Path Planning for Multiple Warehouse Autonomous Ground Vehicles

Group - 12:

Abhiram Dapke

Akshay Bapat

Sanket Acharya

Introduction

Use of online shopping has been growing exponentially

Warehouse robots that pick up packages and transport them to drop-off areas are crucial

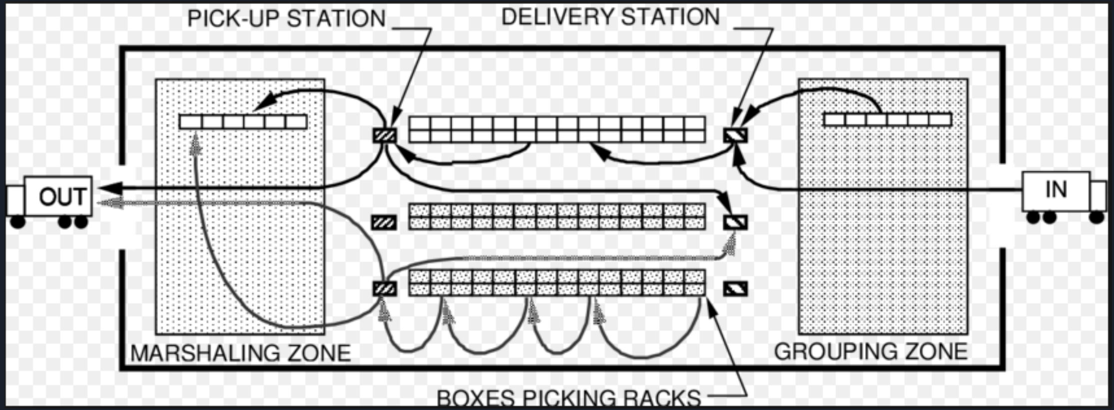
For this project: downsized version of a warehouse having multiple Autonomous Ground Vehicle (AGVs) considered

Problem Statement

- 8 line-following holonomic AGVs (4-connected grid) with diameter 60 cm working in a workspace (similar to a warehouse) of dimensions 28.2m x 17m
- Plan a path according to initial requirements of pick-up and drop-off locations for each robot

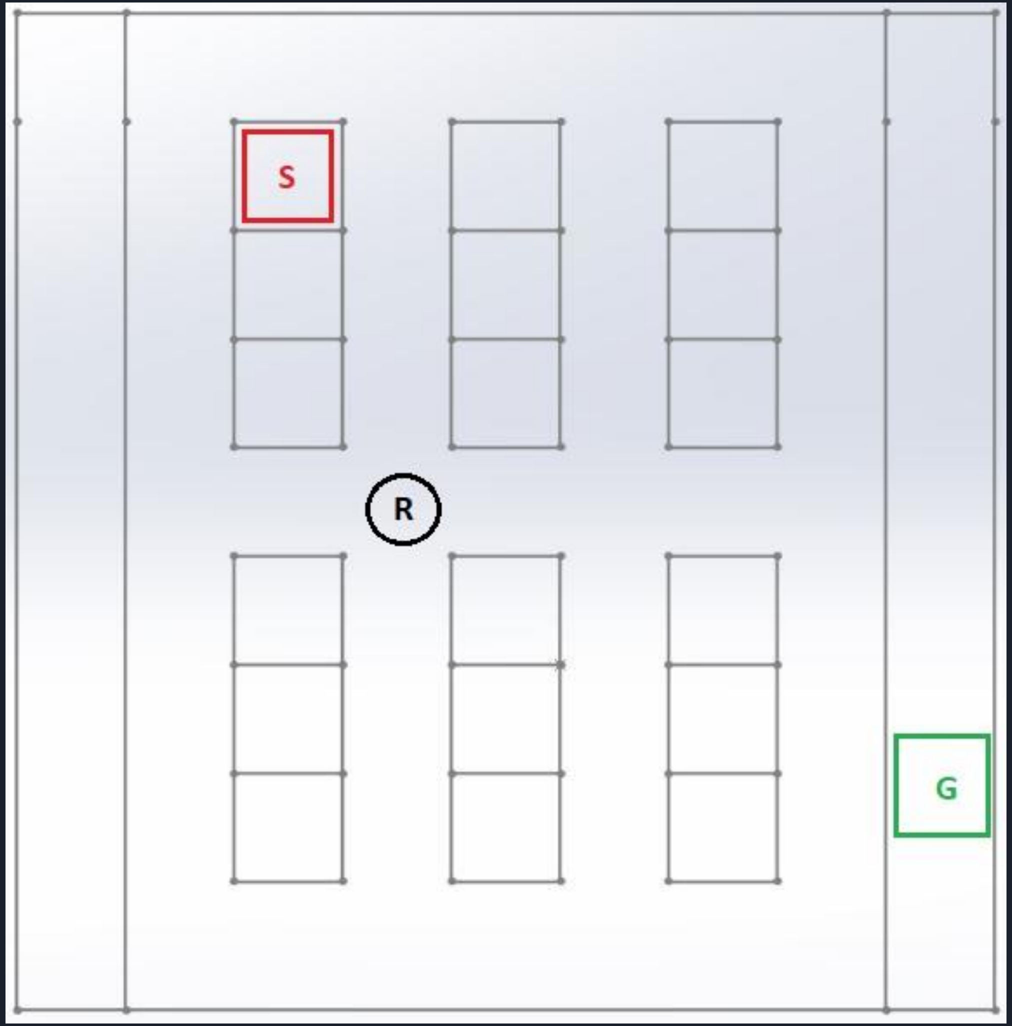
Two major areas that we need for warehouse space planning

