Loosely Synchronized Rule-based planning for Multi-Agent Path Finding with Asynchronous Actions

Shuai Zhou (SCUT), Shizhe Zhao (SJTU), Zhongqiang Ren (SJTU)
UM-SJTU Joint Institute, Shanghai Jiao Tong University,
SHIEN-MING WU School of Intelligent Engineering, South China University of Technology
Department of Automation, Shanghai Jiao Tong University,

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

I. Abstract

Multi-Agent Path Finding (MAPF) seeks collision-free paths for multiple agents from their respective starting locations to their respective goal locations while minimizing path costs. Although many MAPF algorithms were developed and can handle up to thousands of agents, they usually rely on the assumption that each action of the agent takes a time unit, and the actions of all agents are synchronized in a sense that the actions of agents start at the same discrete time step, which may limit their use in practice. Only a few algorithms were developed to address asynchronous actions, and they all lie on one end of the spectrum, focusing on finding optimal solutions with limited scalability. This paper develops new planners that lie on the other end of the spectrum, trading off solution quality for scalability, by finding an unbounded sub-optimal solution for many agents. Our method leverages both search methods (LSS) in handling asynchronous actions and rule-based planning methods (PIBT) for MAPF. We analyze the properties of our method and test it against several baselines with up to 1000 agents in various maps. Given a runtime limit, our method can handle an order of magnitude more agents than the baselines with about 25% longer makespan.

II. Motivation & Problem Formulation

https://d1.awsstatic.com/psc-digital/2023/gc-400/dnb-customer-lookbook-ebook/DNB-Customer-Lookbook-eBook.pdf

https://www.robotics247.com/article/bmw_logistics_using_autonomous_transport_robots

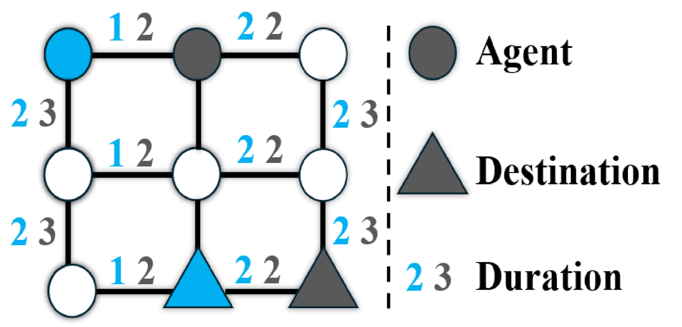


Fig. Multi-Agent Path Finding with Asynchronous actions

Multi-Agent Path Finding (MAPF)
- Search joint collision-free path for a group of agents
- Minimize the sum of costs (Soc) and makespan
- **Agents move synchronously with unit time durations**
- **Well studied**

Multi-Agent Path Finding with Asynchronous Actions (MAPF-AA)
- Search joint collision-free paths for a group of agents
- Minimize the sum of costs (Soc) and makespan
- **Agents move asynchronously with different durations.**
- **Relatively understudied.**

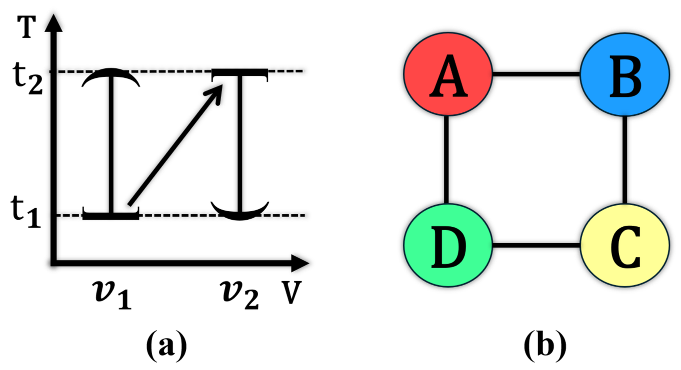
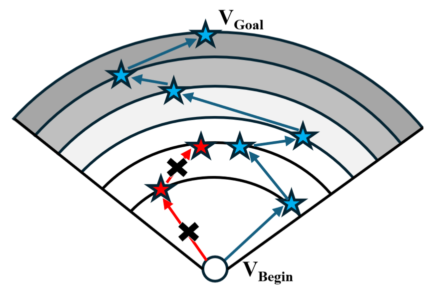


