# Lifelong Multi-Agent Path Finding in Large-Scale Warehouses

(Extended Abstract)

Jiaoyang Li,[1] Andrew Tinka,[2] Scott Kiesel,[2] Joseph W. Durham,[2]

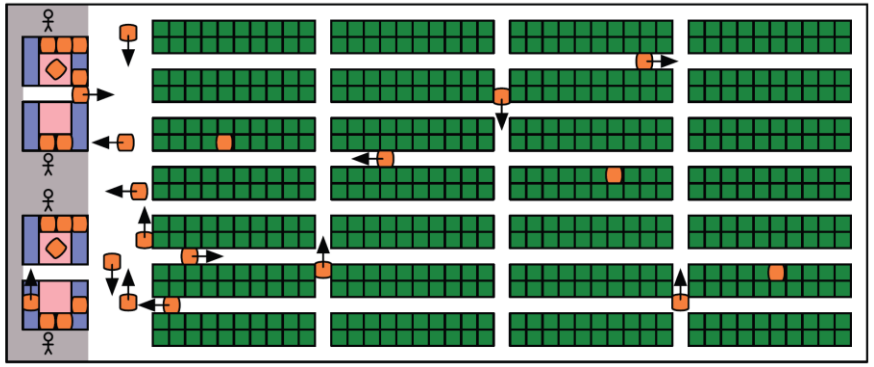
T. K. Satish Kumar[1] and Sven Koenig[1]

