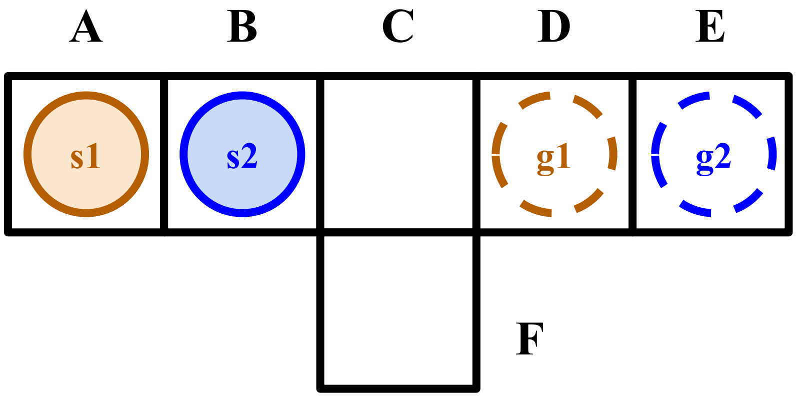
Multi-Agent Path Finding with STNU

Shao-Hung Chan
ID: 2542361779
Date: 12/5/2019

Multi-Agent Path Finding (MAPF)

- Given:
  - A discretized grid environment
  - A series of agents with individual starting points and goals
- At each time step, each agent can:
  - Stay on its current location
  - Move to one neighboring grid
- Find collision-free paths for the agents
- Does not take the **configuration of robots** into account
- May need more accurate time schedules for agents

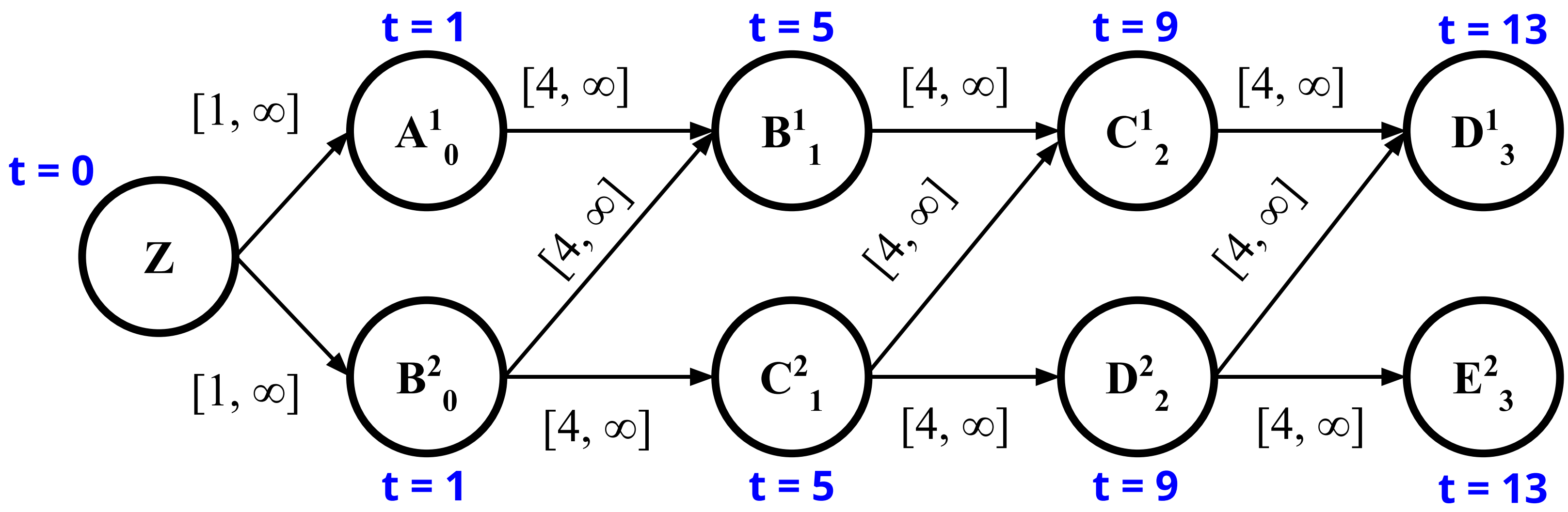


time step	0	1	2	3
R1	A	B	C	D
R2	B	C	D	E

Simple Temporal Network (STN)