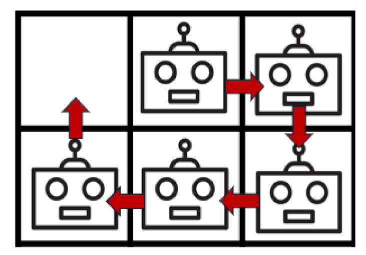
Fig. Duration conflict model

Collision model
- Used by: LSS [1], Causal-PIBT [2]
- Occupation:
  - Edge : $[t_1,t_2]$
  - Begin vertex: $[t_1,t_2)$
  - End vertex: $(t_1,t_2]$
- Collison:
  - vertex collision
  - edge collision

[1] Ren, Z.; Rathinam, S.; and Choset, H. 2021. Loosely synchronized search for multi-agent path finding with asynchronous actions. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 9714–9719. IEEE.
[2] Okumura, K.; Tamura, Y.; and Defago, X. 2021. Time- Independent Planning for Multiple Moving Agents. Proceedings of the AAAI Conference on Artificial Intelligence,35(13): 11299–11307.

III. Related Work





Optimal/Bounded suboptimal	**Unbounded suboptimal**
Low scalability	**High scalability**
Loosely synchronized search (LSS)[1]	Priority Inheritance with Backtracking (PIBT) [4]
Continuous Conflict Based Search (CBS)[2]	- Limited to MAPF
Increasing Cost Tree Search for Non-Unit Cost Domains (ε-ICTS)[3]	
- Existing approach for MAPF-AA	

Goal of this work

PIBT + LSS ⇨ LSRP

Bring the scalability of PIBT to Multi-Agent Path Finding with Asynchonous Actions, developing method for the other spectrum.

[1] Ren, Z.; Rathinam, S.; and Choset, H. 2021. Loosely synchronized search for multi-agent path finding with asynchronous actions. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 9714–9719. IEEE.
[2] Andreychuk, A.; Yakovlev, K.; Surynek, P.; Atzmon, D.; and Stern, R. 2022. Multi-agent pathfinding with continuous time. Artificial Intelligence, 305: 103662
[3] Walker, T. T.; Sturtevant, N. R.; and Felner, A. 2018. Extended Increasing Cost Tree Search for Non-Unit Cost Domains. In IJCAI, 534–540.
[4] Okumura, K.; Machida, M.; Defago, X.; and Tamura, Y. ´2022. Priority Inheritance with Backtracking for Iterative Multi-agent Path Finding. Artificial Intelligence, 103752.

IV. Method & Analysis

Setting
- Priority ε
  - Each agent is assigned a unique priority
  - Determine the order of planning
  - Reset priority when agent reach goal
- Cache Φ
  - Store pre-planned move action
  - Initialized to empty

Asynchronous Push (ASY-PUSH)
- Trigger condition
  - agents compete for vertex/edge
- Mechanism
  - Priority inheritance and back inform
  - Wait and move using help of Φ

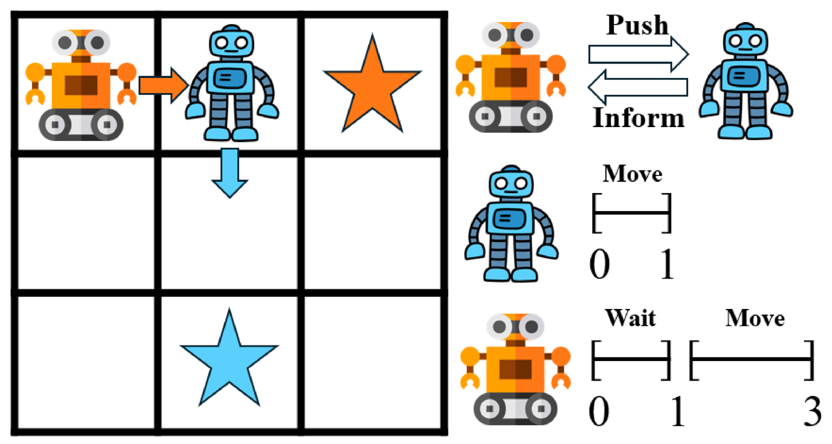


Fig. Toy example of ASY-PUSH

Analysis
- Reachability
  - Each agent reach goal at least once under C-graph
  - see more details in our paper*