[1] University of Southern California

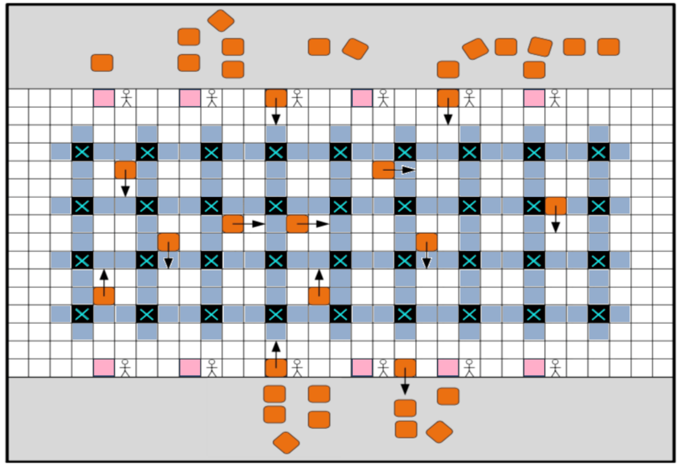
[2] Amazon Robotics

AAMAS-20

## Fulfillment center

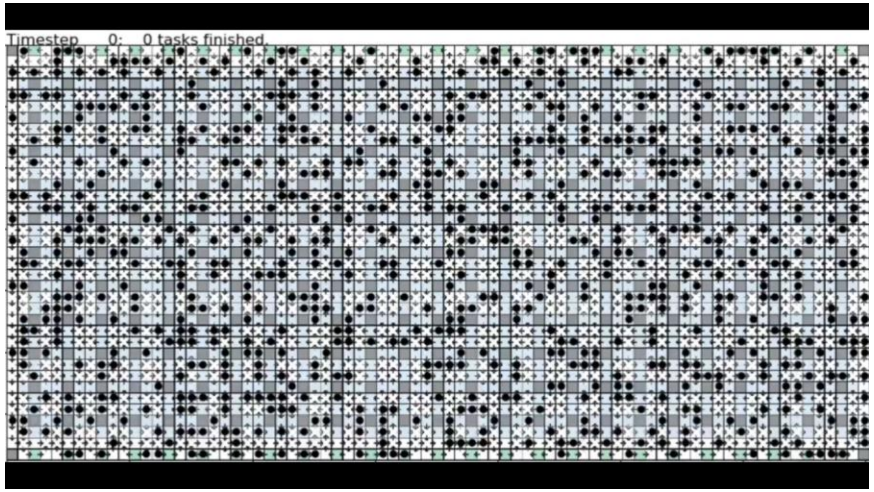



## Sorting center

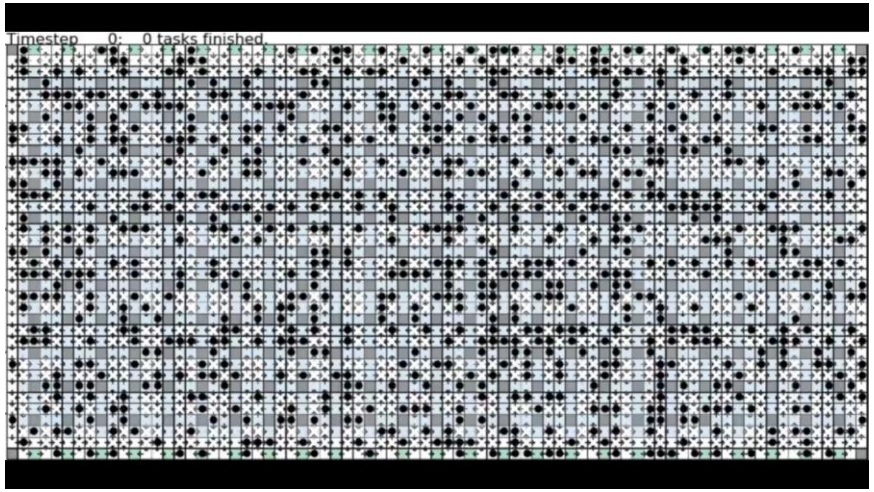


Video and picture sources:
[top left] High-speed robots part 1: meet bettybot in "human exclusion zone" warehouses. https://www.youtube.com/watch?v=8gy5tYVR-28&list=PL1JBGaGtAhqTLBCFWTB5pw6KghwkJhA3g&index=2&t=0s
[top right] Inside the amazon warehouse where humans and machines become one. https://www.wired.com/story/amazon-warehouse-robots/
[bottom left] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI), pages 1752–1760, 2007.
[bottom right] Qian Wan, Chonglin Gu, Sankui Sun, Mengxia Chen, Hejiao Huang, and Xiaohua Jia. Lifelong multi-agent path finding in a dynamic environment. In Proceedings of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 875–882, 2018.

## Traditional single-agent pathfinding solver




## Our multi-agent pathfinding solver




800 agents on a 37x77 sorting-center map with 50 working stations and 275 chutes.

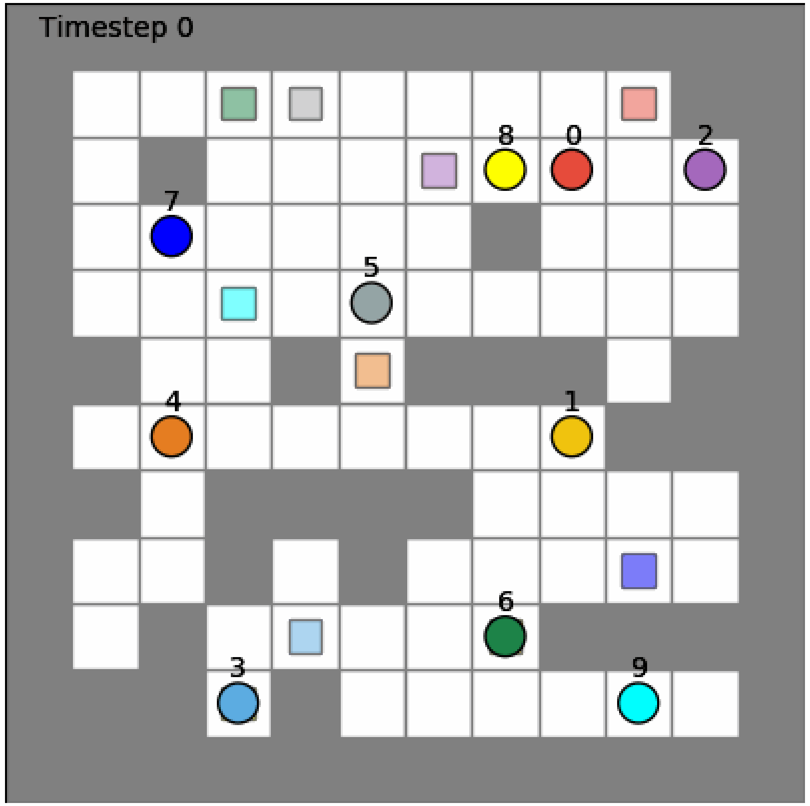
# Overview

- Multi-Agent Path Finding (MAPF) and lifelong MAPF
- Three existing methods for solving lifelong MAPF
  - Method 1: Solving lifelong MAPF as a whole.
  - Method 2: Solving a MAPF instance (incrementally) for all agents at every timestep.
  - Method 3: Solving a MAPF instance for a subset of agents at every timestep.
- Our method for solving lifelong MAPF
  - Solving a Windowed MAPF instance for all agents every $h$ timesteps.
- Experiments