- Constraint satisfaction problem
- Given a graph G:
  - Nodes: events
  - Edges: constraints
    - e.g: $4 \leq B^1_1 - A^1_0 \leq \infty$
  - Find the time points of ndes that satisfy the edge constraints
- Can be solved in polynomial time

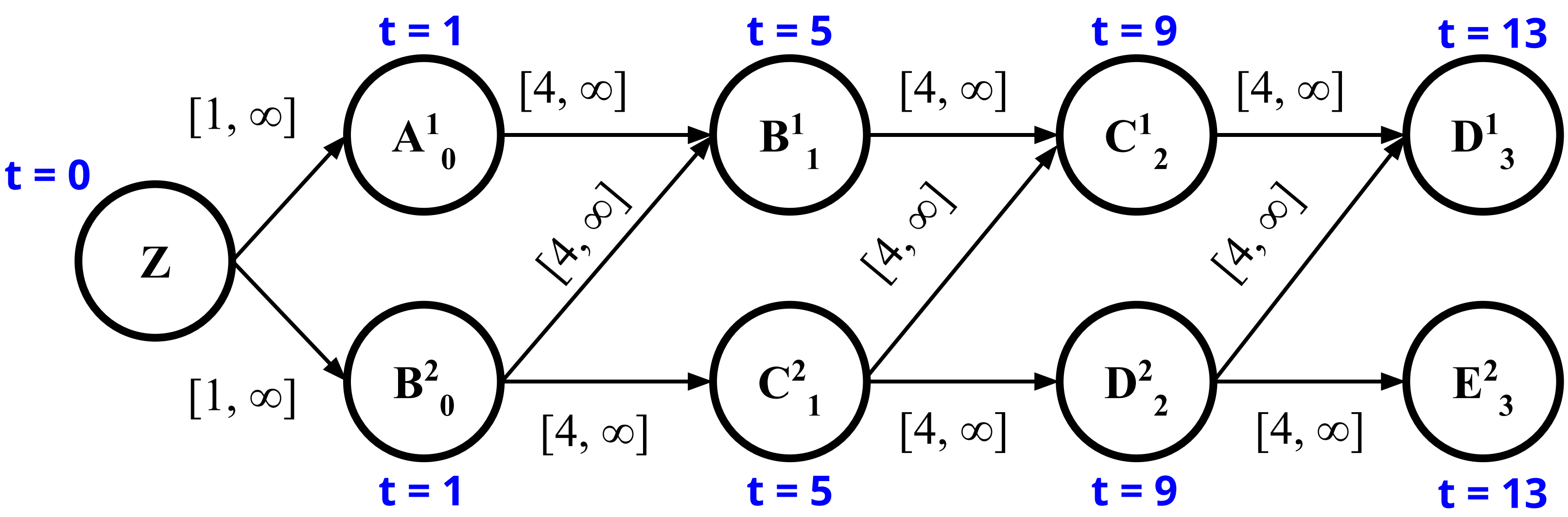
time step	0	1	2	3
R1	A	B	C	D
R2	B	C	D	E



Simple Temporal Network (STN)

