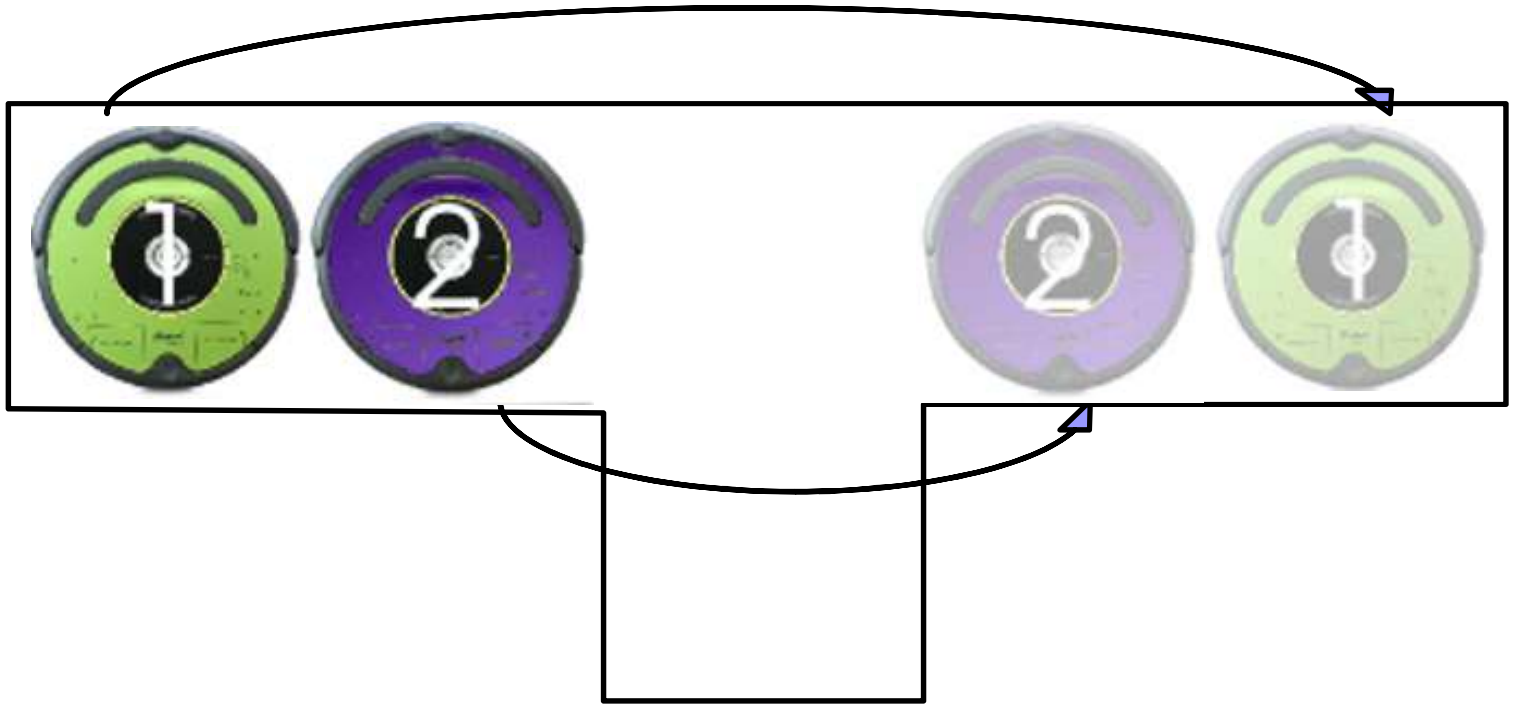
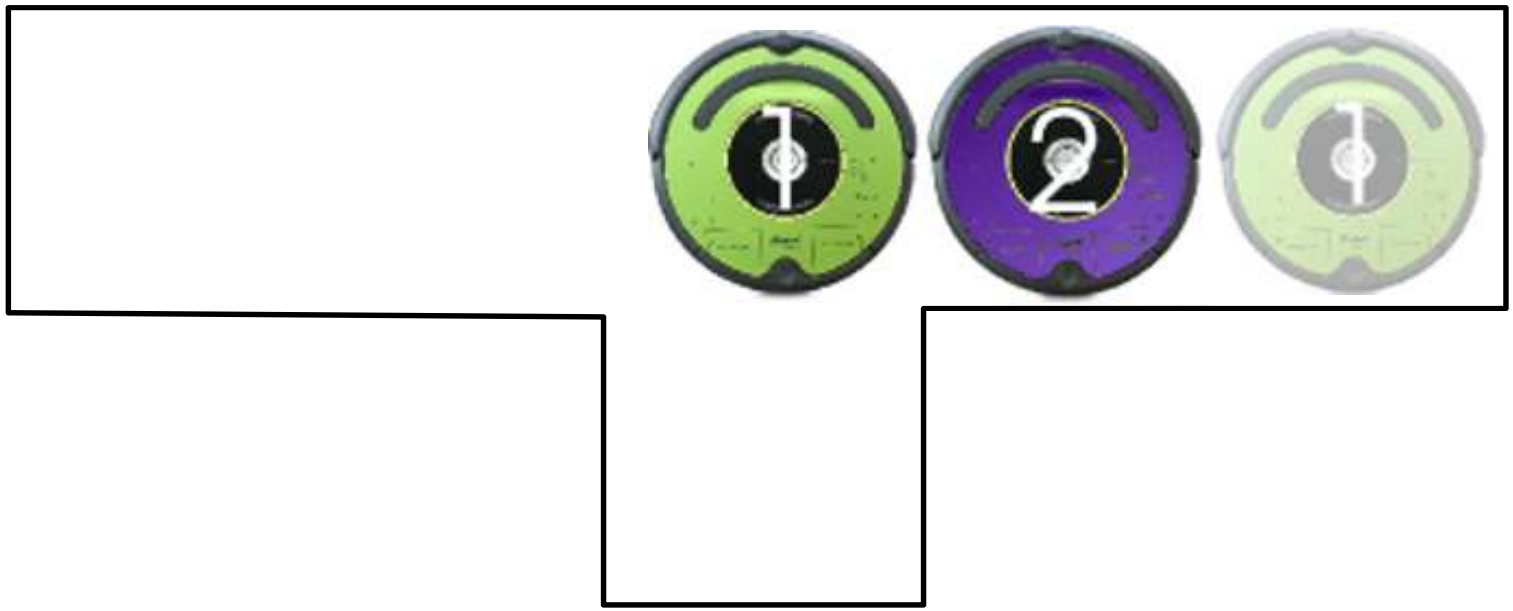
- Tutorial on Heuristic Search at AAMAS-19
- Daniel Harabor, Monash University
- Sven Koenig, University of Southern California
- Nathan Sturtevant, University of Alberta

Multi-Agent Path Finding (MAPF)

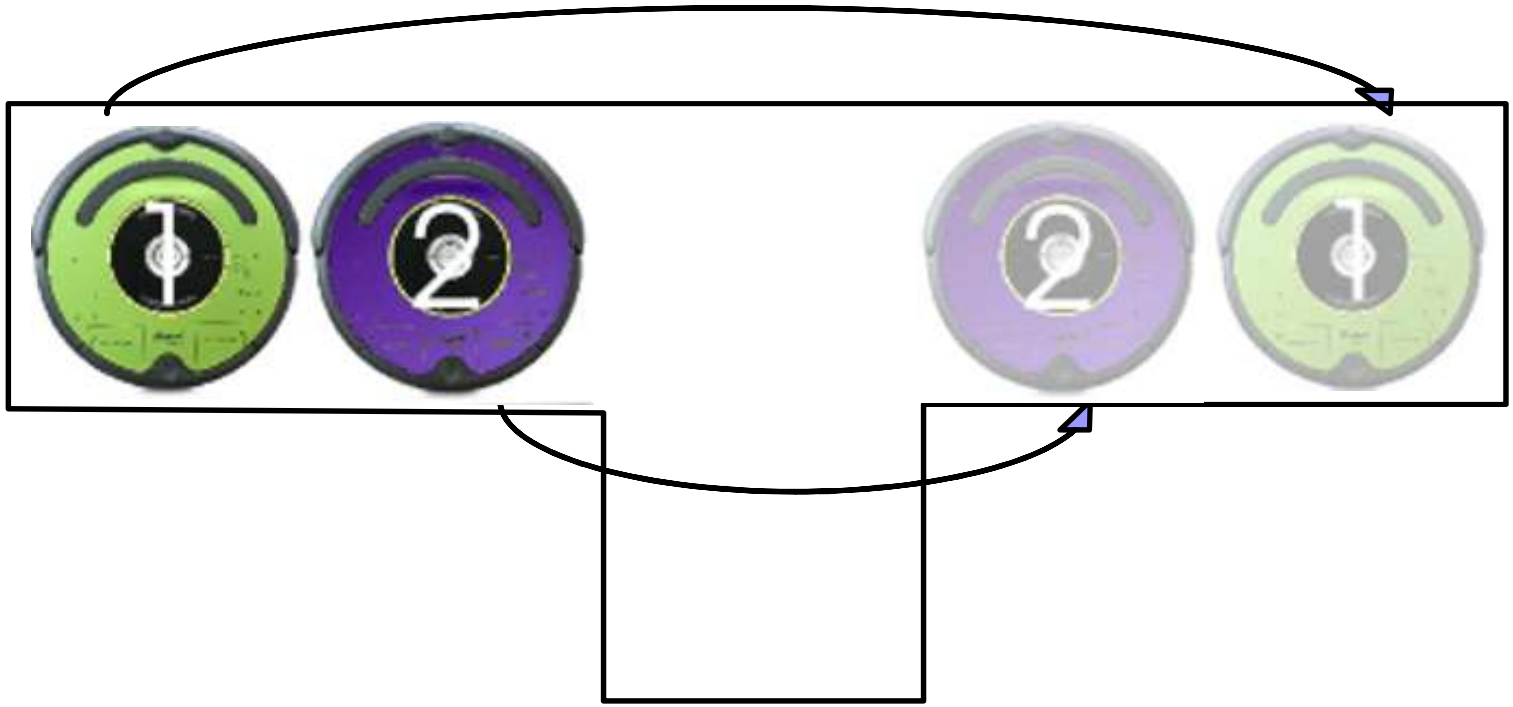


Multi-Agent Path Finding (MAPF)