# Multi-Agent Path Finding (MAPF)

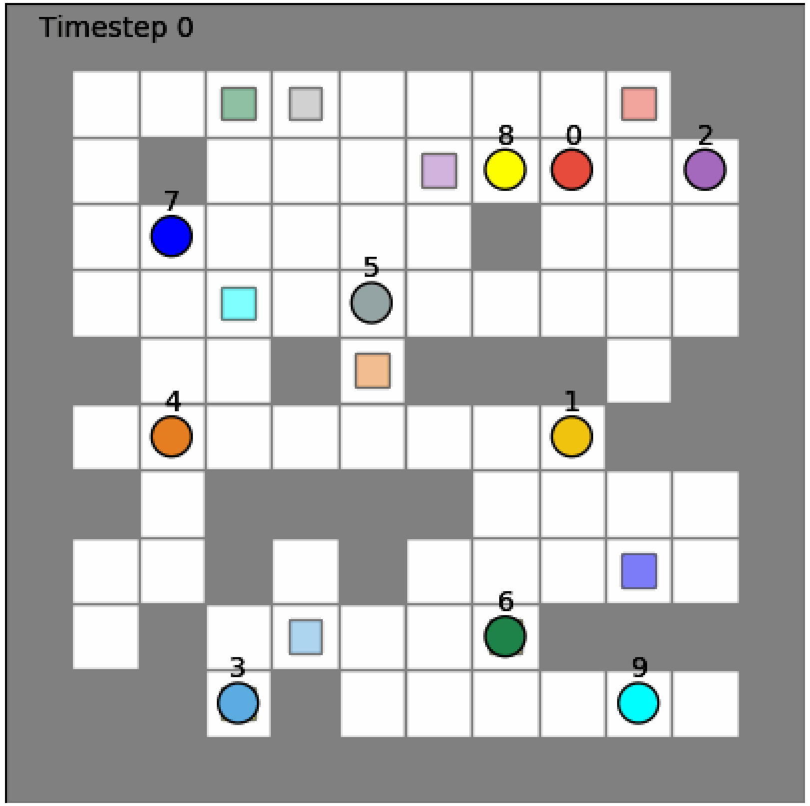
- **Inputs**
  - A graph
  - $m$ agents, each with
    - a start location,
    - a goal location.

- **Objective**
  - Finding a set of *collision-free* paths, one for each agent, while minimizing the sum of the travel times.



# Multi-Agent Path Finding (MAPF)

- **MAPF algorithms**
  - Complete and optimal
    - ICTS [Sharon et al 2011],
    - M* [Wagner et al 2011],
    - CBS [Sharon et al 2012],
    - EPEA* [Goldenberg et al 2014],
    - MDD-SAT [Surynek et al 2016],
    - BCP [Lam et al 2019].
  - Complete and suboptimal
    - BIBOX [Surynek 2009],
    - TASS [Khorshid et al 2011],
    - Push and Rotate [de Wilde et al 2014],
    - ECBS [Barer et al 2014],
    - ECBS with highways [Cohen et al 2015].
  - Incomplete
    - WHCA*[Silver 2005],
    - Push and Swap [Luna et al 2011],
    - PBS [Ma et al 2019],
    - PIBT [Okumura et al 2019],
    - DDM [Han et al 2020].



# Multi-Agent Path Finding (MAPF)

- **Lifelong MAPF**
  - Agents are constantly assigned new goal locations.

# Prior Work – Method 1

- Solving lifelong MAPF **as a whole** [Nguyen et al 2017].
  - Formulate lifelong MAPF as an answer set programming problem.

  - Drawbacks
    - Needs to know all goal locations a priori.
    - Has limited scalability.

