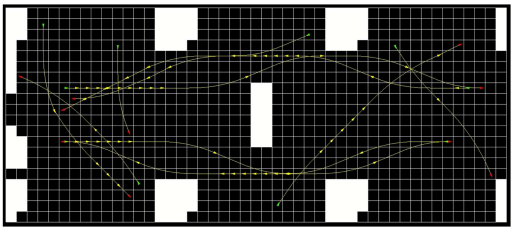
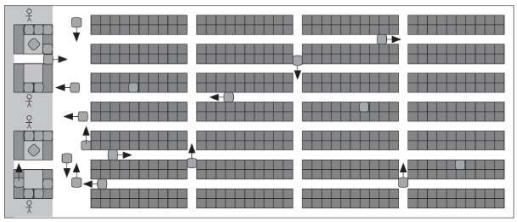
CBS Extensions

- Non-holonomic agents
- Mix of non-anonymous agents (= designated goal locations) and anonymous agents (= assignable goal locations)
- Agents of different sizes
- Deadlines
- Uncertainty about the speeds of agents





see earlier credits

[1] H. Ma and S. Koenig., "Optimal Target Assignment and Path Finding for Teams of Agents", AAMAS 2016.
[2] L. Cohen et al., "Optimal and Bounded-Suboptimal Multi-Agent Motion Planning", SoCS 2019.
[3] J. Li et al., "Multi-Agent Path Finding for Large Agents", AAAI 2019.
[4] H. Ma et al., "Multi-Agent Path Finding with Deadlines", IJCAI 2018.

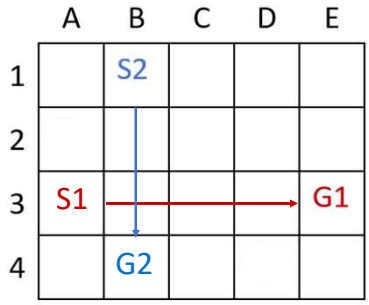
1

Robustness to Delays during Planning and Plan Execution

2

Robust Planning with CBS Extensions

- A MAPF plan is *k*-robust iff no collision occurs even if each agent can be delayed during execution by at most *k* timesteps



[1] D. Atzmon et al., "k-Robust Multi-Agent Path Finding", SoCS, 2017.

3

Robust Planning with CBS Extensions

- A MAPF plan is *p*-collision free iff the probability that no collision occurs during its execution is at most *p*

[1] D. Atzmon et al., "Probabilistic Robust Multi-Agent Path Finding", ICAPS, 2020.

4

Robust Planning with CBS Extensions

- A MAPF plan is *p*-collision free iff the probability that no collision occurs during its execution is at most *p*
  - Detection of potential collisions:
    two agents moving at the nominal speed can now collide even if they are in the same location at different time steps
  - Resolution of potential collisions:
    add a third possibility, namely that the collision will occur
  - Verification that a MAPF plan is collision-free:
    Calculate the probability that no collision occurs and ensure that it is at least *p*

[1] D. Atzmon et al., "Probabilistic Robust Multi-Agent Path Finding", ICAPS, 2020.

5

Robust Plan Execution

- Planning always uses models that are not completely accurate
  - Robots have unmodeled kinematic constraints
  - …
- Plan execution will therefore deviate from the plan
- Strategy 1: If one robot is delayed, delay all robots
  - Problem: Bad throughput
- Strategy 2: Replan whenever plan execution deviates from the plan
  - Problem: Planning is slow since finding good plans is NP-hard

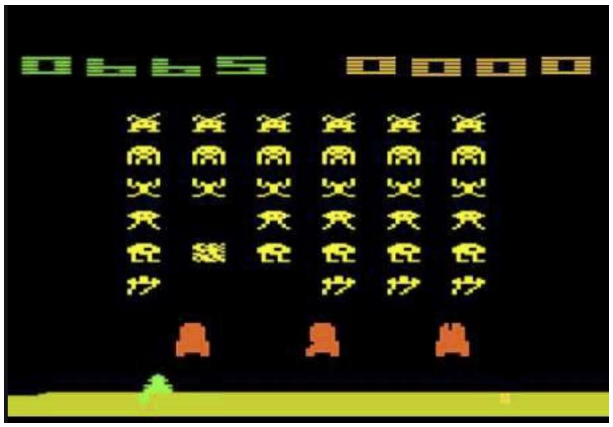
[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