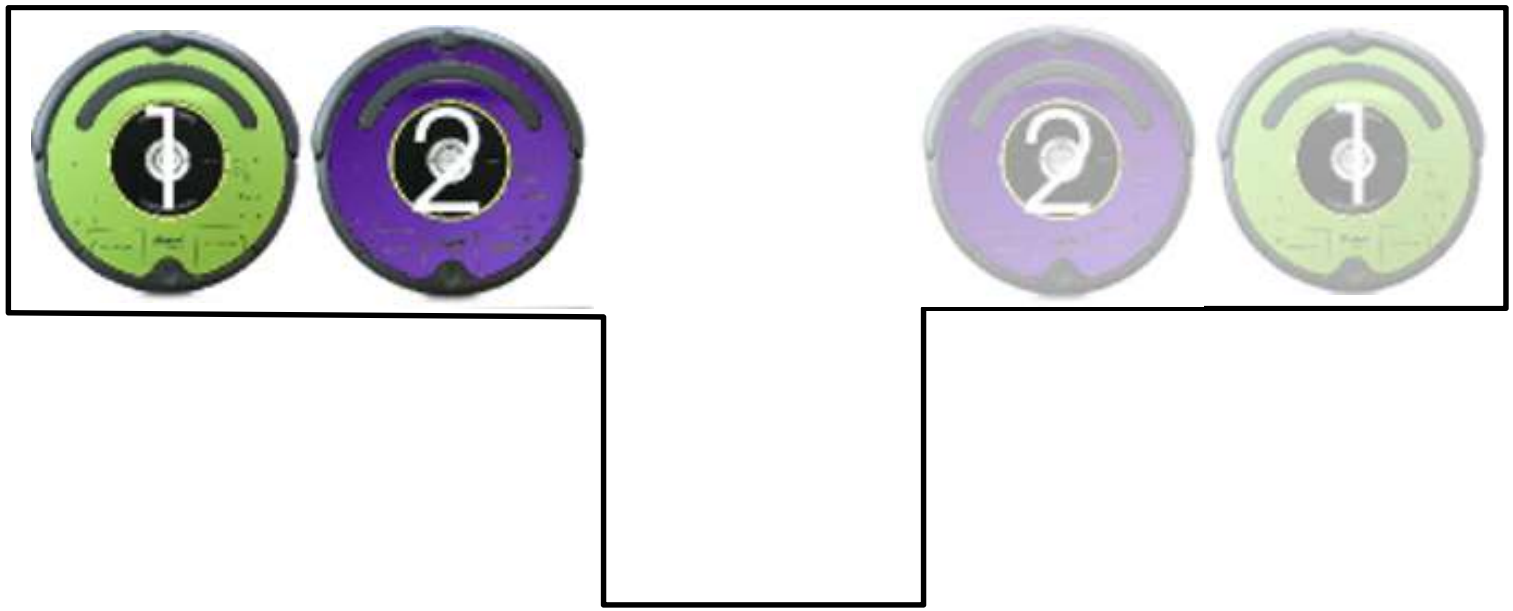
Multi-Agent Path Finding (MAPF)



Multi-Agent Path Finding (MAPF)



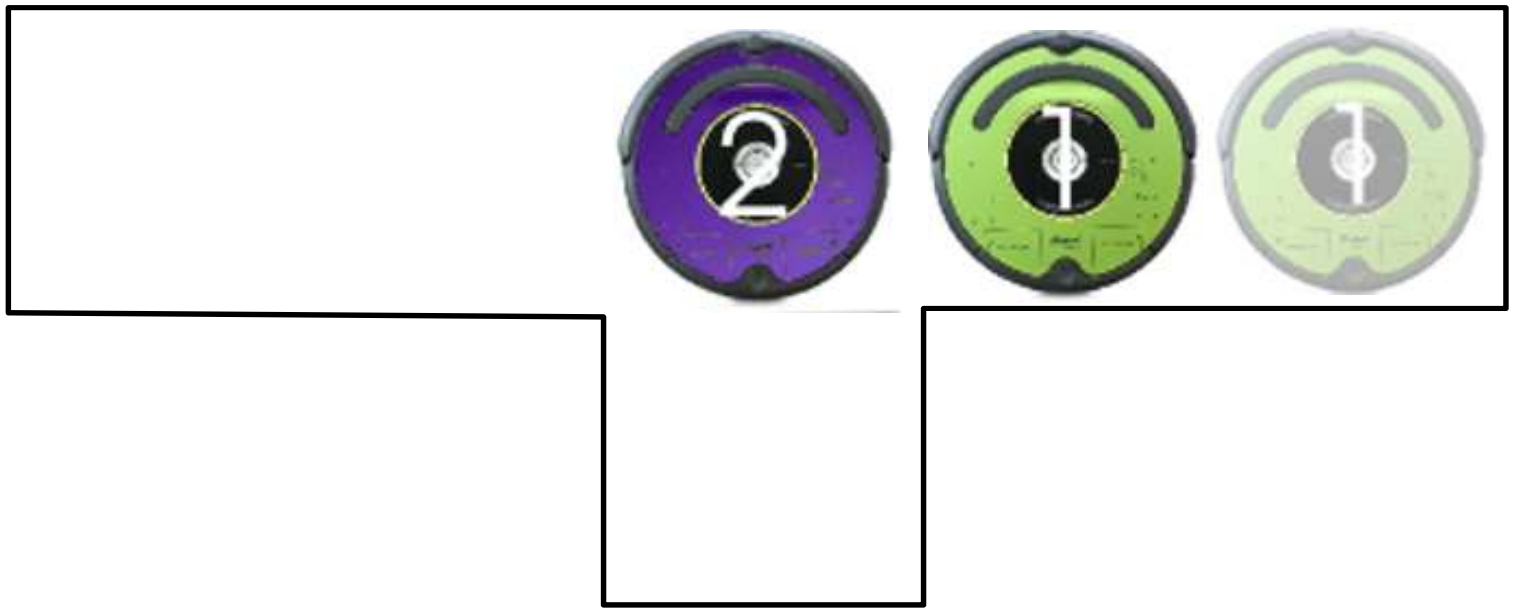
Multi-Agent Path Finding (MAPF)



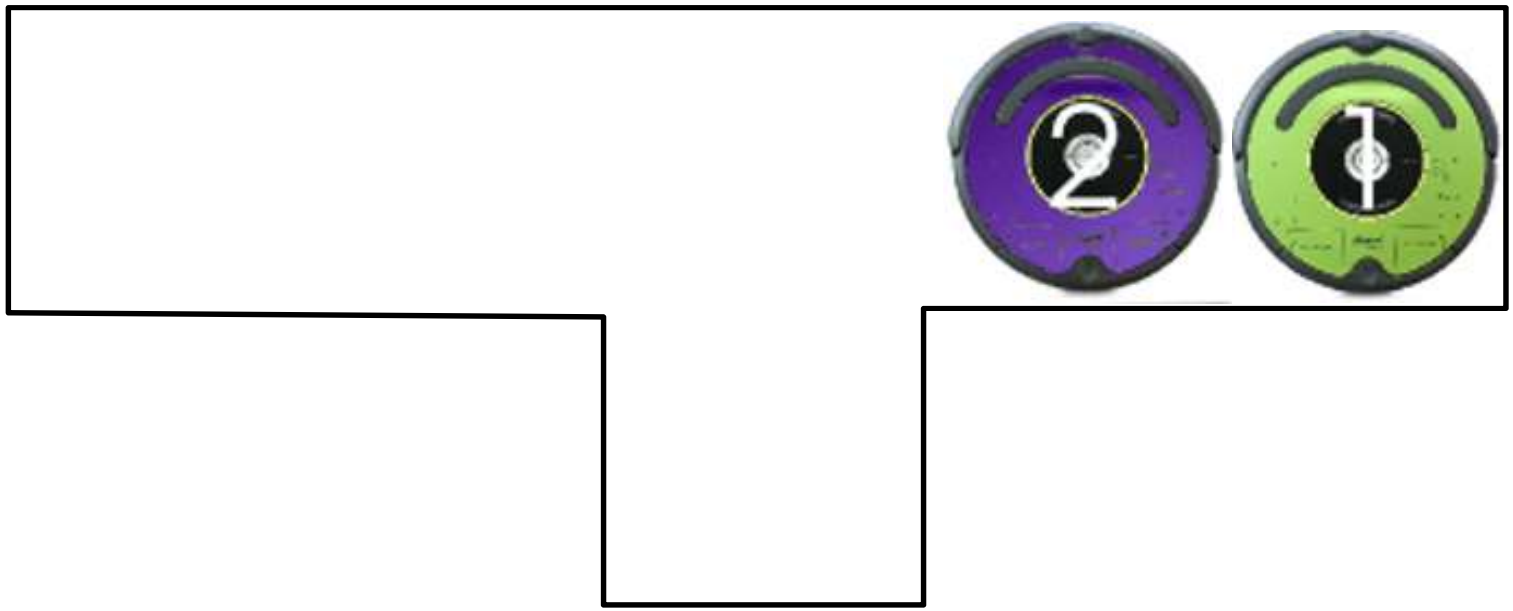
Multi-Agent Path Finding (MAPF)

