New Techniques for Pairwise Symmetry Breaking in Multi-Agent Path Finding

Jiaoyang Li[1], Graeme Gange[2], Daniel Harabor[2], Peter J. Stuckey[2], Hang Ma[3], and Sven Koenig[1]

[1]University of Southern California

[2]Monash University

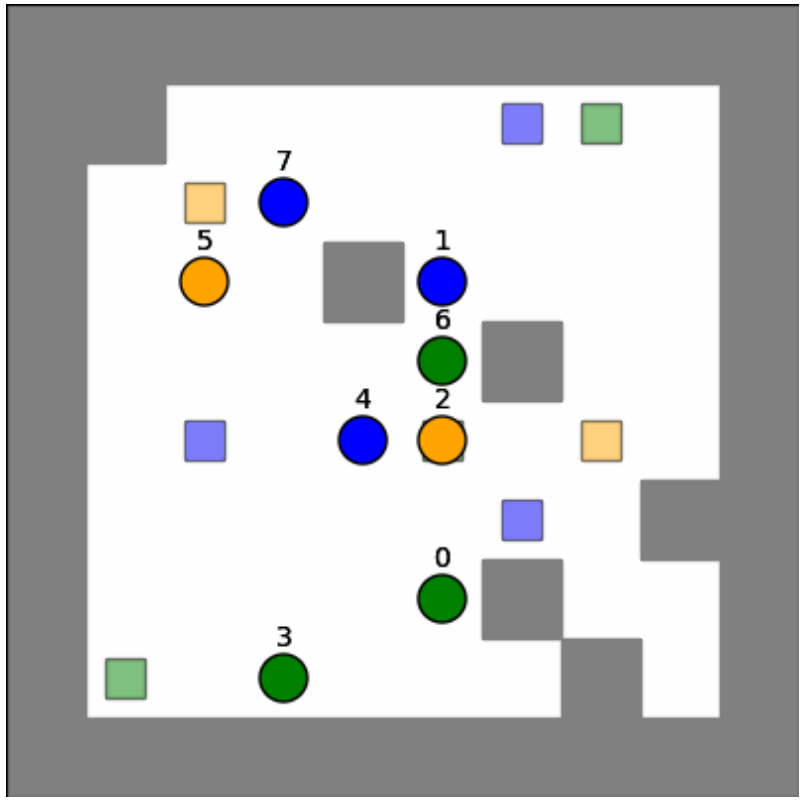
[3]Simon Fraser University

ICAPS 2020

Outline

- Problem definition
- Background:
  - Conflict-based search
  - Rectangle symmetry
- Corridor symmetry
- Target symmetry
- Empirical evaluation

Multi-Agent Path Finding (MAPF)

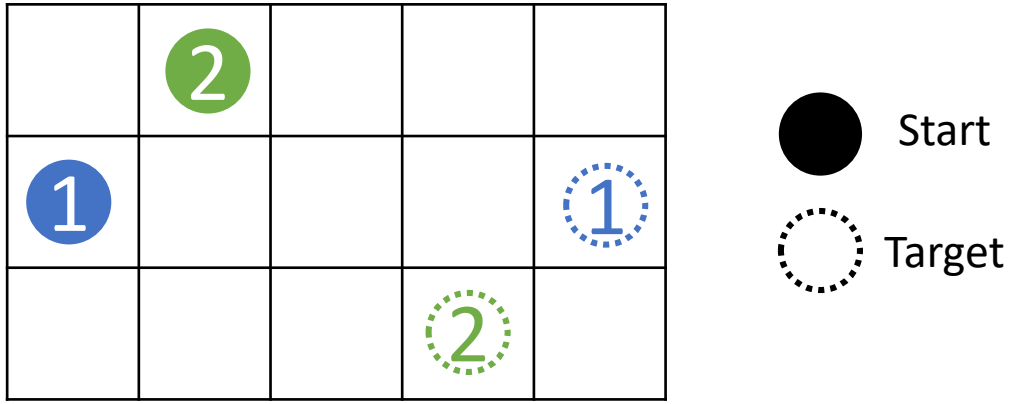


New Techniques for Pairwise Symmetry Breaking in Multi-Agent Path Finding
Jiaoyang Li, Graeme Gange, Daniel Harabor, Peter J. Stuckey, Hang Ma, and Sven Koenig

Figure and video sources:
[1] https://www.youtube.com/watch?v=8gy5tYVR-28&t=30s
[2] https://en.wikipedia.org/wiki/Cossacks:_European_Wars#/media/File:3_cossacks_european_wars.JPG
[3] https://futureoflife.org/wp-content/uploads/2019/04/Why-ban-lethal-AI-1030x595.jpg
[4] https://theconversation.com/we-can-design-better-intersections-that-are-safer-for-all-users-92178

Multi-Agent Path Finding (MAPF)

- **Given:**
  - A graph, and
  - A set of agents, each with a start location and a target location.



