MAPF – Final Report

Seminar on Artificial Inteligence II

2018

Team #1

David Nohejl

Chaman Shafiq

Věra Škopková

Little change of the task

- Pick-up and Delivery -> MAPF, solving conflicts
- Given:
  - Map of the environment
  - Finite number of robots
  - Start and goal locations for each robot
- Solving:
  - Conflict-free paths for all agents
  - Problems caused by physical properties of robots

Possible improvements of basic plans

- Use proximity sensor to detect if the way is free
  - Distinguish another robot from real obstacle
    - Waiting in a queue X reacting on other agent
- Contingent plans (with increasing complexity)
  - Wait until the way is free
  - Plans with alternatives
    - Somehow react to actual situation
  - Communicate with other robots

Observation

WHAT WE DISCOVERED

Properties of Ozobots

- The best maps are the ones that are drawn by a highlighter
- Running of all ozobots synchronously is not simple
- Ozobots see real obstacles by proximity sensors quite well
  - But they have problems to see each other
- Communication works good running separately but on junctions Ozobots never start to communicate
- Using Ozoblockly is sometimes exhausting
  - You can do silly mistakes you would never do in a common programming language
  - Code for communication can be loaded into ozobot only by flashing

Execution of plans

- Making of turns takes nontrivial amount of time
  - Wait a while if you are not turning
  - Find a k-robust plan
- It is hard to choose ideal distance between nodes not to crash and to detect each other at once
  - At the end we rather didn´t use the proximity sensors at all for more complicated plans
- Simple attempts for collision detection don´t save the situation at most cases
  - Step back – can crash all plans
  - Communication – did not work

Demonstrations

- Big example with 4 robots (video)
- Train with 4 robots (video)
- Delay of turns (video)
- K-robust (video)
- Communication (video)

- StepBack (demo)