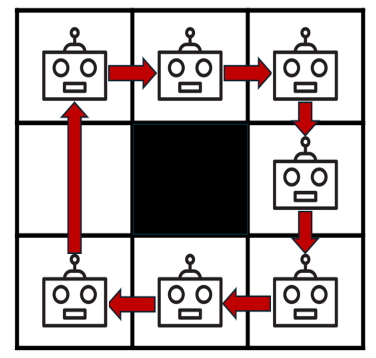


Fig. C-graph

Failure case
- Condition
  - Two agents to swap in a non-C-graph
  - Caused by greedy strategy and Priority reset

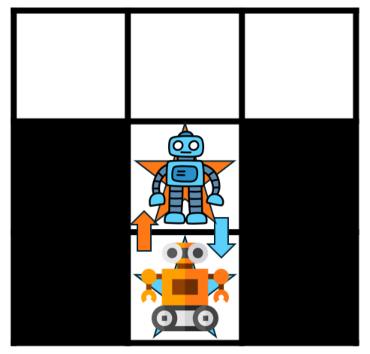


Fig. Failure case

Asynchronous Swap (ASY-SWAP)
- Trigger condition
  - failure case of LSRP
- Mechanism
  - Detect necessity of swap
  - Detect executability of swap
  - Non-greedy search strategy and pull
- Defect
  - Unable to solve all the failure cases