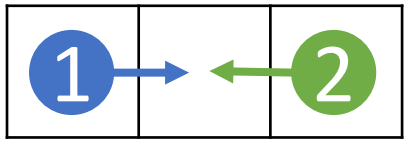
Multi-Agent Path Finding (MAPF)

Actions:

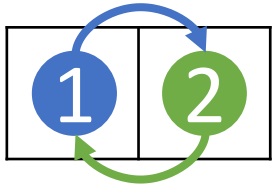
- *Move*: move to a neighboring location.
- *Wait*: wait at its current location.

Collisions:

- *Vertex collision*: two agents stay at the same location at the same timestep.

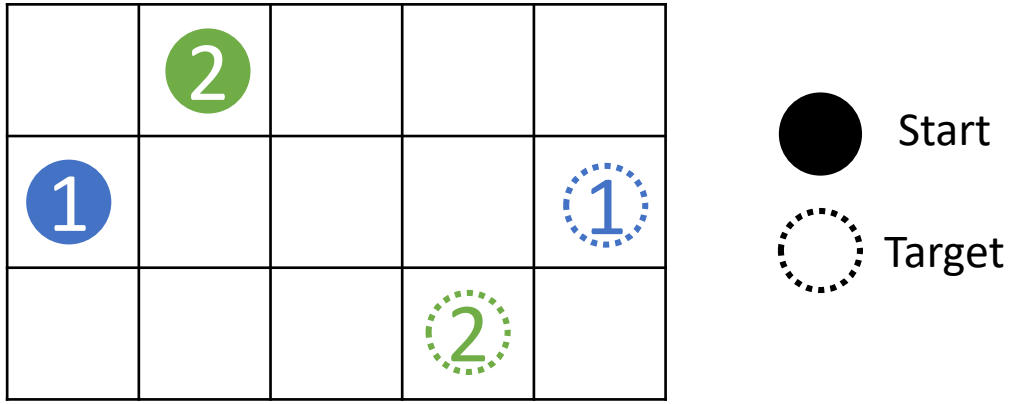


- *Edge collision*: two agents traverse the same edge in opposite directions at the same timestep.



Multi-Agent Path Finding (MAPF)