6

Robust Plan Execution: ML

not our work

- PRIMAL: mix of deep reinforcement learning and imitation learning



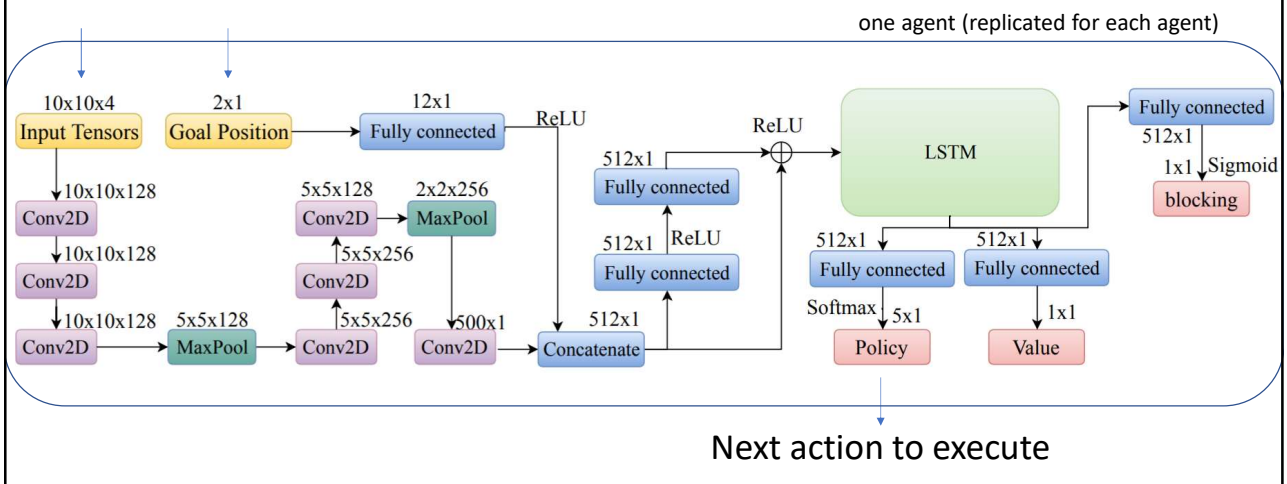
https://medium.com/free-code-camp/explained-simply-how-deepmind-taught-ai-to-play-video-games-9eb5f38c89ee https://www.pcworld.com/article/2889432/google-ai-program-masters-classic-atari-video-games.html

[1] V. Mnih et al., "Human Level Control through Deep Reinforcement Learning", Nature 518, 2015.

7

Robust Plan Execution: ML

- PRIMAL: mix of deep reinforcement learning and imitation learning

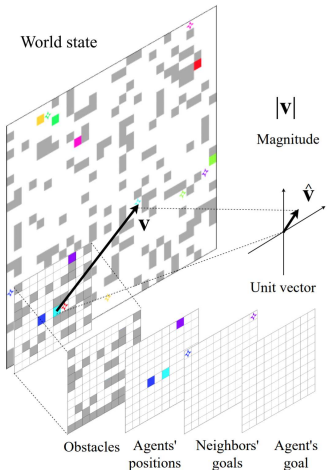


[1] G. Sartoretti et al., "PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning," IEEE RAL 4(3), 2019.

8

Robust Plan Execution: ML

- PRIMAL: mix of deep reinforcement learning and imitation learning

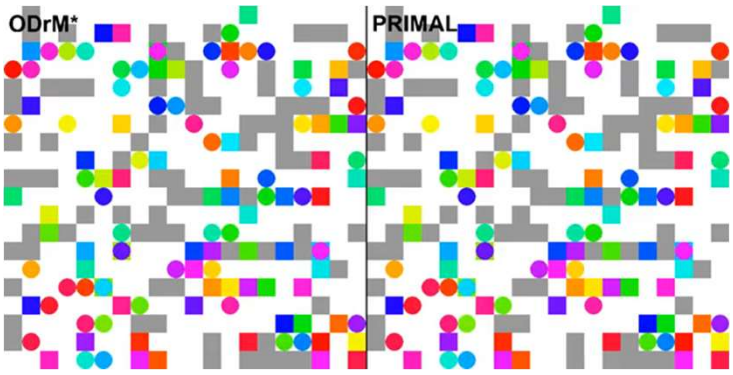


[1] G. Sartoretti et al., "PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning," IEEE RAL 4(3), 2019.