Multi-Agent Path Finding (MAPF)

Multi-Agent Path Finding (MAPF)



Multi-Agent Path Finding (MAPF)

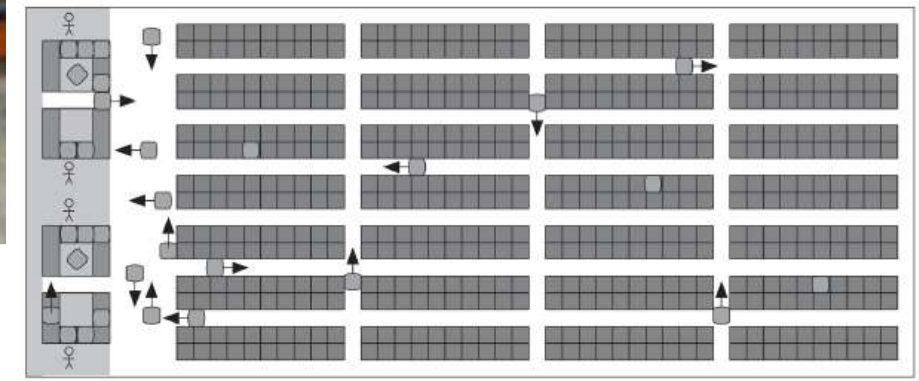


- Optimization problem with the objective
  to minimize task-completion time (called makespan) or
  the sum of travel times (called flowtime)

Multi-Agent Path Finding (MAPF)

- Application: Amazon fulfillment centers

amazon



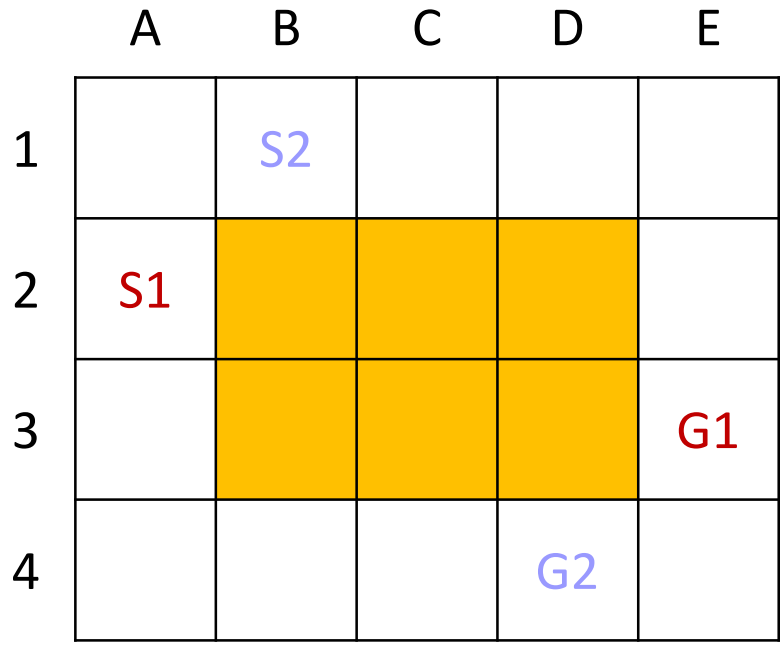
Multi-Agent Path Finding (MAPF)

- Optimal MAPF algorithms
  - Theorem [Yu and LaValle]: MAPF is NP-hard to solve optimally for both makespan or flowtime minimization

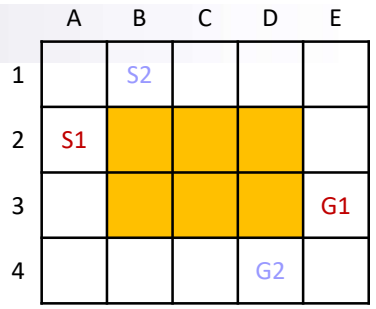
[www.random-ideas.net]

- Bounded-suboptimal MAPF algorithms
  - Theorem: MAPF is NP-hard to approximate within any factor less than 4/3 for makespan minimization on graphs in general

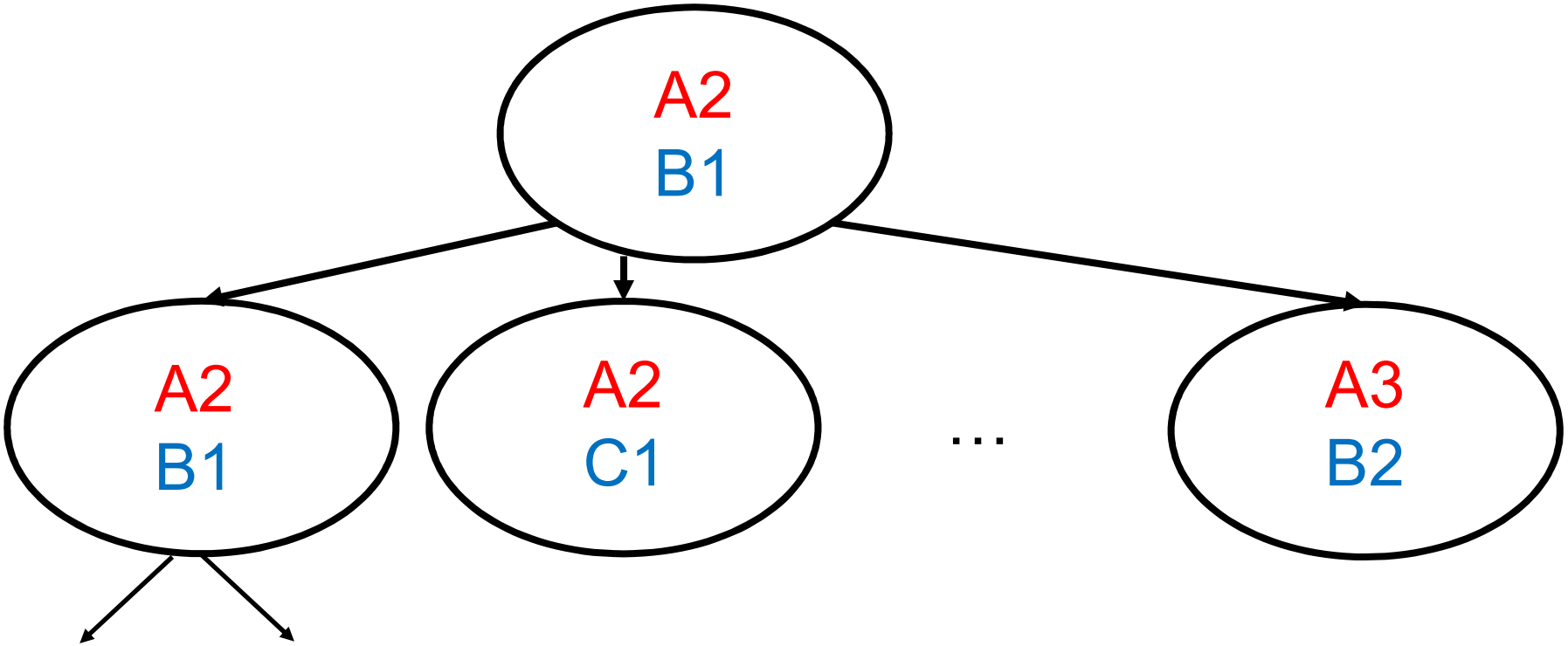
Multi-Agent Path Finding (MAPF)



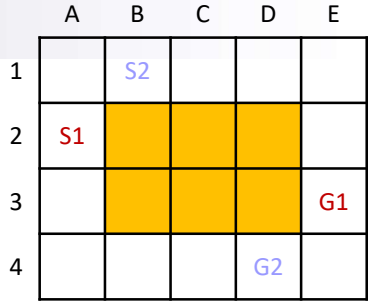
A*-Based Search



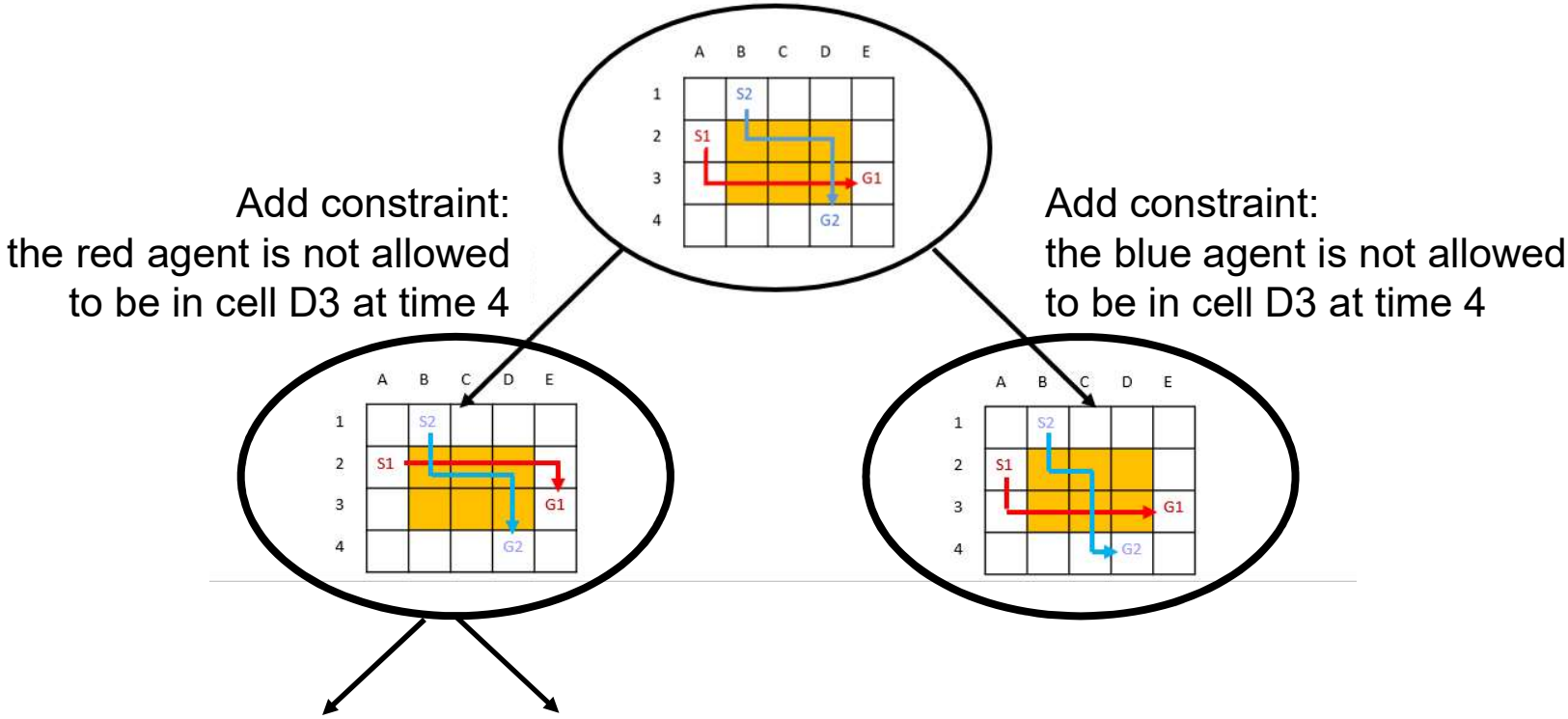
- A*-based search: Optimal (or bounded-suboptimal) MAPF solver