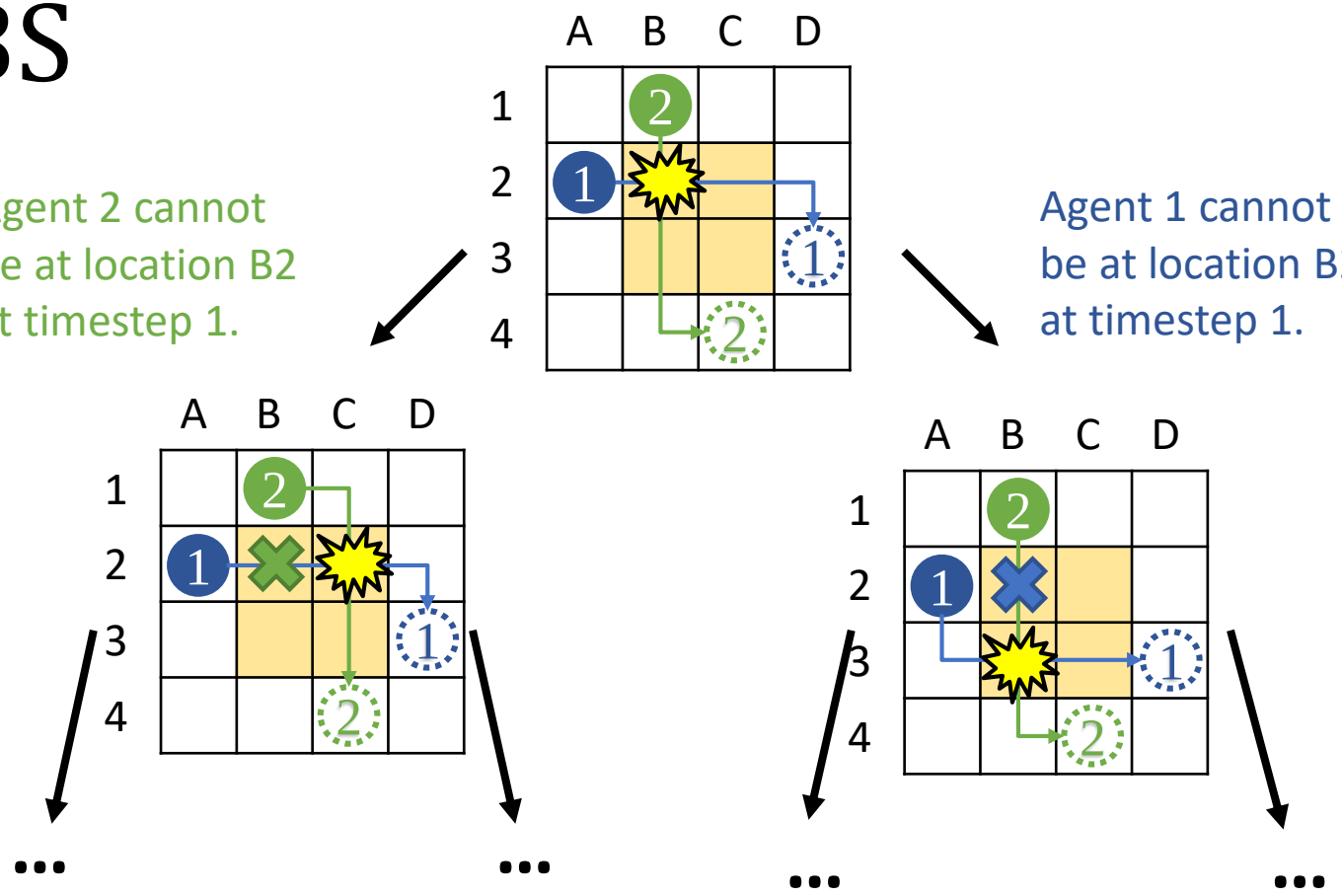
- **Given:**
  - A graph, and
  - A set of agents, each with a start location and a goal location.
- **Goal:**
  - Find collision-free paths for all agents, and
  - Minimize the sum of their travel times.



Multi-Agent Path Finding (MAPF)

- There are many optimal MAPF algorithms, such as
  - Search-based algorithms,
  - ILP-based algorithms,
  - SAT-based algorithms, and
  - CP-based algorithms.
- Most of the state-of-the-art variants of optimal MAPF algorithms (e.g., CBSH, BCP, SMT-CBS, lazy-CBS) deploy a strategy of planning paths individually first and resolving collisions afterward.
- Collision symmetries can lead to unacceptable runtimes if undetected.

