [20, 30]
- $X_4$ in [60, 70]

# Solution by Decomposition

| | 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 |

d-graph

- Select value for 1

  → 15 [10,20]

# Solution by Decomposition

| | 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 |

d-graph

- Select value for 1

  → 15

- Select value for 2, consistent with 1

  → 45 [40,50], 15+[30,40]

# Solution by Decomposition

| | 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 |

d-graph

- Select value for 1

  → 15

- Select value for 2, consistent with 1

  → 45

- Select value for 3, consistent with 1 & 2

  → 30 [20,30], 15+[10,20],45+[-20,-10]

# Solution by Decomposition

| | 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 |

d-graph ➡ **$O(N^2)$**

- Select value for 1

  → 15

- Select value for 2, consistent with 1

  → 45

- Select value for 3, consistent with 1 & 2

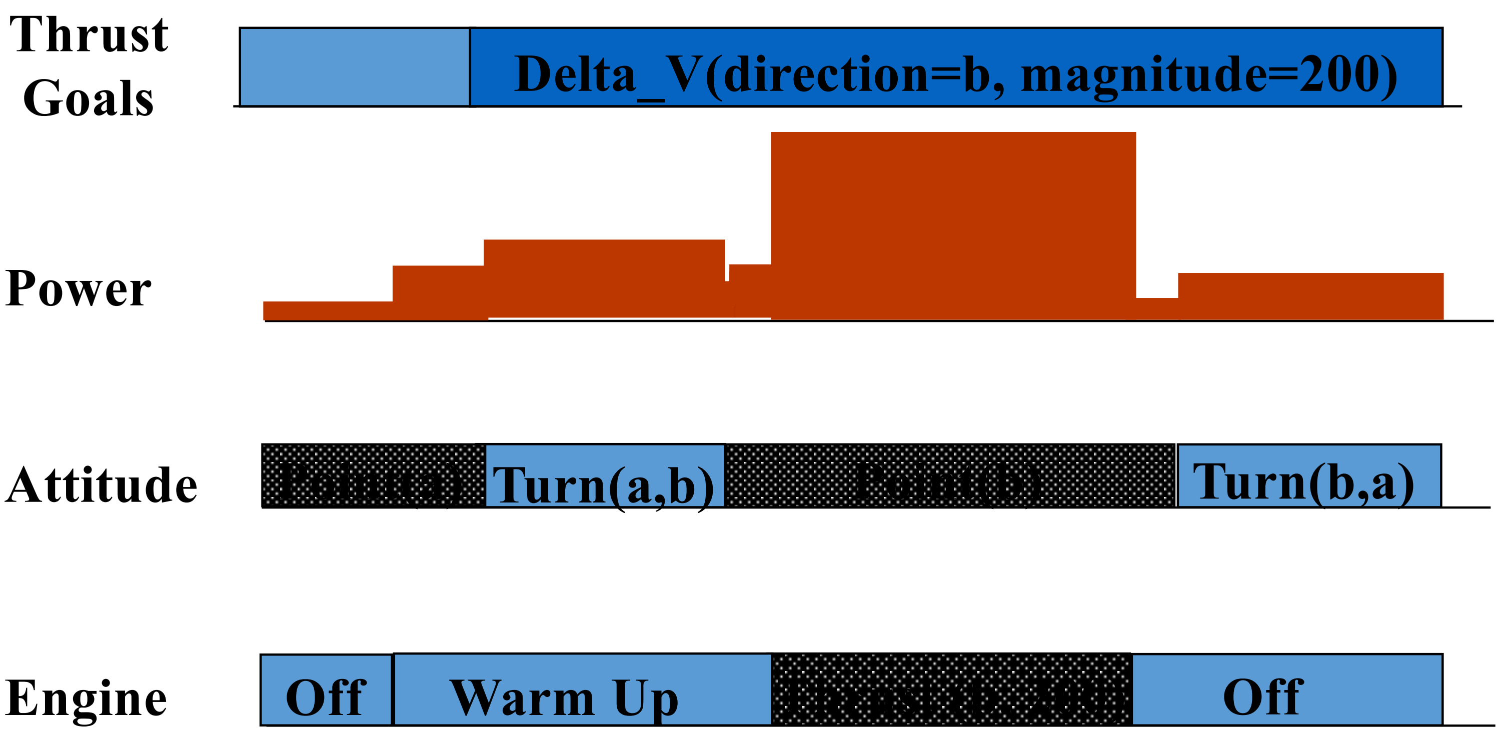
  → 30

- Select value for 4, consistent with 1,2 & 3

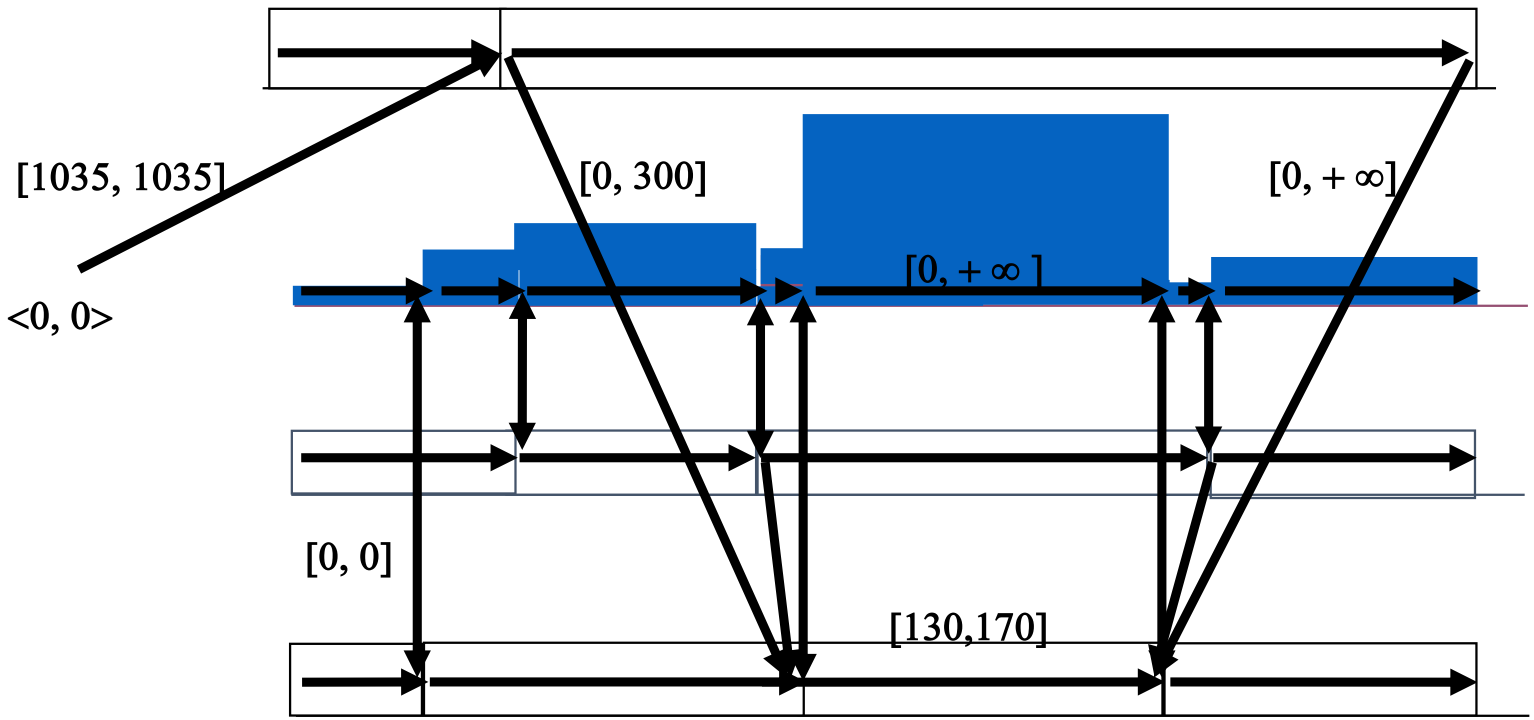
# Outline

- Motivation
- Definition
- Simple Temporal Network (STN)
- Properties of STN
- **A Completed Plan Forms an STN**
- Conclusions
- ToDo Example

# A Completed Plan Forms an STN



# A Completed Plan Forms an STN



# Outline

- Motivation
- Definition
- Simple Temporal Network (STN)
- Properties of STN
- A Completed Plan Forms an STN
- **Conclusions**
- ToDo Example

# Conclusions

- For time, TCSP → uses STN
- STNs have been used to provide flexible planning and scheduling systems for more than a decade
- Efficient algorithms for checking consistency
- However, STNs cannot represent uncertainty (e.g., actions with uncertain durations) or conditional constraints (e.g., only do X if test result is negative).

# Outline

- Motivation
- Definition
- Simple Temporal Network (STN)
- Properties of STN
- A Completed Plan Forms an STN
- Conclusions
- **ToDo Example**

# Example

- Goal: Fly from New York to Rome
  - Leave NewYork after 4p.m.,June 8
  - Return to New York before 10p.m.,June 18
  - Away fromNewYork no more than 7 days
  - In Rome at least 5 days
  - Return flight lasts no more tan 7 hours
- Represent
  - Constraints problem
  - Directed graph
  - Distance matrix