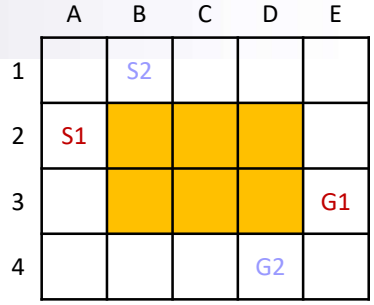
Conflict-Based Search



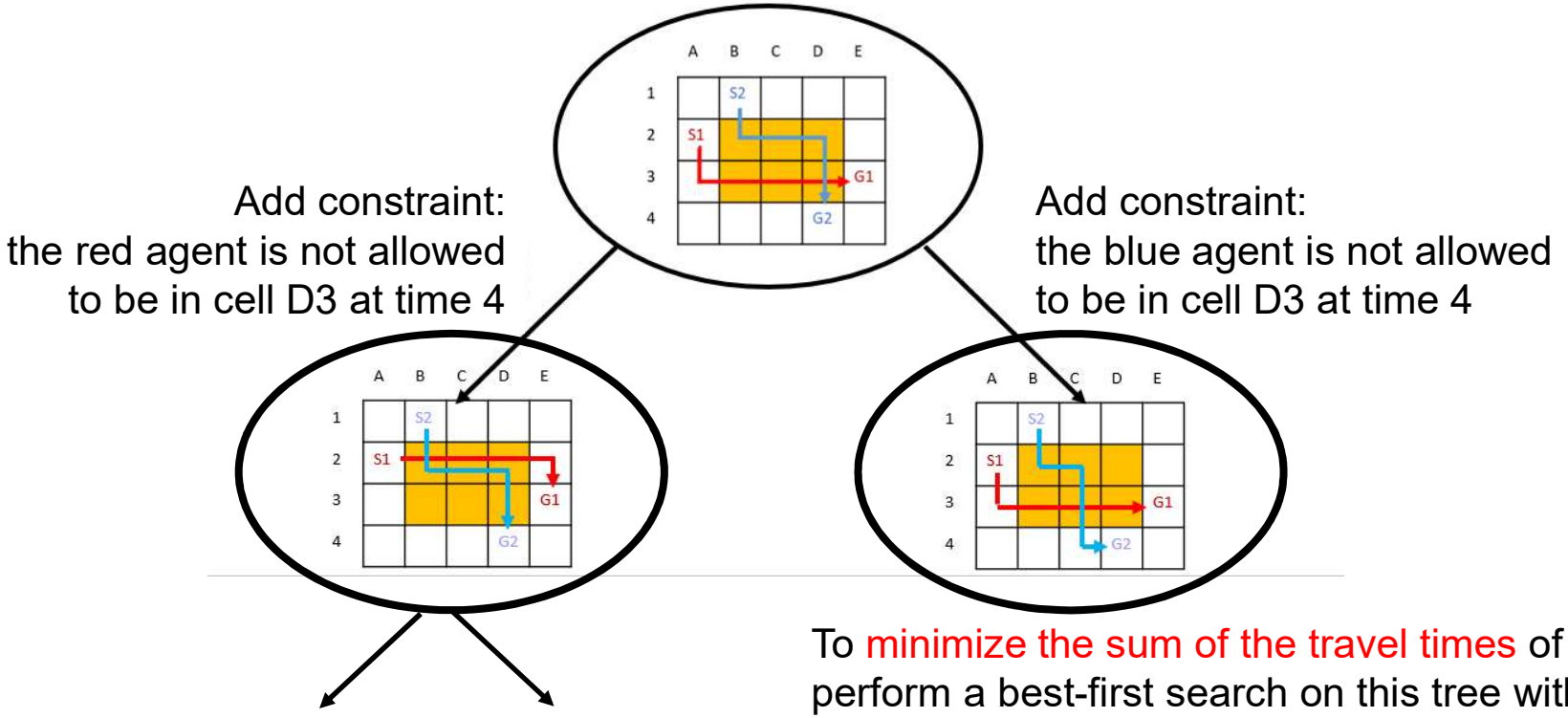
- Conflict-based search [Sharon, Stern, Felner and Sturtevant]: Optimal (or bounded-suboptimal) MAPF solver that plans for each agent independently



Conflict-Based Search

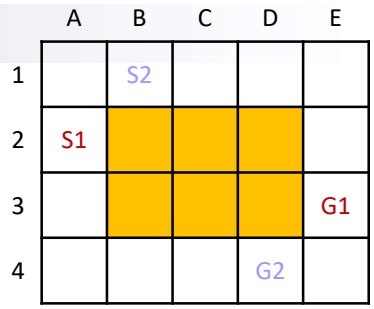


- Conflict-based search [Sharon, Stern, Felner and Sturtevant]: Optimal (or bounded-suboptimal) MAPF solver that plans for each agent independently



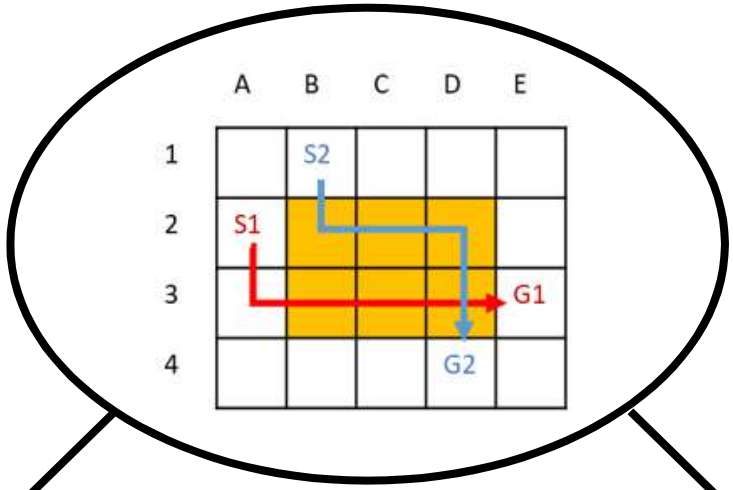
To minimize the sum of the travel times of all agents perform a best-first search on this tree with

- g = sum of travel times of all agents
- h = 0



Improvement 1

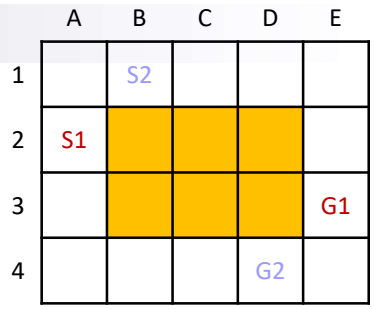
- Use more informed (= non-zero) h-values



