Pick-up and delivery using MAPF

Seminar on AI 2018

Team #1

David Nohejl

Chaman Shafiq

Věra Škopková

Problem Statement

- Given:
  - Map of the environment
  - Pair of Locations (Pick-up and delivery/goal locs)
  - Starting point
- Multiple robots – every robot will complete his task:
  - 1) pick up an item at a pick-up location
  - 2) deliver that item to goal location

Motivation

- Ware-houses
- Dispatching of fireman, ambulance, police

- Expected results
  - Safeness: Finding safe, quick path, avoid traffic jams
  - Ecology: Save energy, environment
  - Usage of as little robots/cars/etc. as possible

Mission

- Avoiding obstacles while traveling and respecting the other agents path
- Follow physical laws e.g. two robots cannot be at same place etc.
- Find shortest path for both locations from current location
- Shortest path can be (in real life)
  1) Fewest turns
  2) Safe path
  3) less energy consumption
  4) Less traffic route (avoiding heavy traffic/block junctions)

Vision

- Minimizing the limitations (making the robot efficient enough to use his capabilities fully like human use their senses) and work dynamically.
- We will see the limitations at the end of this presentation.

Plans for achieving mission

- Offline planning → execution on robots
- Abstraction – discrete steps
- Plans should be collision free
- Follow physical laws:

  E.g.
  - Two agents cannot be at the same place at the same time
  - Only one agent can go through one way at one time step

Abstraction vs Reality

- Let´s observe how plans will work
  - Robots move continuously and everyone might be possible will have slightly different speed from others. (due to low battery, friction among the shafts, wheels and corresponding surface)
  - It is impossible to start robots at exactly same time (can generate different path scenarios)
  - There can be some unexpected problem – another moving robot in the way, an obstacle, collision with other robots and obstacles, get loss somewhere or finish at dead end etc.
- Robots should be able to react to such situations
- We are currently focused on planning due to some limitations. We will use only virtual “items” that will be transported.

Possible improvements

- Use proximity sensor to detect if the way is free
  - Distinguish another robot from real obstacle
    - Waiting in a queue for pickup if queue is long then take another detour if there is another pickup location available.
- Contingent plans (with increasing complexity)
  - Wait until the way is free
  - Plans with alternatives (re-planning in the robot on board) which is also not possible by using ozobot.
    - Usage of branches and loops in “fixed“ plan
  - Communication with other robots – in semaphore simulations (which is not possible at the moment but in future we can)

Limitations

- Is communication possible between robots in Ozoblockly currently?
- Hardware limitations
  - How much and fast are we able to do re-planning on-board (Time complexity)?
  - Is it possible to plan the whole path in the robot (Space Complexity)?

Available algorithms for MAPF

- We can try different algorithms for MAPF and compare them or choose the best one for our problem
  - Highways
  - A* variants
  - MA-CBS
  - FAR

References

- http://idm-lab.org/bib/abstracts/papers/socs15b.pdf
- https://www.aaai.org/ocs/index.php/SOCS/SOCS12/paper/viewFile/5402/5182
- https://vvvvw.aaai.org/Papers/ICAPS/2008/ICAPS08-047.pdf
- https://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/viewFile/5233/5640
- http://idm-lab.org/bib/abstracts/papers/aamas17.pdf