[1] Van Nguyen, Philipp Obermeier, Tran Cao Son, Torsten Schaub, and William Yeoh. Generalized target assignment and path finding using answer set programming. In In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), pages 1216–1223, 2017.

# Prior Work – Method 2

- Solving a MAPF instance (incrementally) **for all agents at every timestep** [Wan et al 2018; Svancara et al 2019].
  - Start locations: current locations of all agents
  - Goal locations: next goal locations of all agents

  - Drawbacks
    - Needs to replan paths at every timestep (or at least at those timesteps when some agents have reached their goal locations).
    - Might do a lot of repeated or redundant work.

[1] Qian Wan, Chonglin Gu, Sankui Sun, Mengxia Chen, Hejiao Huang, and Xiaohua Jia. 2018. Lifelong multi-agent path finding in a dynamic environment. In Proceedings of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). 875–882.
[2] Jirı Svancara, Marek Vlk, Roni Stern, Dor Atzmon, and Roman Bartak. Online multi-agent pathfinding. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI), pages 7732–7739, 2019.

# Prior Work – Method 3

- Solving a MAPF instance **for only the agents with new goal locations at every timestep** [Cap et al 2015; Ma et al 2017; Liu et al 2019].
  - Start locations: current locations of agents with new goal locations
  - Goal locations: new goal locations
  - Drawbacks
    - Needs to plan paths at every timestep (or at least at those timesteps when some agents have reached their goal locations).
    - Could generate poor-quality solutions.
    - Only works for a special class of maps (i.e., well-formed maps).

[1] Michal Cap, Jirı Vokrınek, and Alexander Kleiner. Complete decentralized method for on-line multi-robot trajectory planning in well-formed infrastructures. In Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS), pages 324–332, 2015.
[2] Hang Ma, Jiaoyang Li, T. K. Satish Kumar, and Sven Koenig. Lifelong multi-agent path finding for online pickup and delivery tasks. In Proceedings of the 16th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), pages 837–845, 2017.
[3] Minghua Liu, Hang Ma, Jiaoyang Li, and Sven Koenig. Task and path planning for multi-agent pickup and delivery. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), pages 1152–1160, 2019.

# Our Method

- Solving a **Windowed MAPF** instance **for all agents every $h$ timesteps.**
  - In a Windowed MAPF instance,
    - collisions need to be resolved only for the first $w$ timesteps ($w \geq h$).
    - an agent might be assigned a sequence of goal locations. → Multi-Label A* [Grenouilleau et al 2019]

- Many existing MAPF solvers can be easily adapted to solve Windowed MAPF, e.g.,
  - CBS (complete and optimal),
  - ECBS (complete and bounded suboptimal),

  detect collisions only for the first $w$ timesteps
  - CA* (incomplete), → avoid collisions with higher-priority agents only for the first $w$ timesteps.
  - PBS (incomplete). → detect collisions only for the first $w$ timesteps, and avoid collisions with higher-priority agents only for the first $w$ timesteps.

[1] Florian Grenouilleau, Willem-Jan van Hoeve, and John N. Hooker. A multi-label A* algorithm for multi-agent pathfinding. In Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS), pages 181–185, 2019.

# Our Method

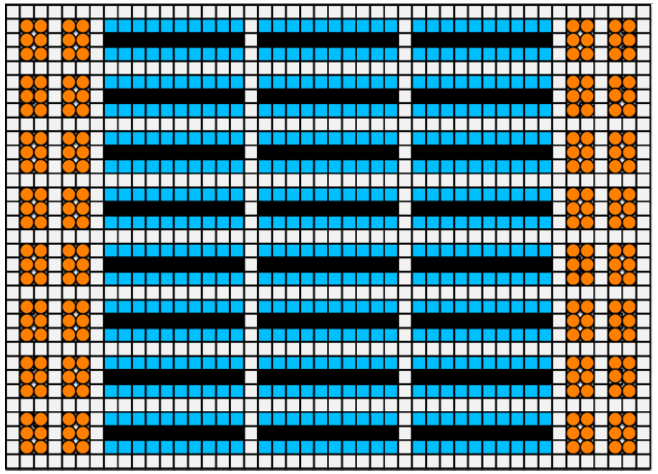
- Solving a **Windowed MAPF** instance **for all agents every $h$ timesteps.**
  - In a Windowed MAPF instance,
    - Collisions need to be resolved only for the first $w$ timesteps ($w \geq h$).
    - An agent might be assigned a sequence of goal locations.