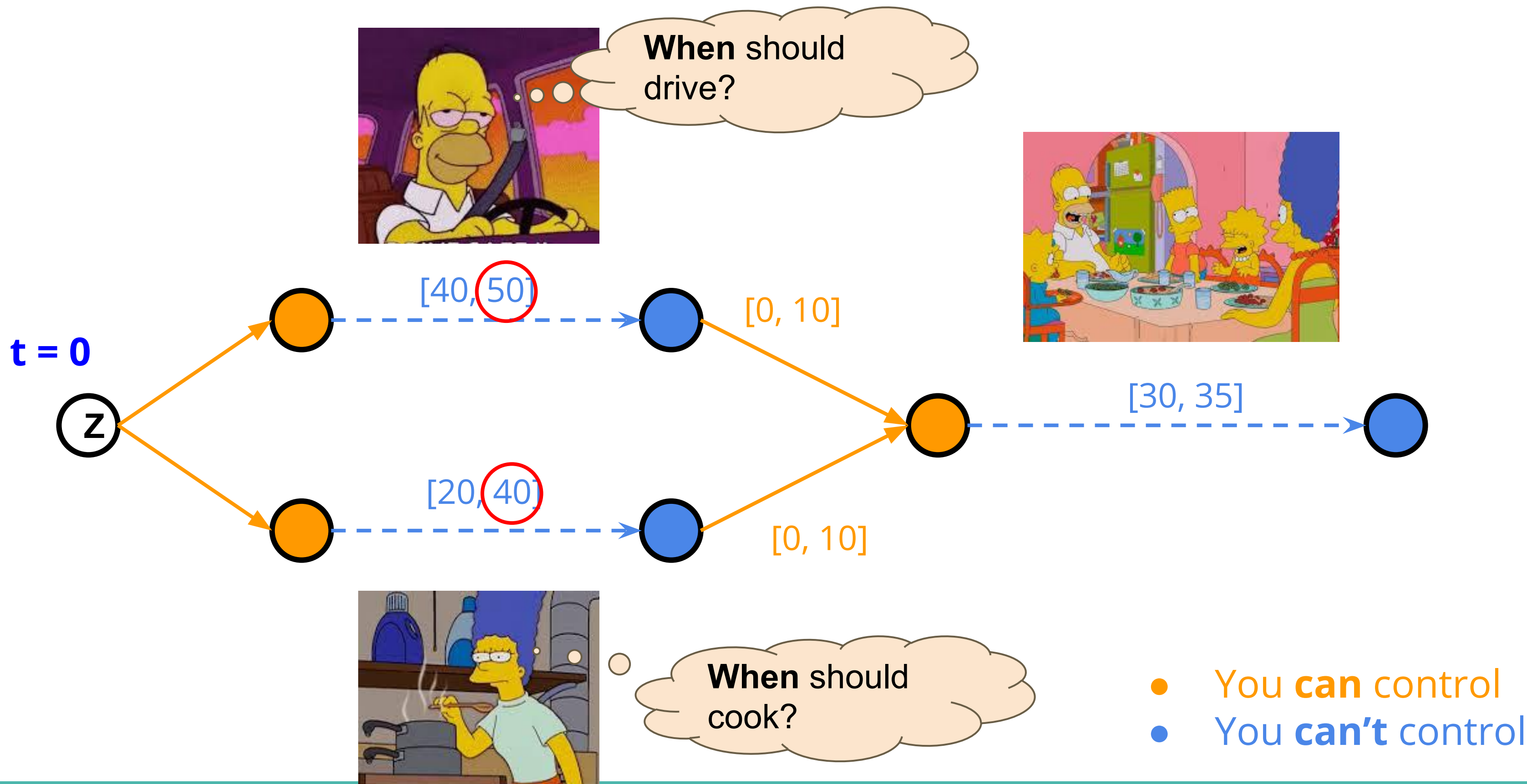
- Constraint satisfaction problem
- Given a graph G:
  - Nodes: events
  - Edges: constraints
    - e.g: $4 \leq B^1_1 - A^1_0 \leq \infty$
  - Find the time points of ndes that satisfy the edge constraints
- Can be solved in polynomial time
- ***What if there exists velocity noises on the robots?***