Receiving and Shipping Activities



Storage Activities

WorkSpace



Assumptions

- 8 robots are used.
- The robots can move only along fixed lines on the workspace floor.
- AGVs have identical maximum speeds.
- No human interference is considered in the workspace.
- Sets of 3 shelves arranged in an 8x8 grid in the workspace: each shelf having dimensions 80cm x 80cm

Classical Prioritized Planning

- A centralized planning approach
- Assigns priorities to all the robots beforehand
- In decreasing order of priorities, for the i^{th} robot, plan the shortest path to goal considering all static obstacles and all dynamic obstacles (robots) from 1, 2, …, i-1
- Takes care of all conflicts between robots beforehand

Notation

Δ - Obstacle space

π_i - Path planned for robot i

$R_i\Delta(\pi_i)$ - A function that maps the trajectories of robot i to regions of space-time (that is, coordinates of the nature (x,y,t)) when its center point follows a trajectory π_i

Pseudo-code for Prioritized Planning

$\Delta \leftarrow \phi$

for i $\leftarrow 1,2,\ldots,n$ *do:*

$\pi_i \leftarrow shortest\_path(S_i, G_i, \Delta)$

if $\pi_i = \phi$ *then:*

report failure and terminate

$\Delta \leftarrow \Delta \cup R_i \Delta(\pi_i)$

Function $shortest\_path(S_i, G_i, \Delta)$*:*

return shortest path for robot i from start node S_i *to goal node* G_i *while avoiding the obstacle space* Δ

Shortest Path Algorithm - A* algorithm

- A*: Set the cost to come of all unvisited nodes as infinity

Expand a visited node n and get its successors (4-connected space) n’

Add each successor in a priority queue with priority = (cost to come) + (cost to goal)

cost to come = cost to reach n’ from start node

cost to go = cost (approximation function) to reach goal node from n’

Always searches first in the direction of least cost to goal

D* Lite algorithm

- D* Lite is based on Lifelong Planning A*
- D* Lite is an incremental heuristic search algorithm by Sven Koenig and Maxim Likhachev.
- Calculates heuristic cost from goal instead of start node

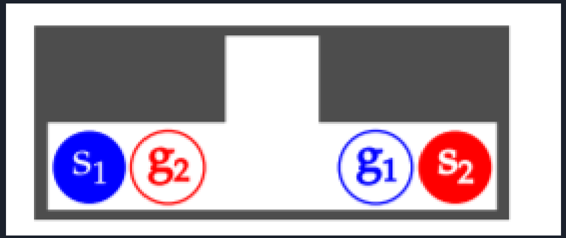
D* Lite Pseudo Code:

procedure Main()

$s_{last} = s_{start}$;
Initialize();
ComputeShortestPath();
while $(s_{start} \neq s_{goal})$
    /* if $(rhs(s_{start}) = \infty)$ then there is no known path */
    $s_{start} = \arg\min_{s' \in Succ(s_{start})}(c(s_{start}, s') + g(s'))$;
    Move to s_{start};
    Scan graph for changed edge costs;
    if any edge costs changed
        $k_m = k_m + h(s_{last}, s_{start})$;
        $s_{last} = s_{start}$;
        for all directed edges (u, v) with changed edge costs
            $c_{old} = c(u, v)$;
            Update the edge cost $c(u, v)$;
            if $(c_{old} > c(u, v))$
                if $(u \neq s_{goal})$ $rhs(u) = \min(rhs(u), c(u, v) + g(v))$;
            else if $(rhs(u) = c_{old} + g(v))$
                if $(u \neq s_{goal})$ $rhs(u) = \min_{s' \in Succ(u)}(c(u, s') + g(s'))$;
            UpdateVertex(u);
        ComputeShortestPath();

Source: S. Koenig and M. Likhachev. D* Lite. In Proceedings of the AAAI Conference of Artificial Intelligence (AAAI), pages 476-483, 2002.