CBS



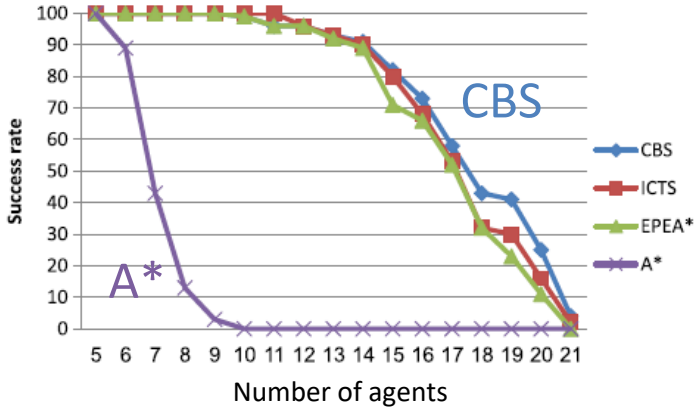
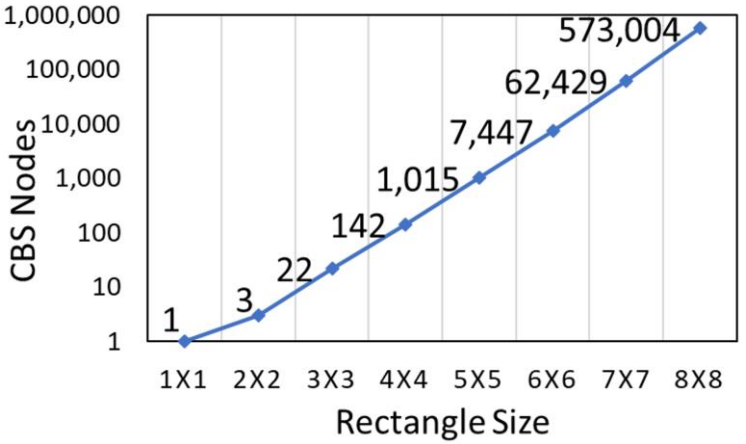
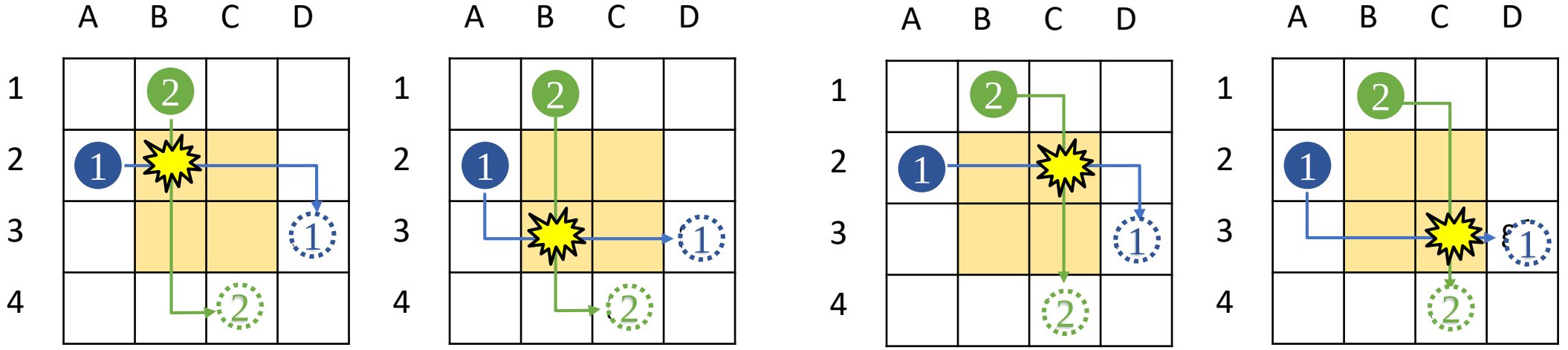


Fig. 8. Success rate vs. number of agents 8×8 grid.

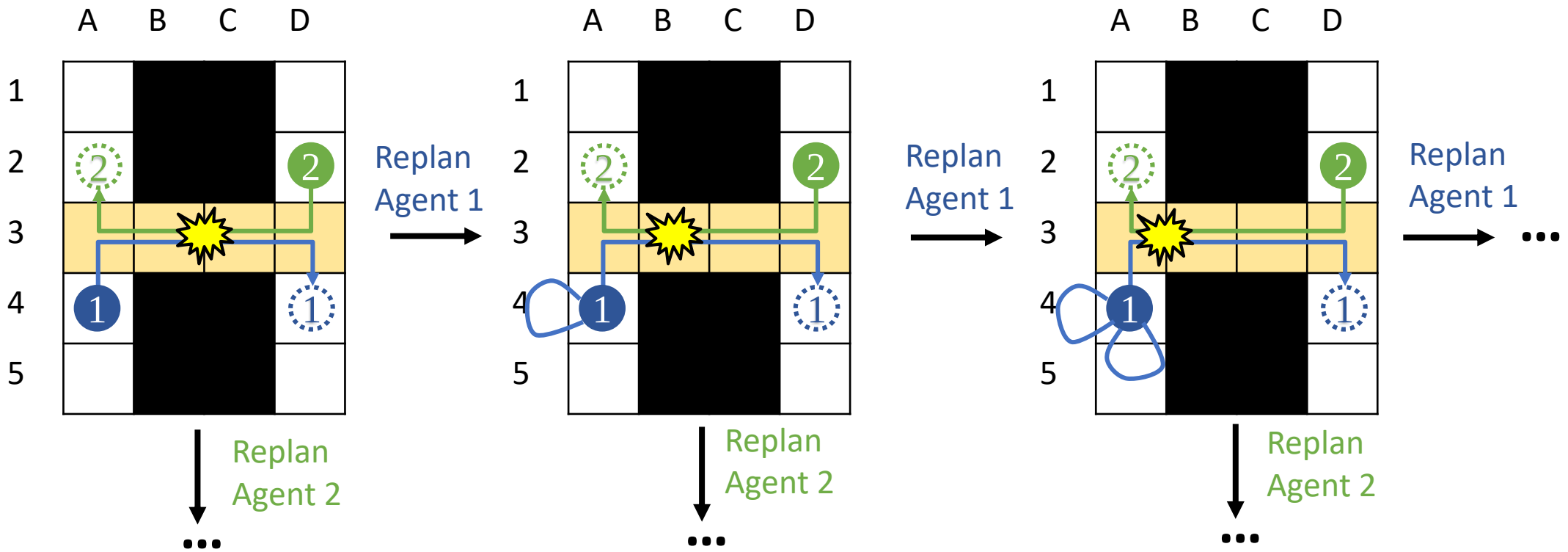
[Sharon et al, 2015]