9

Robust Plan Execution: ML

- Training: 20 days in a supercomputing center
- Ideal plan execution: 64 agents – 20x20 map – 20% obstacle density



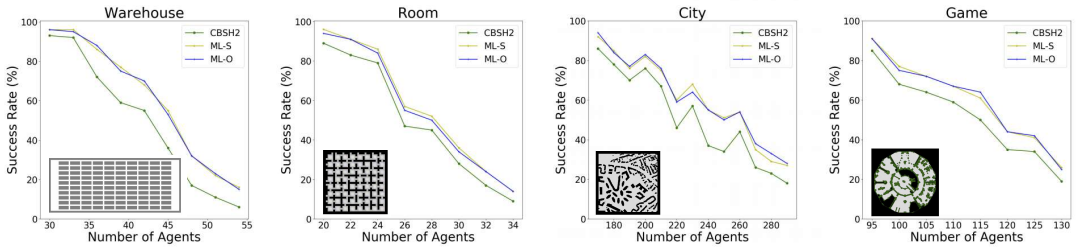
MAPF planner PRIMAL (learning)

[1] G. Sartoretti et al., "PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning," IEEE RAL 4(3), 2019.

11

ML for Planning

- Enhancing planning with machine learning
  - Planners use lots of hard-coded decision strategies
  - Machine learning can often learn to make better decisions
  - The resulting planners can be more efficient and/or effective



[1] T. Huang et al., "Learning to Resolve Conflicts for Multi-Agent Path Finding with Conflict-Based Search," AAAI, 2021.
[2] T. Huang et al., "Learning Node-Selection Strategies in Bounded-Suboptimal Conflict-Based Search for Multi-Agent Path Finding

12

Robust Plan Execution

- Planning always uses models that are not completely accurate
  - Robots have unmodeled kinematic constraints
  - …
- Plan execution will therefore deviate from the plan

- Strategy 1: If one robot is delayed, delay all robots
  - Problem: Bad throughput
- Strategy 2: Replan whenever plan execution deviates from the plan
  - Problem: Planning is slow since finding good plans is NP-hard

[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

13

Robust Plan Execution: Reasoning

- Find a MAPF plan (slow).
- In a feedback loop, repeatedly determine the speed with which each robot should move along its path given its current position (fast).
- Only if the problem becomes unsolvable, find a new MAPF plan.

[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

14

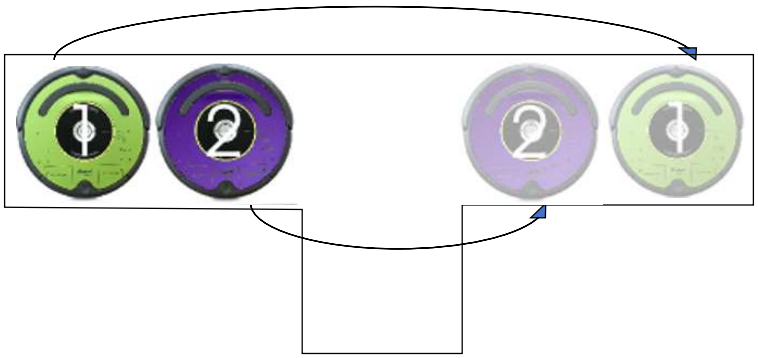
Robust Plan Execution: Reasoning

- MAPF-POST makes use of a simple temporal network to post-process a given MAPF plan in polynomial time to allow for robust plan execution on robots
  - Takes into account edge lengths
  - Takes into account speed limits on edges
  - Takes into account maximum velocities of robots
  - Guarantees safety distances