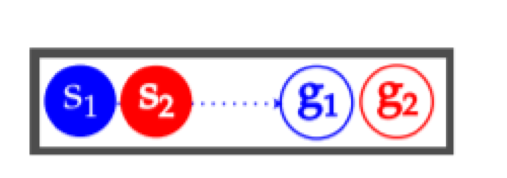
Shortcomings of Prioritized Planning algorithm



Case I

- The picture shows two robots desiring to move from s1 to g1 and s2 to g2 respectively in a corridor that is only slightly wider than a body of a single robot. The scenario assumes that both robots have identical speeds.
- In this case, trajectory of the robot that plans first will be in conflict with all satisfying trajectories of the robot that plans second, irrespective of the sequence.

Shortcomings of Prioritized Planning algorithm



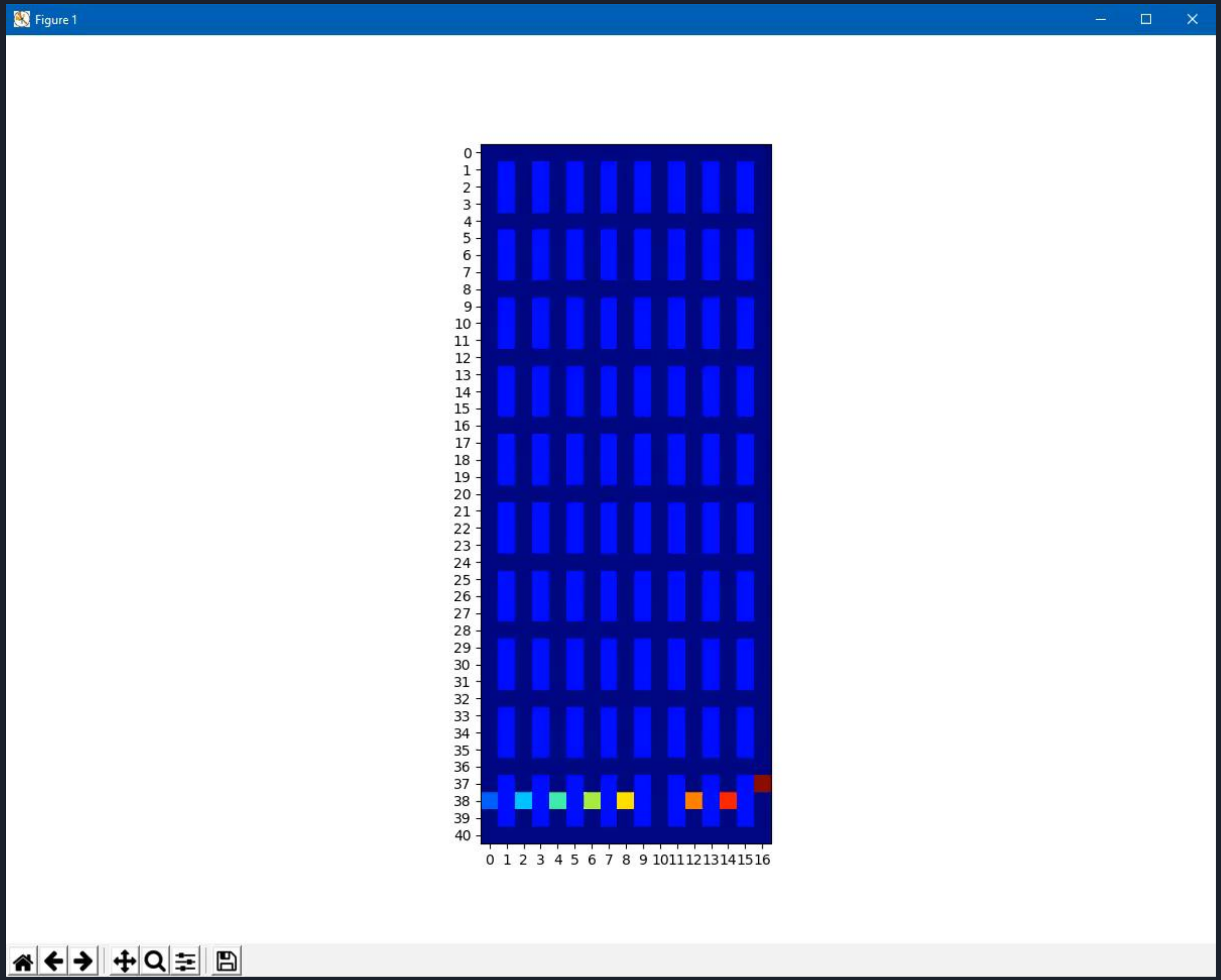
Case II

- In this case, robot 1 travels from s1 to g1, robot 2 travels from s2 to g2. The scenario assumes that robot 1 can travel twice as fast as robot 2.
- Robot 1 plans first and adopts a straight line trajectory from s1 to g1 at the maximum speed, because it ignores the task of robot 2. Robot 2 plans second, but all satisfying trajectories for robot 2 are in conflict with robot 1.