- Advantages:
  1. Works for all kinds of maps.
  2. Does not have to replan paths at every timestep.
  3. Could significantly reduce the runtime of the solvers.
  4. Could still produce high-quality solutions.
     - because resolving all collisions within the entire time horizon is often unnecessary since the paths of the agents can change as new goal locations arrive.

# Experiment 1 – Fulfillment Center

A comparison with Method 3:

| Agents | Holding endpoints | | Dummy paths | | Our method | |
|---|---|---|---|---|---|---|
| | Throughput | Runtime (s) | Throughput | Runtime (s) | Throughput | Runtime (s) |
| 60 | 2.17 | **0.01** | 2.19 | 0.02 | **2.33** | 0.33 |
| 100 | 3.33 | **0.02** | 3.41 | 0.05 | **3.56** | 2.04 |
| 140 | 4.35 | **0.04** | 4.50 | 0.17 | **4.55** | 7.78 |



Well-formed map

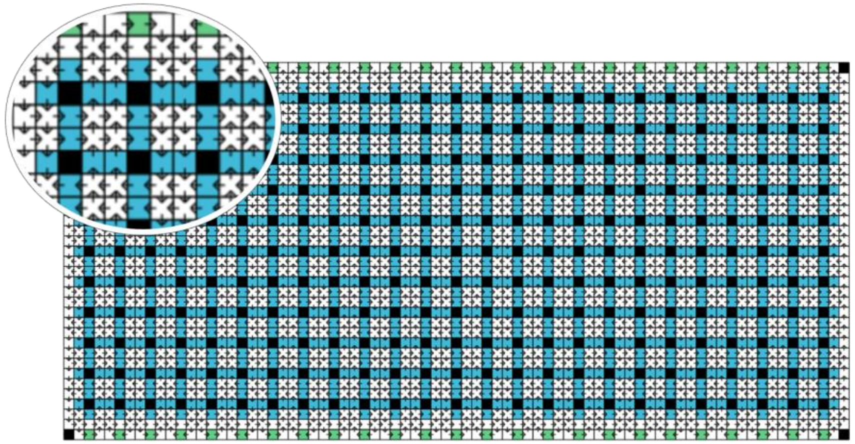
- All methods use PBS as the (Windowed) MAPF solver.
- Our method: resolving collsions for the first $w = 20$ timesteps and replanning paths every $h = 5$ timesteps.
- Throughput: average number of visited goal locations per timestep.
- Runtime: average runtime per run in seconds.

# Experiment 2 – Sorting Center

A comparison with different w:

| | PBS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Agents | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
| Throughput | $w = 5$ | 6.22 | 9.28 | 12.27 | 15.17 | 17.97 | 20.69 | 23.36 | 25.79 | 27.95 |
| | $w = 10$ | 6.27 | 9.36 | 12.41 | 15.43 | 18.38 | 21.19 | 23.94 | 26.44 | 28.77 |
| | $w = 20$ | 6.30 | 9.38 | 12.45 | 15.48 | 18.38 | 21.24 | 23.91 | - | - |
| | $w = \infty$ | 6.32 | 9.36 | 12.46 | 15.46 | 18.40 | 21.30 | - | - | - |
| Runtime (s) | $w = 5$ | 0.13 | 0.31 | 0.61 | 1.12 | 1.87 | 3.01 | 4.73 | 7.30 | 10.97 |
| | $w = 10$ | 0.16 | 0.42 | 0.89 | 1.66 | 2.91 | 4.81 | 7.79 | 12.66 | 21.31 |
| | $w = 20$ | 0.22 | 0.61 | 1.36 | 2.71 | 5.11 | 9.28 | 17.46 | - | - |
| | $w = \infty$ | 0.28 | 0.80 | 1.83 | 3.84 | 7.63 | 16.16 | - | - | - |

| | CA* | | | | CBS | | | ECBS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Agents | 200 | 300 | 400 | Agents | 100 | 200 | Agents | 400 | 500 | 600 |
| Throughput | $w = 5$ | 6.17 | 9.12 | - | $w = 5$ | 3.17 | - | $w = 5$ | 12.03 | 14.79 | 17.28 |
| | $w = \infty$ | 6.20 | 9.16 | - | $w = \infty$ | - | - | $w = \infty$ | 12.28 | 15.20 | - |
| Runtime (s) | $w = 5$ | 0.21 | 1.07 | - | $w = 5$ | 0.14 | - | $w = 5$ | 1.27 | 2.37 | 4.22 |
| | $w = \infty$ | 0.84 | 2.58 | - | $w = \infty$ | - | - | $w = \infty$ | 11.48 | 23.47 | - |



