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 Kumar¹ and Sven Koenig¹

¹ University of Southern California
² Amazon Robotics

AAMAS-20
Fulfillment 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=PL1JBGaGtAhqTLBCFWT85p6KghwkJHj398&index=2&t=0s
[top right] Inside the amazon warehouse where humans and machines become one. https://www.wired.com/story/amazon-warehouse-robots/

Sorting center
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.

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.

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).

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.

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

Multi-Label A* [Grenouilleau et al 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:

<table>
<thead>
<tr>
<th>Agents</th>
<th>Holding endpoints</th>
<th>Dummy paths</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throughput</td>
<td>Runtime (s)</td>
<td>Throughput</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>2.17</td>
<td>0.01</td>
<td>2.19</td>
</tr>
<tr>
<td>100</td>
<td>3.33</td>
<td>0.02</td>
<td>3.41</td>
</tr>
<tr>
<td>140</td>
<td>4.35</td>
<td>0.04</td>
<td>4.50</td>
</tr>
</tbody>
</table>

- All methods use PBS as the (Windowed) MAPF solver.
- Our method: resolving collisions 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 $w$:

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

<table>
<thead>
<tr>
<th>CA</th>
<th>Agents</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>500</th>
<th>600</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w = 5$</td>
<td>6.17</td>
<td>9.12</td>
<td>-</td>
<td>-</td>
<td>3.17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w = \infty$</td>
<td>6.20</td>
<td>9.16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w = 5$</td>
<td>0.21</td>
<td>1.07</td>
<td>-</td>
<td>-</td>
<td>0.14</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>$w = \infty$</td>
<td>0.84</td>
<td>2.58</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CBS</th>
<th>Agents</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>500</th>
<th>600</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w = 5$</td>
<td>12.03</td>
<td>14.79</td>
<td>17.28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w = \infty$</td>
<td>12.28</td>
<td>15.20</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w = 5$</td>
<td>1.27</td>
<td>2.37</td>
<td>4.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w = \infty$</td>
<td>11.48</td>
<td>23.47</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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


