MULTI-AGENT PATH FINDING

Simulation of manufacturing processes

Chembrolu Surya
Felipe Vianna
Yuu Sakaguchi

Problem

- Find paths for each robot according to sequence of operations they should go to.

- Avoid collision between robots

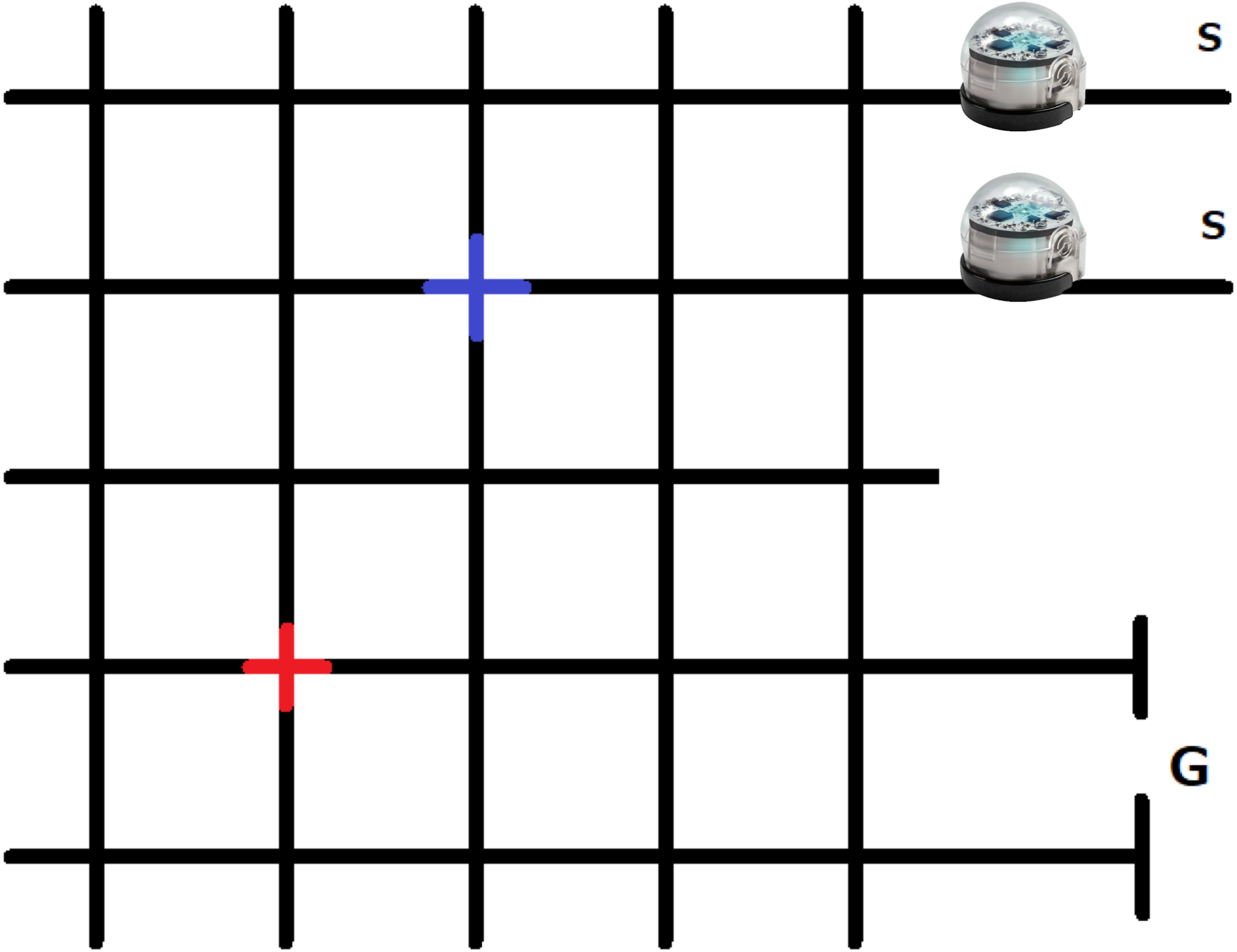
Given parameters

- Grid map – As adjacent Matrix for paths calculations

- Position sequences for each robot: Start, op1, op2, end

- nodesWeights=[1 1 1 1 1 1 1 1 1 1 1 10 1 1 1 1 1 1 5 1 1 1 1 1 1]
- ozo1seq=[1 12 19 4]
- ozo2seq=[2 19 12 5]

Map



Implementation

- Initially it was considered to use A* to find best path, as some nodes has different weights. But to avoid conflicts, not always the best path can be used

- Implemented BFS and DFS. Find options of paths that are ranked from shortest to longest.

- Select path for each robot and check for conflicts. If there is a conflict, pick new path for the section in which conflict occurs for one of the robots. Try new paths while conflict exists.

Implementation

- BFS, DFS and all functions related to selection of paths implemented in matlab. Output the sequence of steps for each robot. Operations result in repeated steps in the path representing their duration

- Python script to generate XML file that would be loaded in OzoBlockly.

Results

- Tested for 2 robots and 2 operations.
- Non-conflict paths are found in seconds.
- Optimal solution not assured.

Results

- Video

Next Steps

- Implement a planner for the operations sequencing. For now it was just decided manually.

- Increase number of operations and robots.

- Implement reactive decisions using sensors. Initial idea was have path finding algorithm embedded in ozobot, but current interface doesn’t allow that.