[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

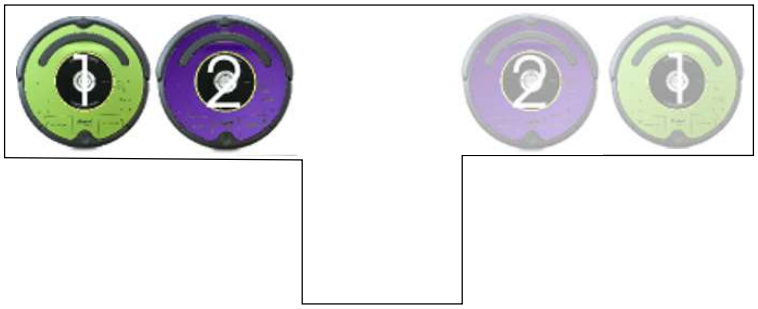
15

Robust Plan Execution: Reasoning



16

Robust Plan Execution: Reasoning



17

Robust Plan Execution: Reasoning

18

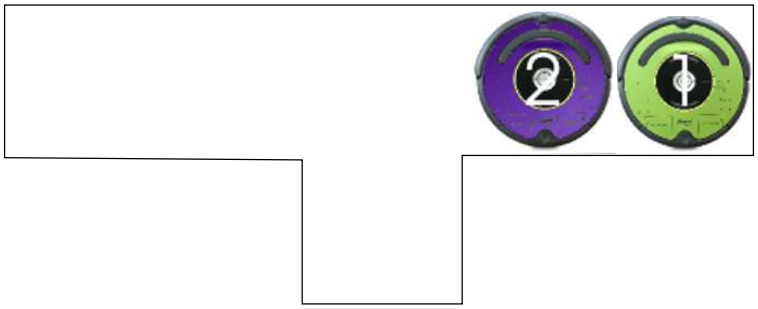
Robust Plan Execution: Reasoning

19

Robust Plan Execution: Reasoning

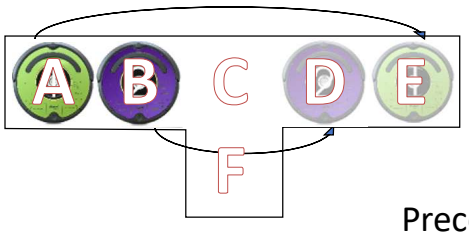
20

Robust Plan Execution: Reasoning



21

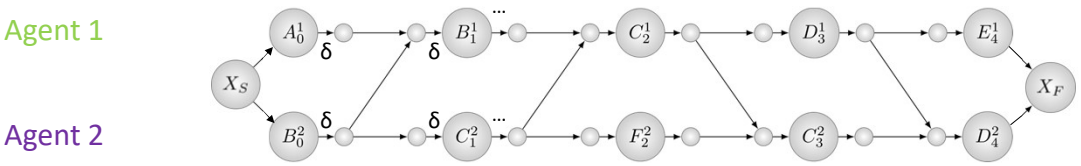
Robust Plan Execution: Reasoning



Agent 1 A → B → C → D → E
Agent 2 B → C → F → C → D

Precedence Graph

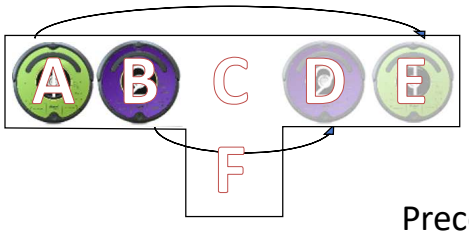
vertex = event that an agent arrives at a location



[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

22

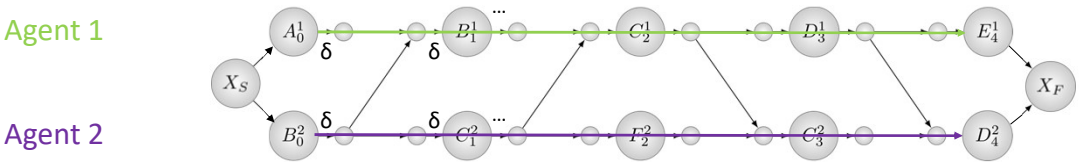
Robust Plan Execution: Reasoning



Agent 1 A → B → C → D → E
Agent 2 B → C → F → C → D

Precedence Graph

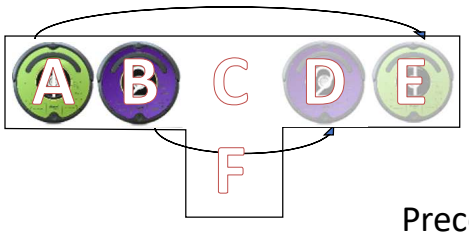
Type 1 edge = order in which the same agent arrives at locations