Rectangle symmetry [AAAI 2019]

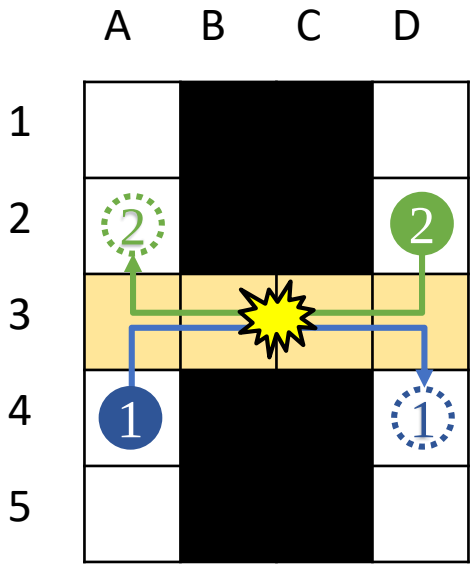


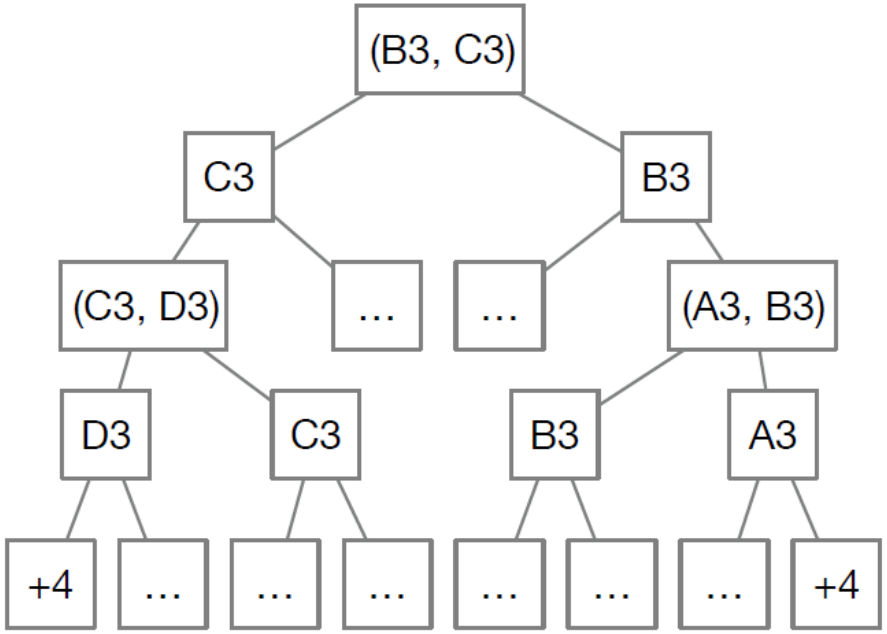


Corridor Symmetry



Corridor Symmetry



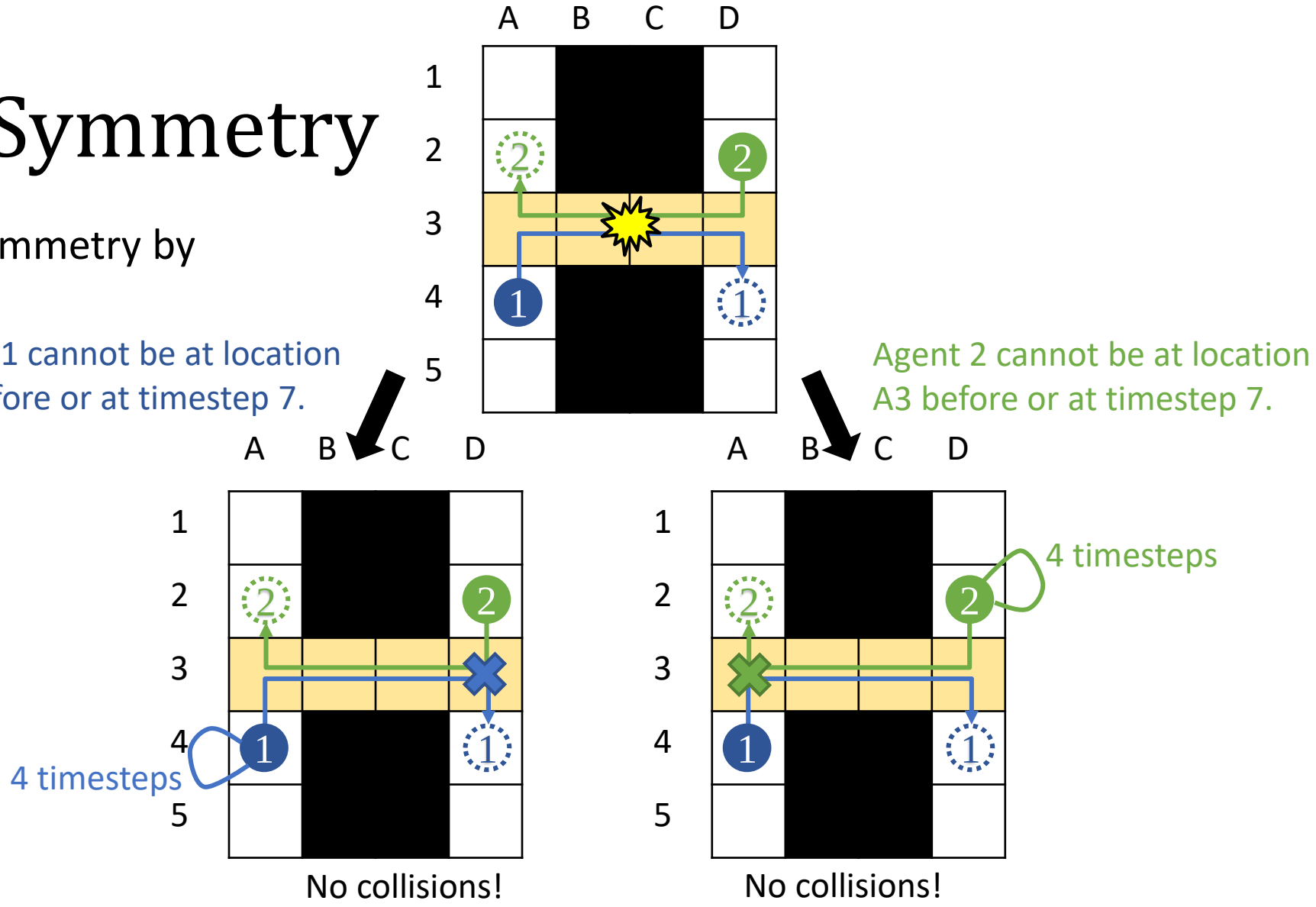