Add constraint:
the red agent is not allowed
to be in cell D3 at time 4

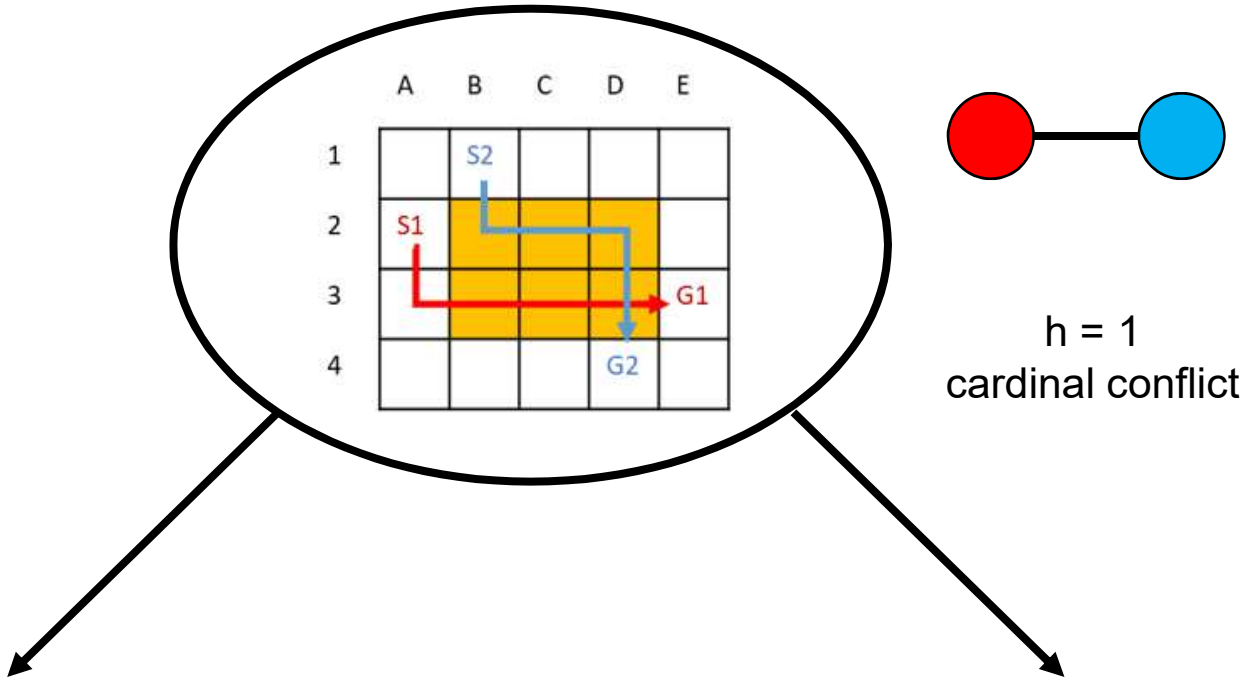
Add constraint:
the blue agent is not allowed
to be in cell D3 at time 4

The sum of travel times of any collision-free solution is at least 11.



Improvement 1

- Use more informed (= non-zero) h-values

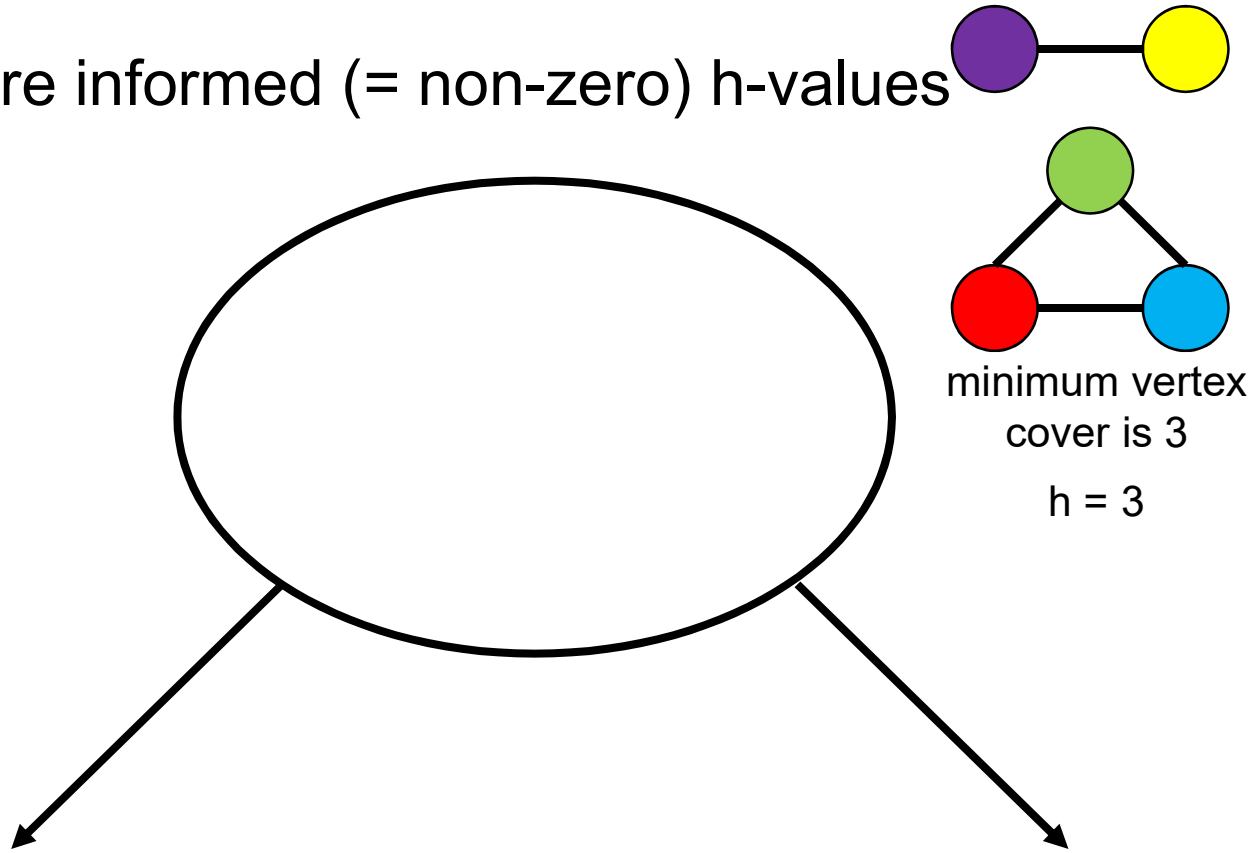


h = 1
cardinal conflict

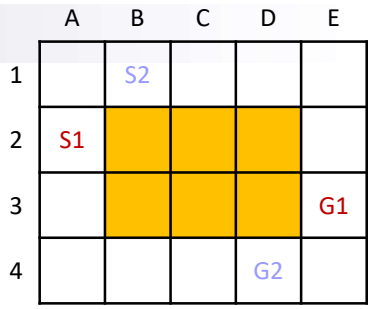
The sum of travel times of any collision-free solution is at least 11.

Improvement 1

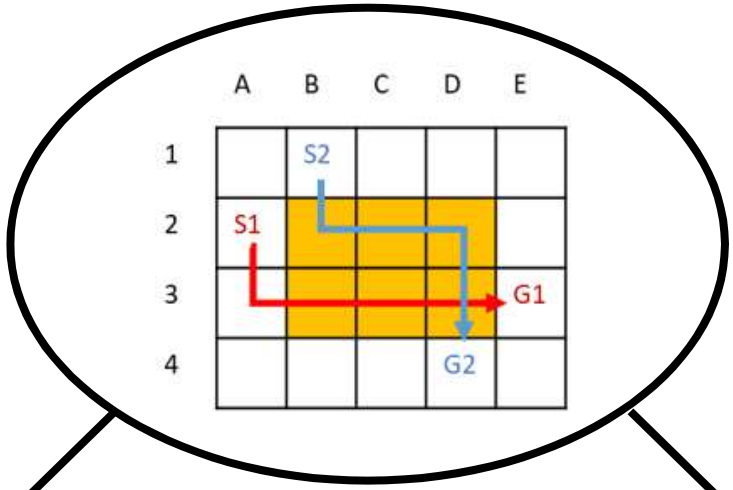
- Use more informed (= non-zero) h-values



Improvement 2



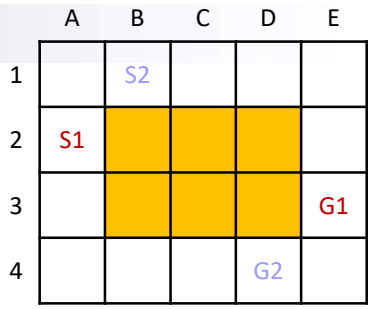
- Symmetry breaking of rectangle conflicts



Add constraint:
the red agent is not allowed to be in cell D3 at time 4

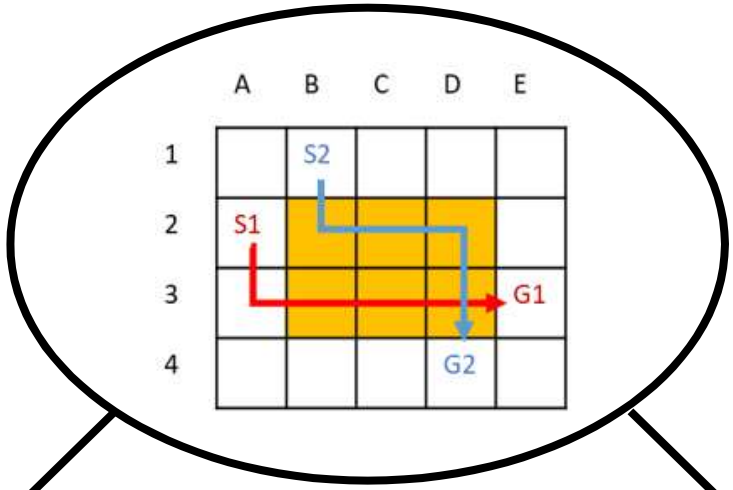
Add constraint:
the blue agent is not allowed to be in cell D3 at time 4

The sum of travel times of any collision-free solution is at least 11 but conflict-based search does not detect it right away.