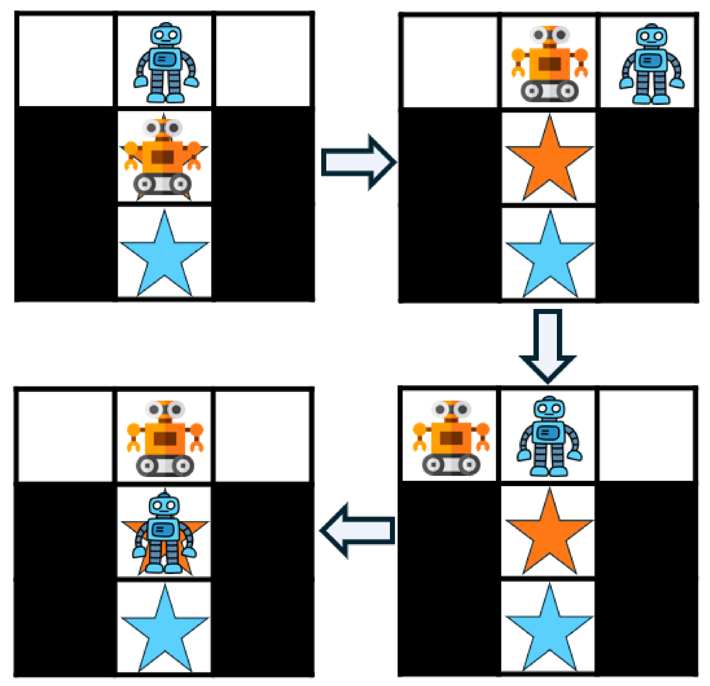
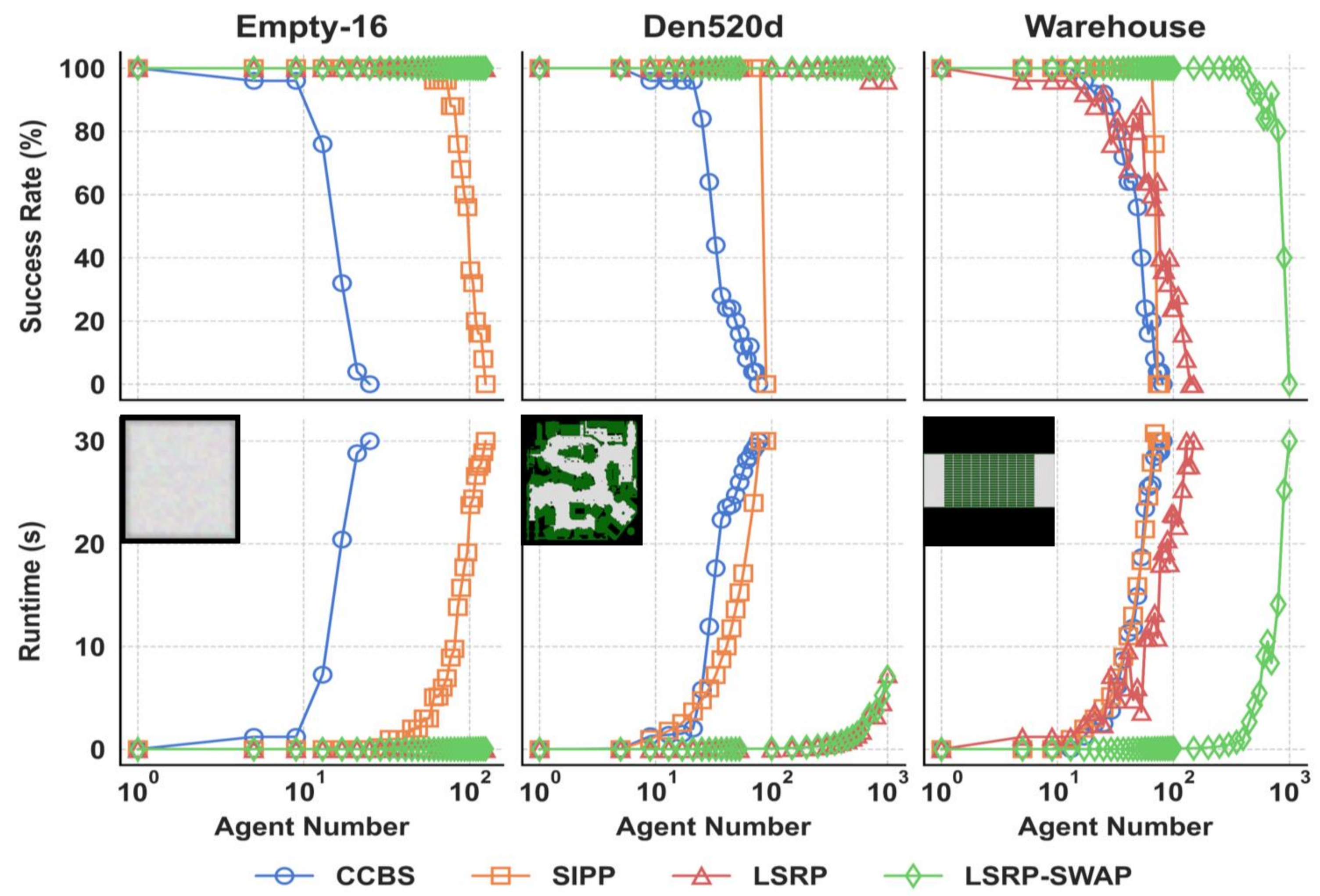
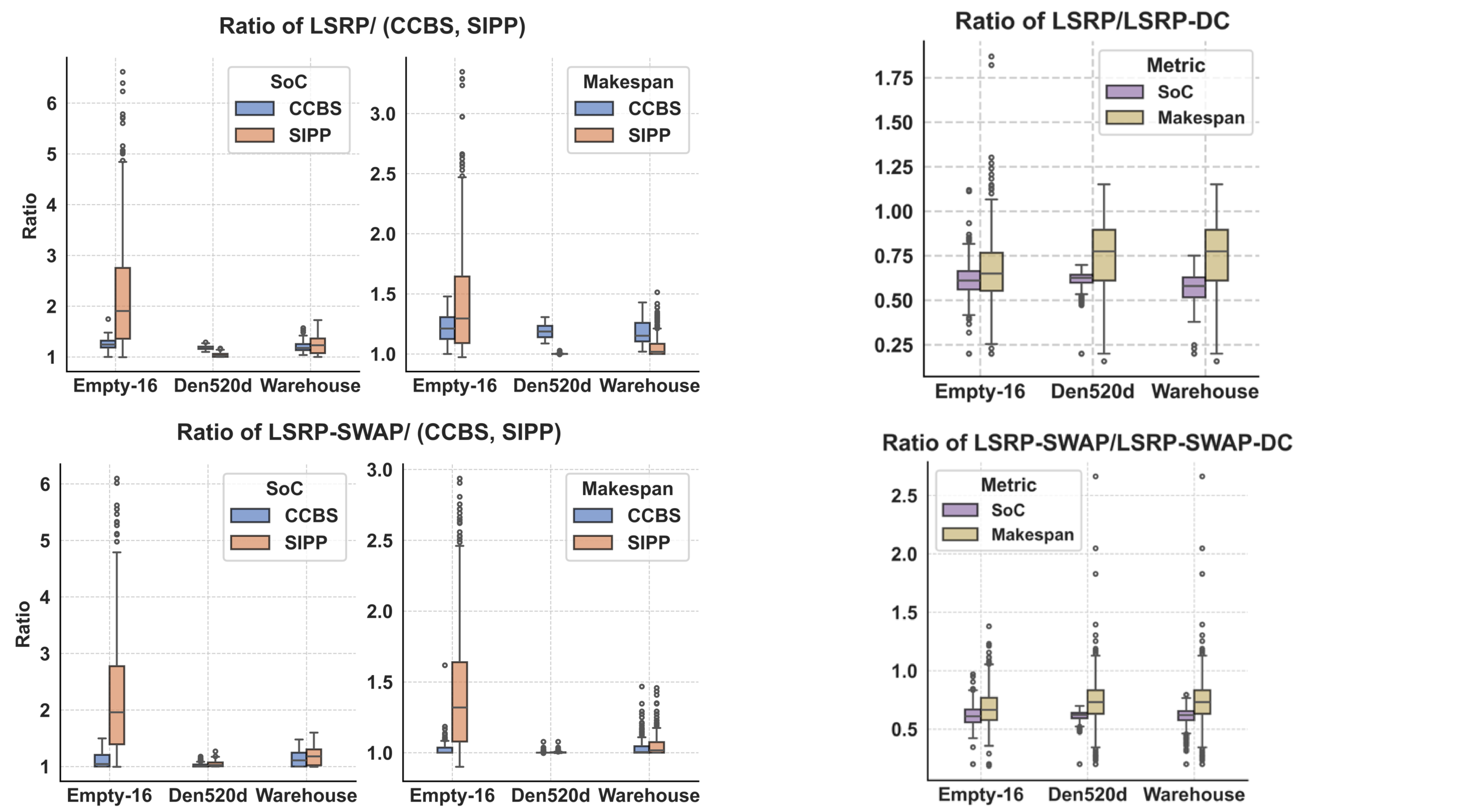