(nodes are denoted by the collision locations)

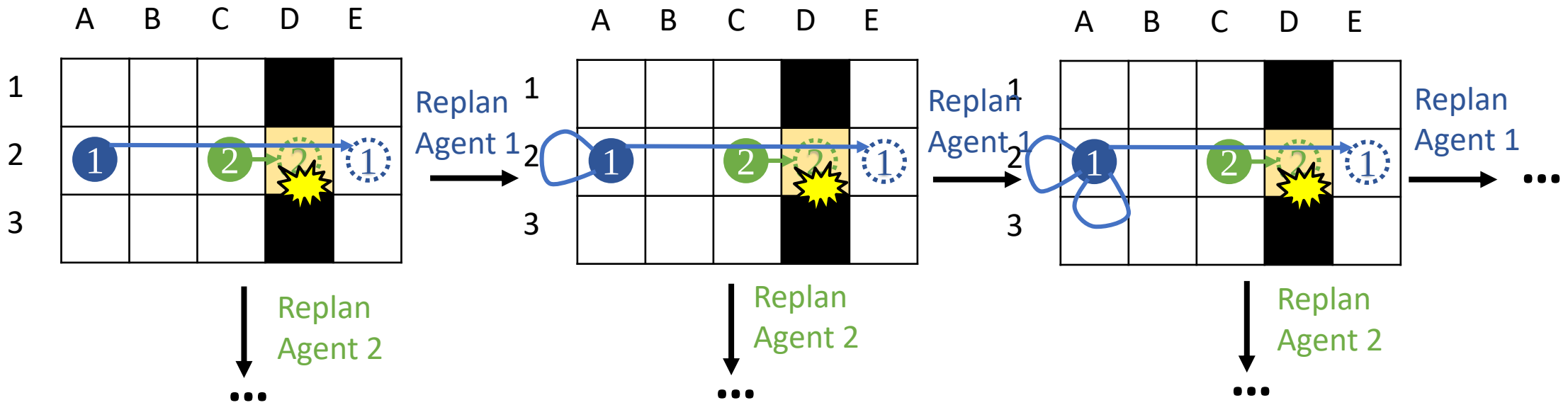
Corridor length	3	5	7	9	11	13	…	k
CBS nodes	16	64	256	1,024	4,096	16,384	…	2^{k+1}