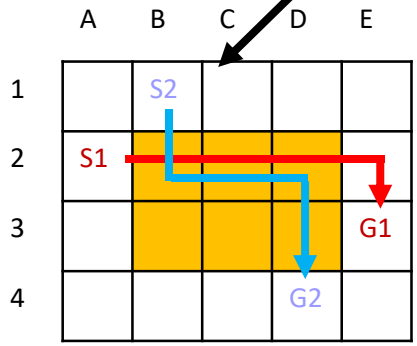


Improvement 2

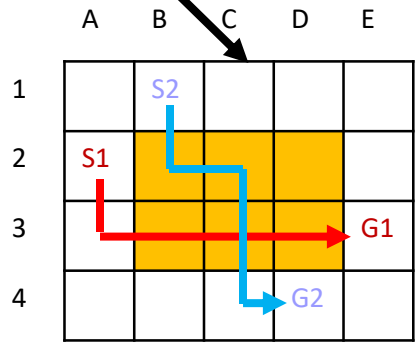
- Symmetry breaking of rectangle conflicts



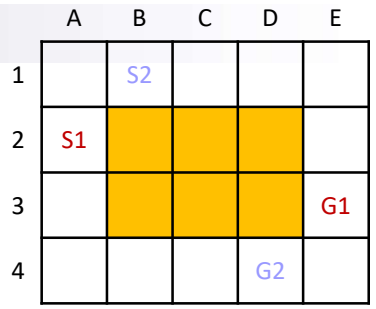
Add constraint:
the red agent is not allowed
to be in cell D3 at time 4



Add constraint:
the blue agent is not allowed
to be in cell D3 at time 4



Improvement 2



- Symmetry breaking of rectangle conflicts

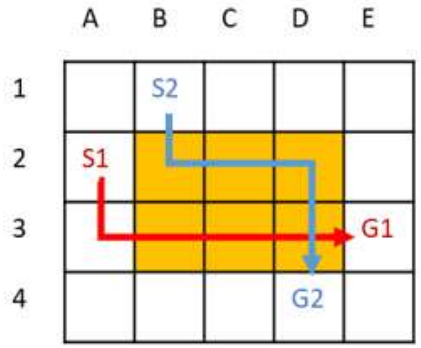
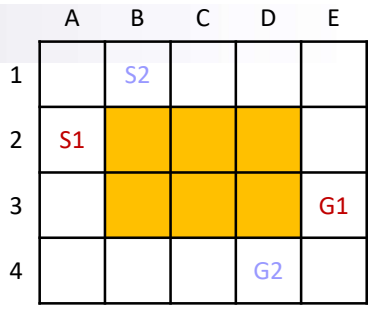


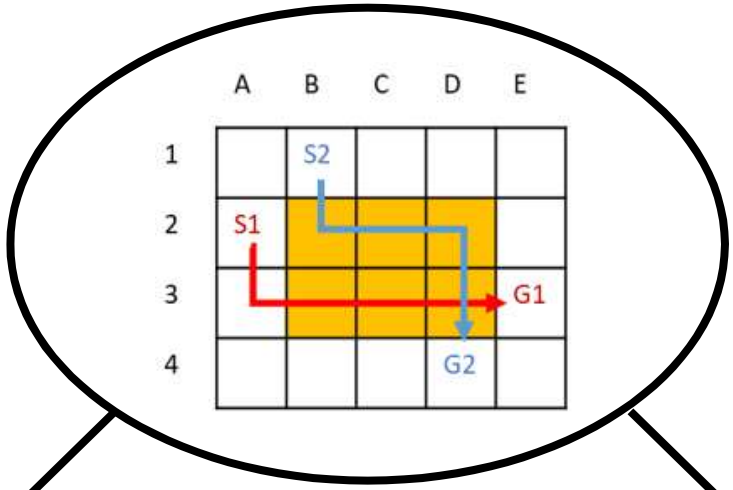
Table 1: Number of expanded CT nodes by CBSH on instances that 2 agents involve in cardinal rectangle conflicts. The first column and first row are the width and length of the rectangular area.

	1	2	3	4	5	6	7	8	9
1	1	1	2	3	4	5	6	7	8
2		3	7	14	26	46	79	133	221
3			22	53	116	239	472	904	1,692
4				142	392	1,016	2,651	6,828	17,747
5					1,015	2,971	8,525	23,733	65,236
6						7,447	24,275	78,002	254,173
7							62,429	222,524	795,197
8								573,004	>1,518,151

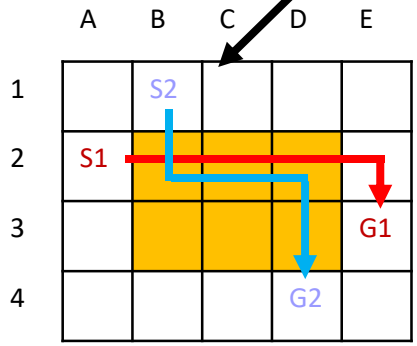


Improvement 2

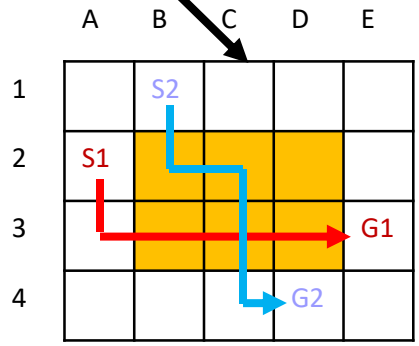
- Symmetry breaking of rectangle conflicts

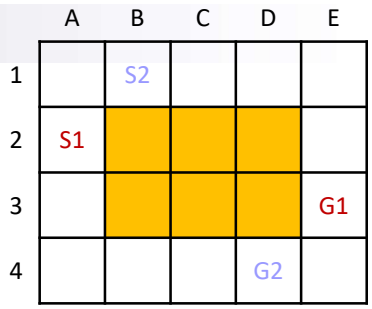


Add constraint:
the red agent is not allowed
to be in cell D3 at time 4



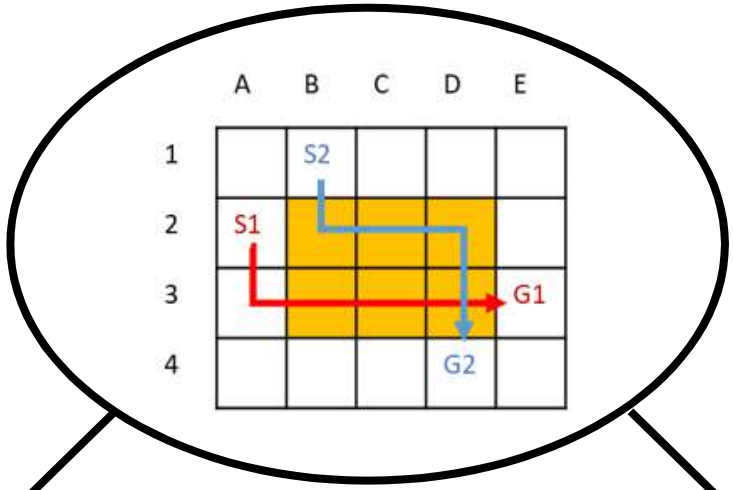
Add constraint:
the blue agent is not allowed
to be in cell D3 at time 4





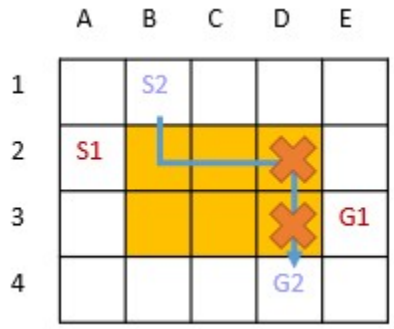
Improvement 2

- Symmetry breaking of rectangle conflicts

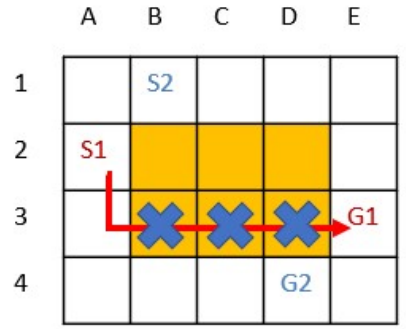


Add constraint:
the red agent is not allowed
to be in cell D3 at time 4
or in cell D2 at time 3

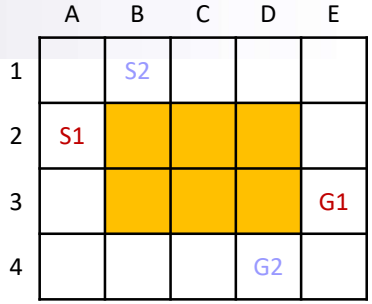
Add constraint:
the blue agent is not allowed
to be in cell D3 at time 4,
in cell C3 at time 3
or in cell B2 at time 2



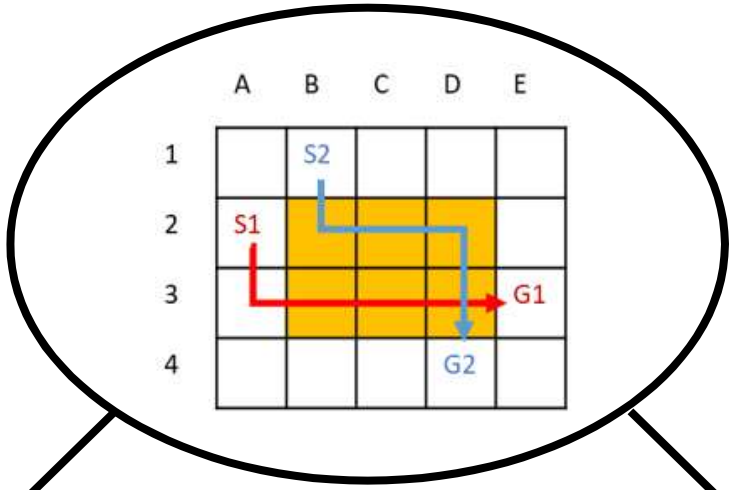
barrier constraints



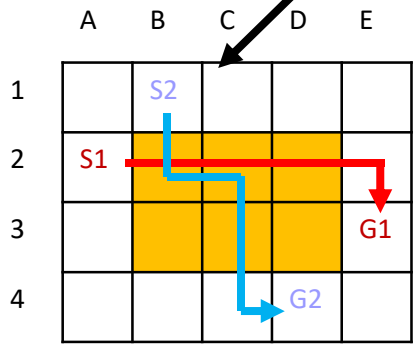
Improvement 3



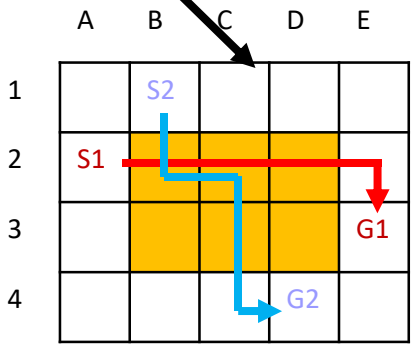
- Disjoint splitting



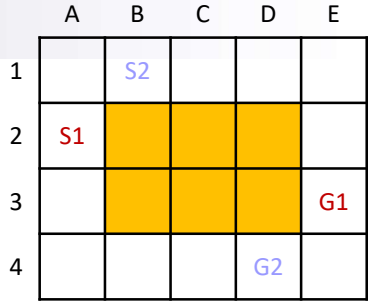
Add constraint:
the red agent is not allowed
to be in cell D3 at time 4