[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

23

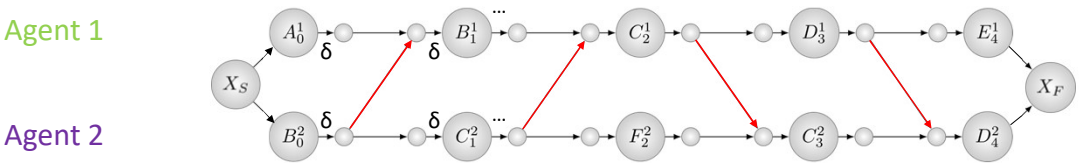
Robust Plan Execution: Reasoning



Agent 1 A → B → C → D → E
Agent 2 B → C → F → C → D

Precedence Graph

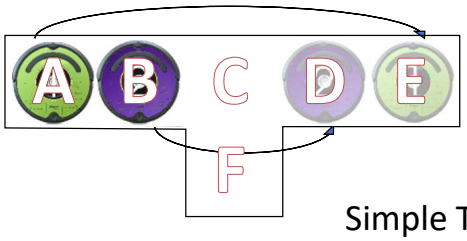
Type 2 edge = order in which two different agents arrive at the same location



[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

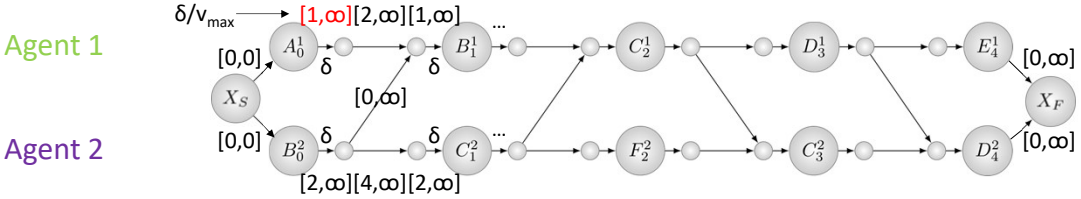
24

Robust Plan Execution: Reasoning



Agent 1 A → B → C → D → E
Agent 2 B → C → F → C → D

Simple Temporal Network



[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.
[2] R. Dechter et al, "Temporal Constraint Networks", AIJ 49(1-3), 1991.

25

Robust Plan Execution: Reasoning

- Minimize makespan and flowtime
  - Determine the earliest arrival times in each location that satisfy the constraints.
  - Calculate the speeds of the robots based on the arrival times.

$$\begin{aligned}&\text{Minimize } \textstyle\sum_{j=1}^{K} t(v^j)\\&\text{such that } t(X_S) = 0\\&\text{and, for all } e = (v, v') \in \mathcal{E}',\\&t(v') - t(v) \geq LB(e)\\&t(v') - t(v) \leq UB(e)\end{aligned}$$

[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.
[2] R. Dechter et al, "Temporal Constraint Networks", AIJ 49(1-3), 1991.

26

Robust Plan Execution: Reasoning

[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

28

Robust Plan Execution: Reasoning

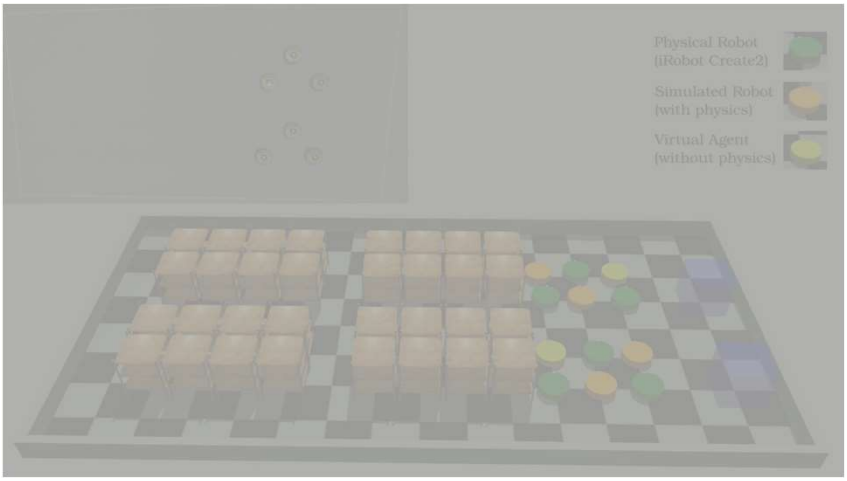
[1] W. Hoenig et al., "Multi-Agent Path Finding with Kinematic Constraints", ICAPS, 2016.

29

Robust Plan Execution: Reasoning

not our work

- Mixed reality simulation



[1] W. Hoenig et al., "Mixed Reality for Robots", IROS, 2015.

30