Corridor Symmetry

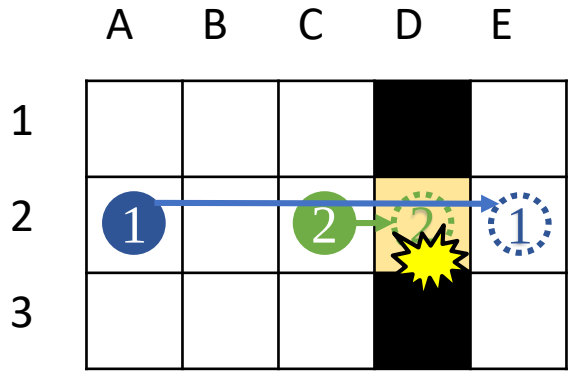
- Resolving corridor symmetry by **range constraints**

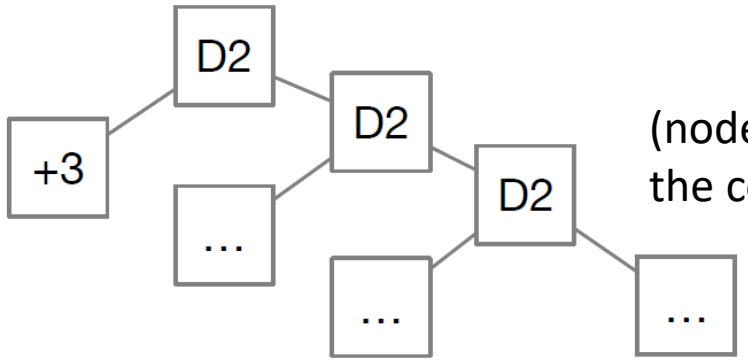


Target Symmetry



Target Symmetry

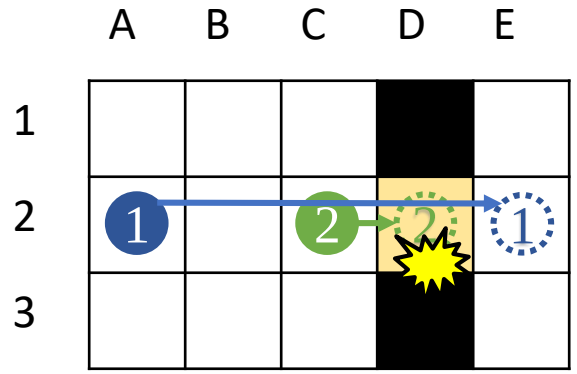




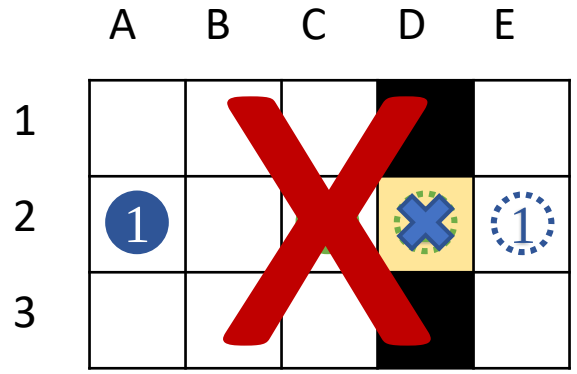
(nodes are denoted by the collision locations)

Target Symmetry

- Resolving target symmetry by **length constraints**

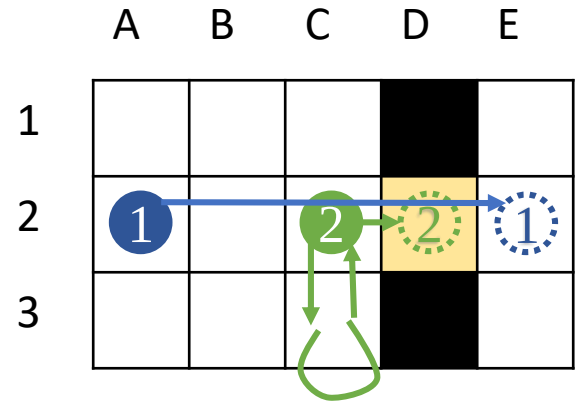


The length of Agent 2's path ≤ 3, which implies that Agent 1 cannot be at location D3 at or after timestep 3.