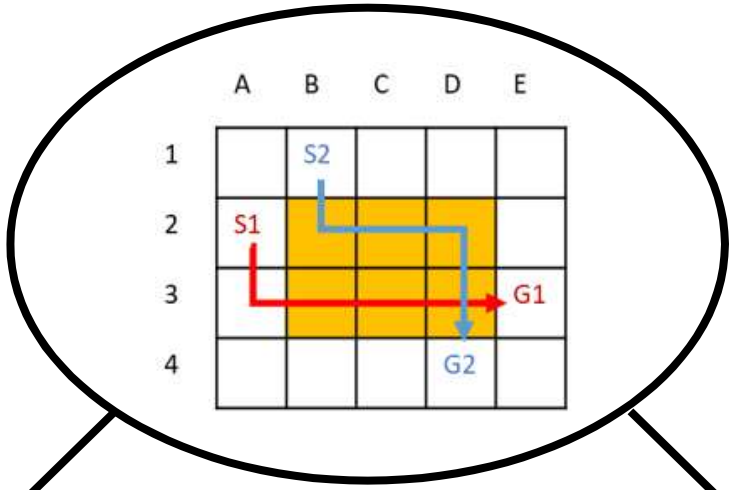
Add constraint:
the blue agent is not allowed
to be in cell D3 at time 4



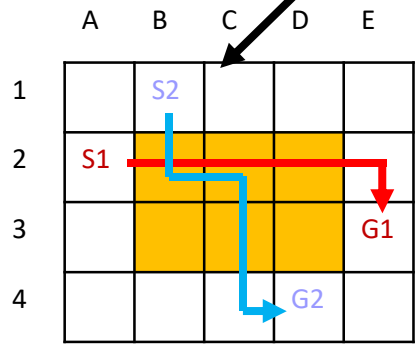
Improvement 3



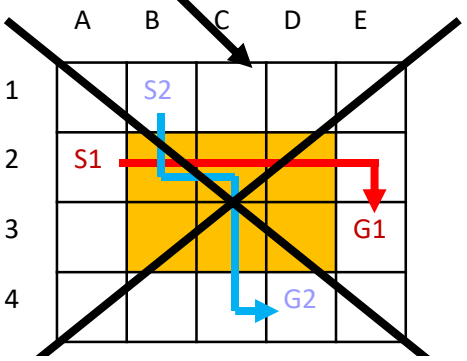
- Disjoint splitting

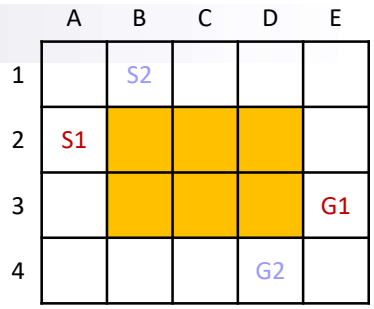


Add constraint:
the red agent is not allowed
to be in cell D3 at time 4



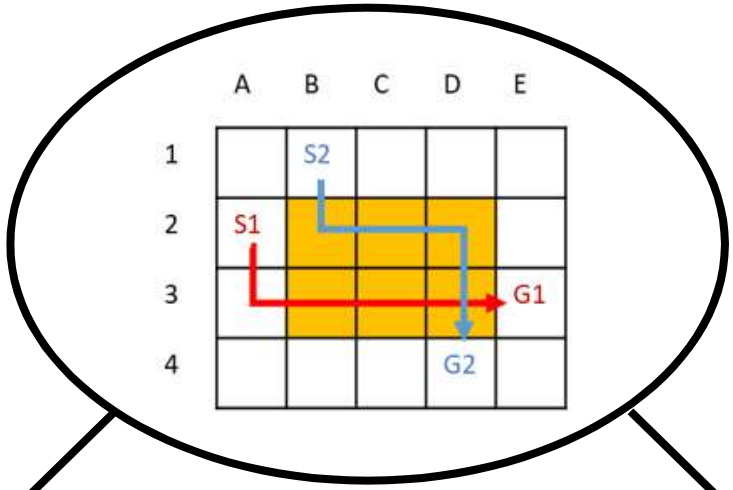
Add constraint:
the red agent must
be in cell D3 at time 4





Improvement 3

- Disjoint splitting

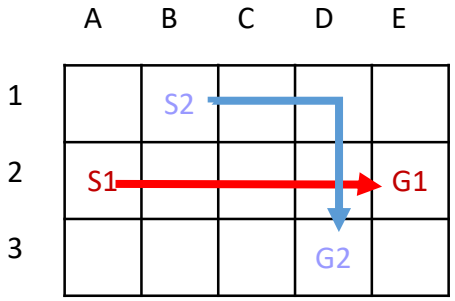


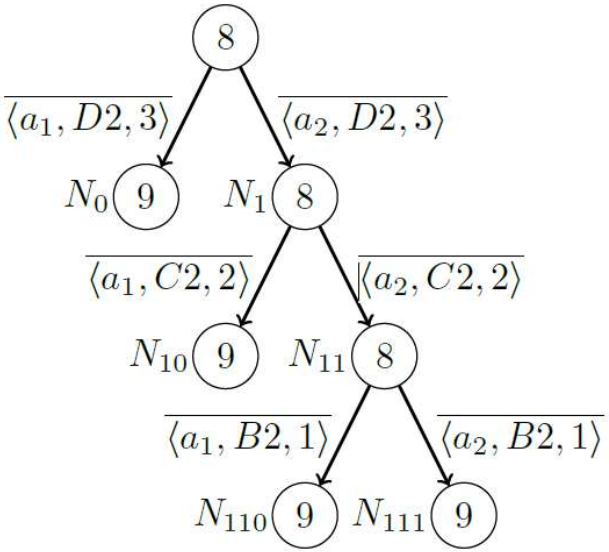
Add constraint:
the red agent is not allowed to be in cell D3 at time 4

Add constraint:
the red agent must be in cell D3 at time 4 which implies that all the other agents are not allowed to be in cell D3 at time 4