Not a well-formed map

- Replanning paths every $h = 5$ timesteps.
- "-" indicates that the runtime of the Windowed MAPF solver exceeds one minute per run.

# Summary

- Lifelong MAPF
  - Definition
  - Three existing methods

- Our method: Solving a Windowed MAPF instance for all agents every $h$ timesteps.
  - Works for all kinds of maps.
  - Does not have to replan paths at every timestep.
  - Could significantly reduce the runtime of the solvers.
  - Could still produce high-quality solutions
  - Scales up to 1,000 agents in simulated sorting centers.

# References for Algorithms on Slide 7


[1] Guni Sharon, Roni Stern, Meir Goldenberg, and Ariel Felner. The increasing cost tree search for optimal multi-agent pathfinding. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), pages 662 – 667, 2011.

[2] Glenn Wagner, and Howie Choset. M*: A complete multirobot path planning algorithm with performance bounds. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) , pages 3260-3267, 2011.

[3] Sharon, Guni, Roni Stern, Ariel Felner, and Nathan Sturtevant. Conflict-based search for optimal multi-agent path finding. In Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI), pages 563–569, 2012.

[4] Meir Goldenberg, Ariel Felner, Roni Stern, Guni Sharon, Nathan R. Sturtevant, Robert C. Holte, and Jonathan Schaeffer. Enhanced partial expansion A*. Journal of Artificial Intelligence Research (JAIR), 50: 141–187, 2014.

[5] Pavel Surynek, Ariel Felner, Roni Stern, and Eli Boyarski. Efficient SAT approach to multi-agent path finding under the sum of costs objective. In Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI), pages 810–818, 2016.

[6] Edward Lam, Pierre Le Bodic, Daniel Damir Harabor, and Peter J. Stuckey. Branch-and-cut-and-price for multi-agent pathfinding. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pages 1289–1296, 2019.

[7] Pavel Surynek. A novel approach to path planning for multiple robots in bi-connected graphs. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 3613–3619, 2009.

[8] Mokhtar M. Khorshid, Robert C. Holte, and Nathan R. Sturtevant. A polynomial-time algorithm for non-optimal multi-agent pathfinding. In Proceedings, of the 4th International Symposium on Combinatorial Search (SoCS), pages 76–83, 2011.

[9] Boris de Wilde, Adriaan ter Mors, and Cees Witteveen. Push and rotate: a complete multi-agent pathfinding algorithm. Journal of Artificial Intelligence Research (JAIR), 51: 443–492, 2014.

[10] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In Proceedings of the 7th Annual Symposium on Combinatorial Search (SoCS), pages 961–962, 2014.

[11] Liron Cohen, Tansel Uras, and Sven Koenig. Feasibility study: using highways for bounded-suboptimal multi-agent path finding. In Proceedings of the 8th International Symposium on Combinatorial Search (SoCS), pages 2–8, 2015.

[12] David Silver. Cooperative pathfinding. In Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE), pages 117–122, 2005.

[13] Ryan Luna, and Kostas E. Bekris. Efficient and complete centralized multi-robot path planning. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), pages 3268–3275, 2011.

[14] Hang Ma, Daniel Harabor, Peter J. Stuckey, Jiaoyang Li, and Sven Koenig. Searching with consistent prioritization for multi-agent path finding. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI), pages 7643–7650, 2019.

[15] Keisuke Okumura, Manao Machida, Xavier Défago, and Yasumasa Tamura. Priority inheritance with backtracking for iterative multi-agent path finding. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), pages 535–542, 2019.

[16] Shuai D Han, and Jingjin Yu. DDM: fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics. IEEE Robotics and Automation Letters (RA-L), 5(2): 1350-1357, 2020.