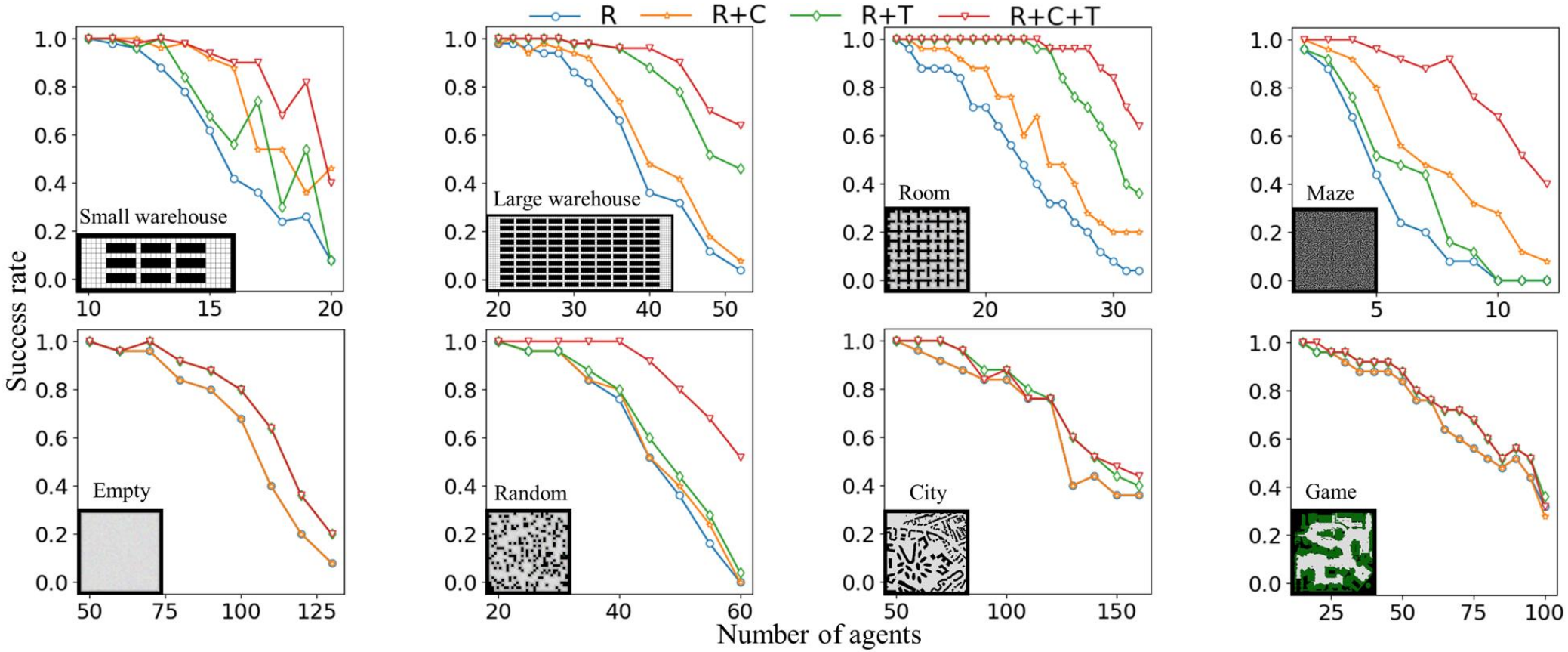
No solutions!

The length of Agent 2's path > 3.



No collisions!

Empirical Results



*Success rate = percentage of solved instances within one minute.

Summary

- **Corridor symmetry** arises when two agents attempt to pass through the same narrow corridor in opposite directions.
- **Target symmetry** arises when the shortest path of one agent passes through the target location of a second agent after the second agent has already arrived at it.
- We propose to use **range and length constraints** to eliminate corridor and target symmetries in a single branching step.
- We experimentally show that our techniques can, in some cases, **more than double the success rate** of CBS and **reduce its runtime by one order of magnitude**.