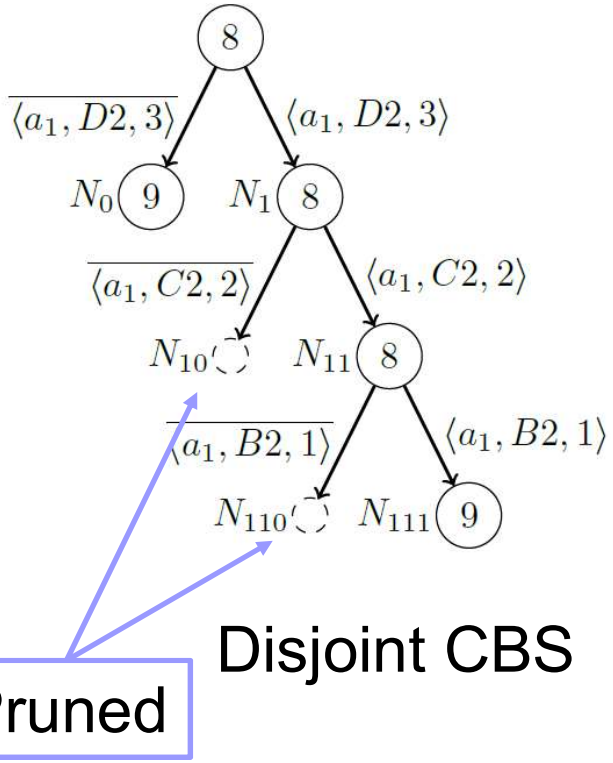
Improvement 3

- Disjoint splitting





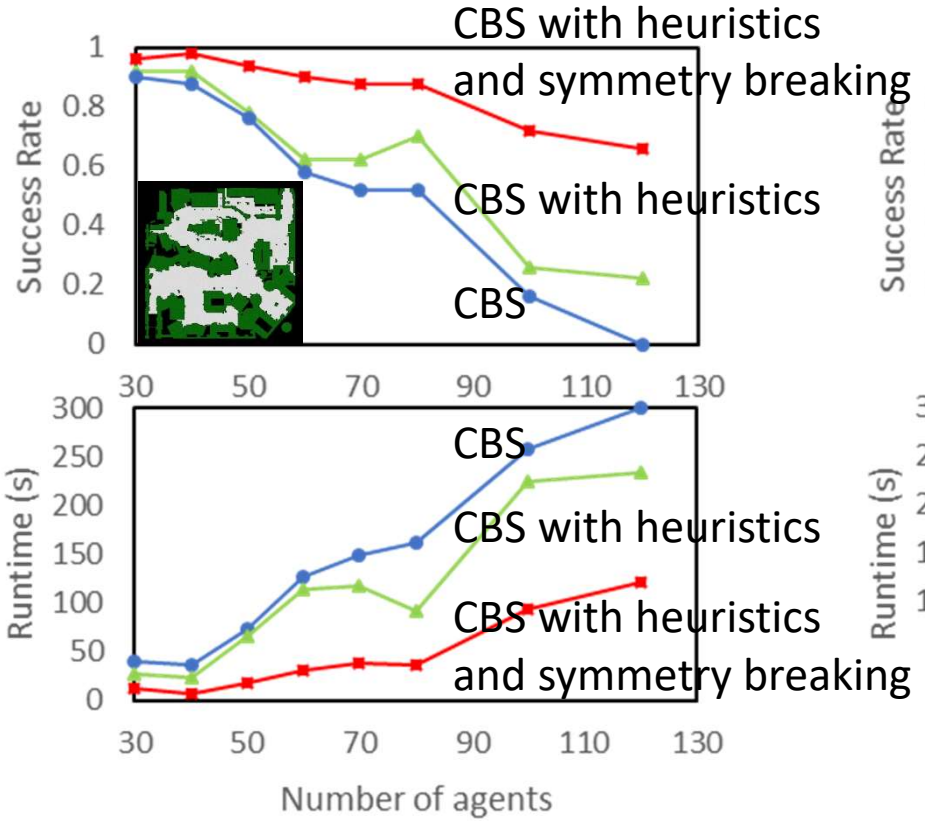
Original CBS



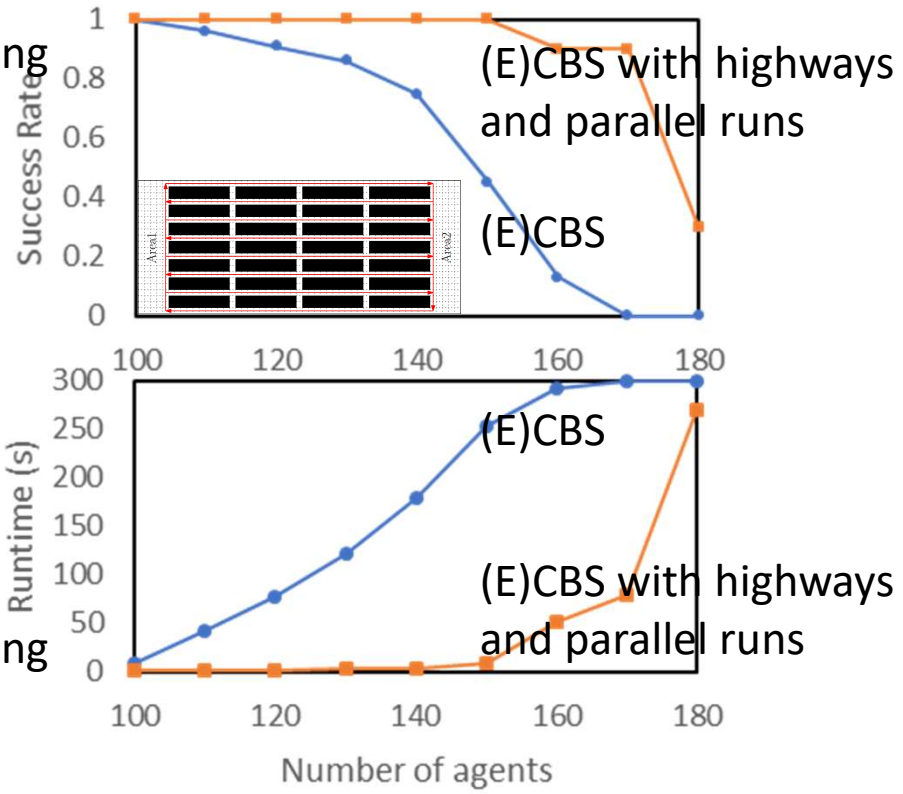
Disjoint CBS

Conflict-Based Search

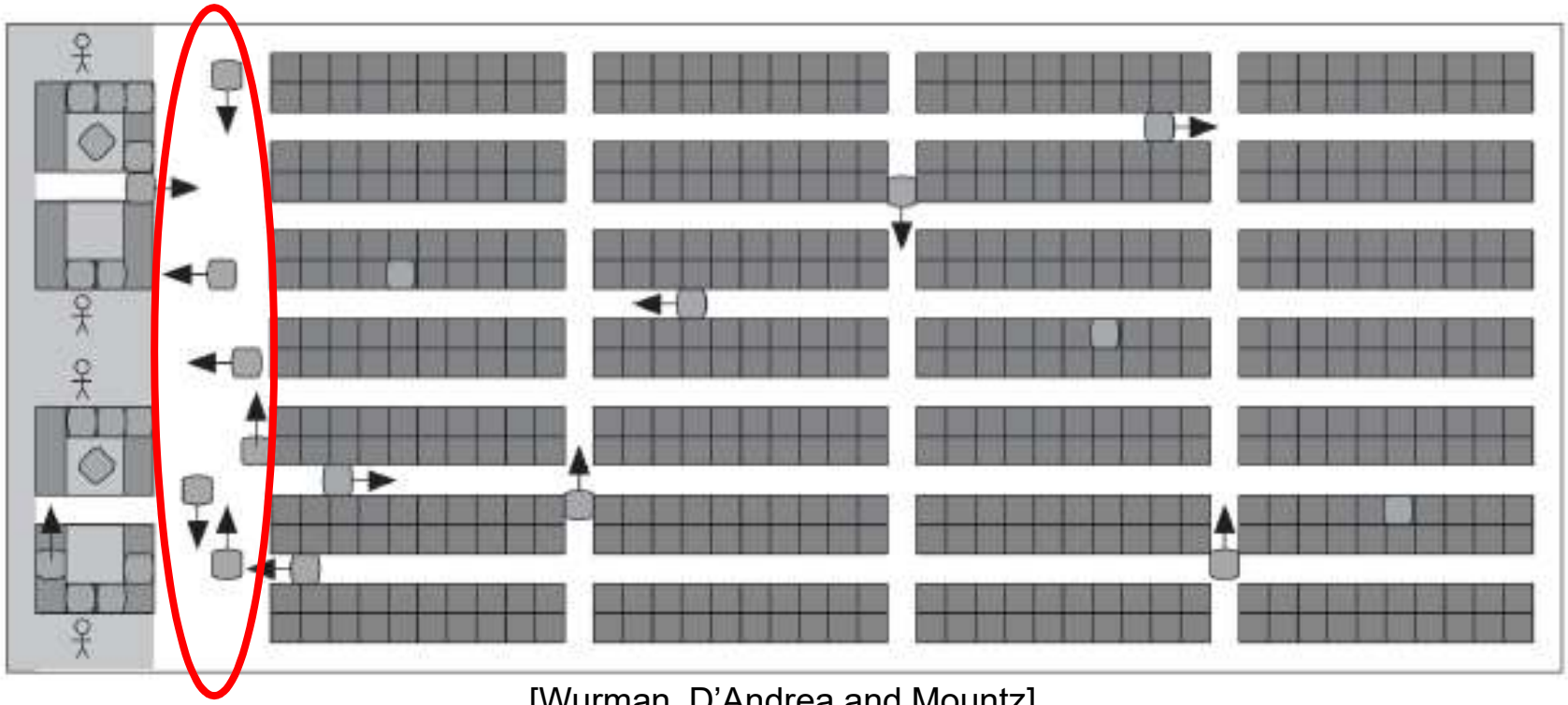
Optimal MAPF Planning



Bounded-Suboptimal MAPF Planning



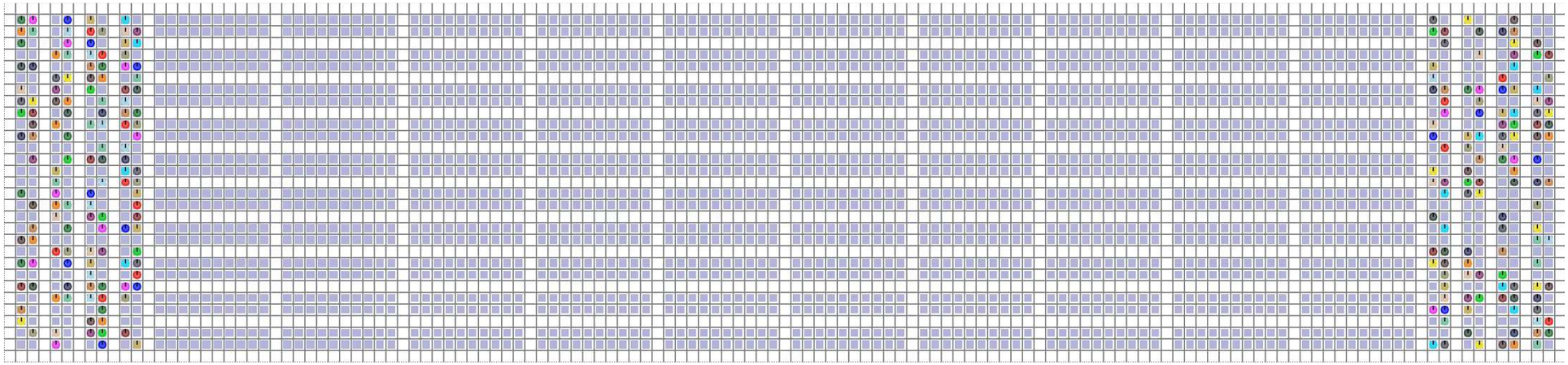
Conflict-Based Search



[Wurman, D'Andrea and Mountz]

Lifelong Multi-Agent Path Finding

- Runtime on 135x31 grids
  - 250 agents and 20,000 random pickup-and-delivery tasks
  - Makespan ≈ 0.5 hour
  - Mean total planning time ≈ 10s



Multi-Agent Path Finding (MAPF)

- For more information on multi-agent path finding, see http://mapf.info