time step	0	1	2	3
R1	A	B	C	D
R2	B	C	D	E

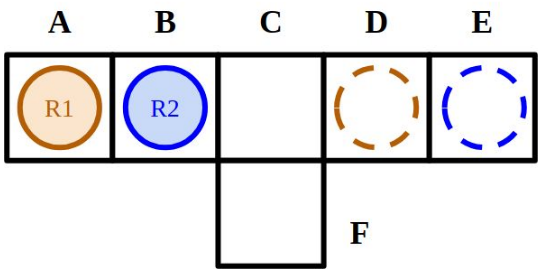


STN with Uncertainty

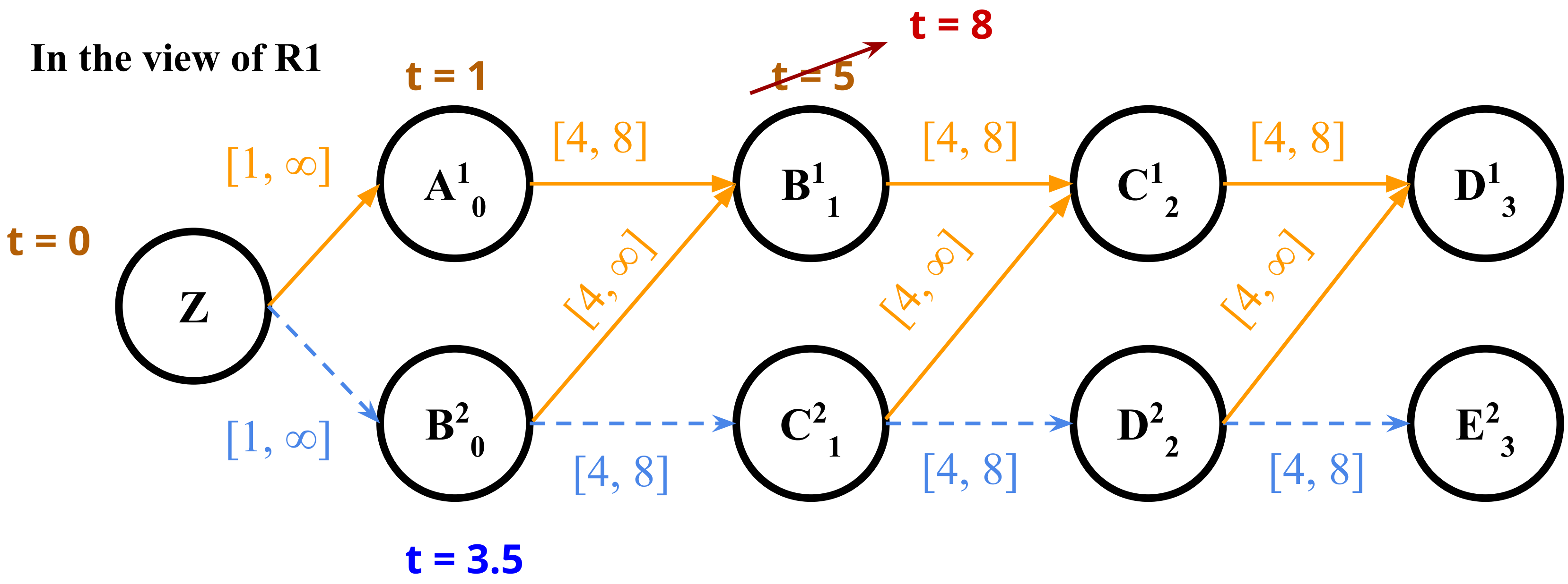
- Simple Temporal Network with Uncertainty (STNU)
- Some events can not be controlled
- Take the **worst case** of the uncertainty into account



Problem: MAPF + STNU



- Handle the uncertainty in MAPF with STNU
- Divide the edges into:
  - ***Free constraints:*** Able to set any time points as long as in the constraint.
  - ***Contingent constraints:*** time period that can only be observed during execution.
- Dynamic controllability
  - The partial sequence executed so far is ensured to extend to a complete solution whatever durations remain to be observed.



Original Scope of the Project

- Try to apply STNU to MAPF to deal with uncertainty
- 2 agents will have their own STNUs assuming itself with no uncertainty
- Given two agents ***i*** and ***j***
  - If the STNU is dynamic controllable:
    - **Execute STNU** → Done, everyone is happy :)
  - Else:
    - **Conflict-direct search** → update the commands and guarantees
    - Check the dynamic controllability again.

Midterm Scope of the Project

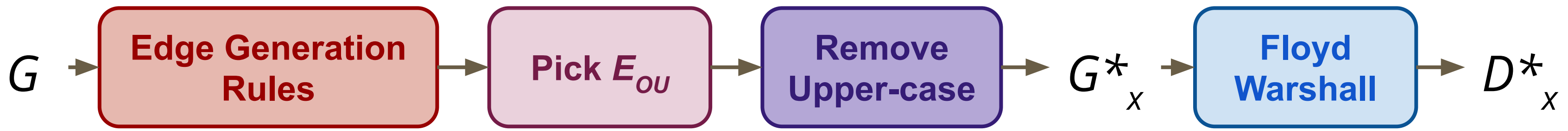
- Try to apply STNU to MAPF to deal with uncertainty
- 2 agents will have their own STNUs assuming itself with no uncertainty
- Given two agents ***i*** and ***j***
  - **Assume** the STNU is dynamic controllable:
    - **Execute STNU** → Done, everyone is happy :)

Final Scope of the Project

- Try to apply STNU to MAPF to deal with uncertainty
- 2 agents will have their own STNUs assuming itself with no uncertainty
- Given two agents ***i*** **(w/o uncertainty)** and ***j*** **(w/ uncertainty)**
  - **Assume** the STNU is dynamic controllable:
    - **Execute STNU** → Done, everyone is happy :)