Fig. Toy example of ASY-SWAP

V. Result





Baselines
- CCBS [1]
  - Optimal solution
  - Limited scalability (20 – 40)
- SIPP [2]
  - Prioritize Planning
  - Unbounded suboptimal solution
  - Limited scalability (100 – 200)

Scalability
- **Handles up to 1000 agents**
- 10x improvement over existing methods
- Sparse maps (Empty-16, Den520d) results show graph reachability
- Cluttered maps (Warehouse) results highlight advantage of swap operation

Solution quality
- **4x in SoC, 1.25x in Makespan**
- Trading off solution quality for scalability
- LSRP and LSRP-SWAP offer comparable quality in SoC and Makespan
- SIPP has a higher ratio than CCBS on Empty-16 due to solving more instances

MAPF-DC
- Ignores asynchronous actions; all agents share common planning timestamps, planned step-by-step

Impact of Asynchronous Actions
- **Solutions 30%-75% cheaper**
- Importance of considering asynchronous actions, especially when agents have different duration.

Paper **Code** **Contact** **Blog**

[1] Andreychuk, A.; Yakovlev, K.; Surynek, P.; Atzmon, D.; and Stern, R. 2022. Multi-agent pathfinding with continuous time. Artificial Intelligence, 305: 103662
[2] Phillips, M.; and Likhachev, M. 2011. Sipp: Safe interval path planning for dynamic environments. In 2011 IEEE International Conference on Robotics and Automation, 5628–5635. IEEE.