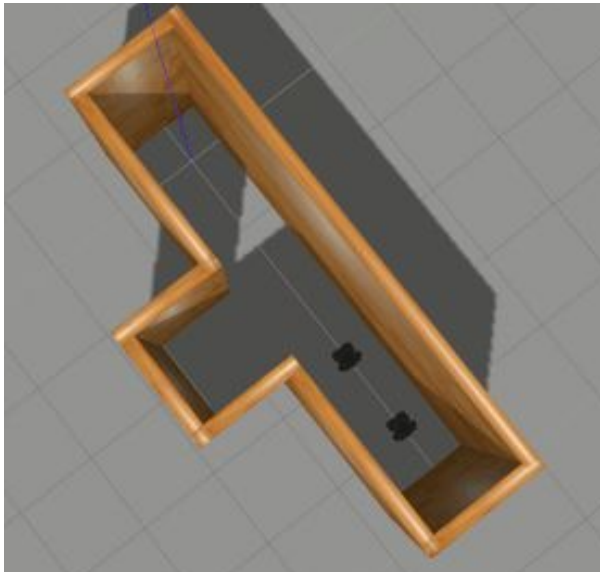
Execution Flow Chart

- Given: STNU graph $G=(V, E)$
  - We can only control X, A_1 and A_2
  - E_O: Ordinary, E_U: Upper-case, E_L: Lower-case
  - $C_1 - C_2 \leq 2$
  - $X - C_1 \leq -1 \rightarrow C_1 - X \geq 1$ (lower bound)
- Preprocessing: Generate **AllMax Graph G^*_x**

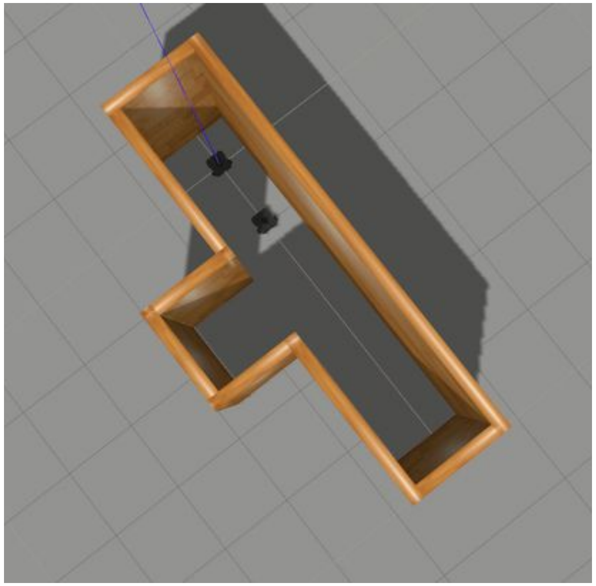


- **FAST-EX Algorithm**: Updating STNU temporally
  - Only update D^*_x entries involving Z.
  - Combines executed vertices into a single vertex
  - Updating distance matrix D^*_x with Johnson's Algorithm
- Generate Real-time Decision
- Interface between STNU and ROS

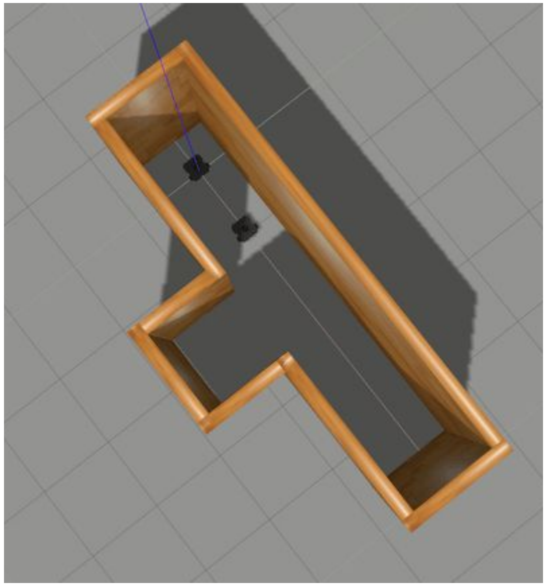
A Toy Demo



w/o Uncertainty



w Uncertainty



STNU Execution