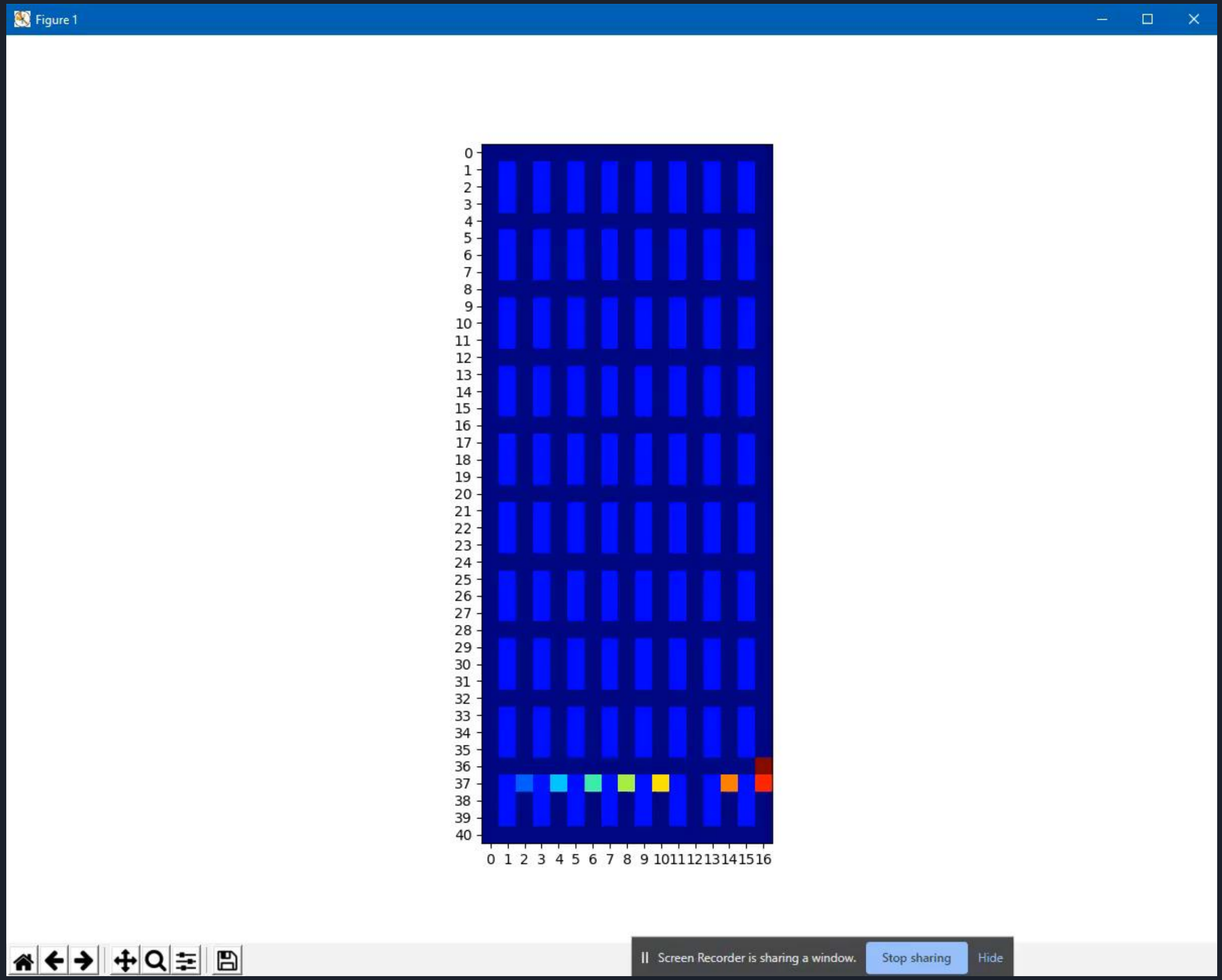
Results

- The video shows that the prioritized planning algorithm has been implemented successfully: the robots do not collide with each other
- Three test cases considered:
  - all target locations at the left
  - all target locations at the right
  - split left-right (demo case)

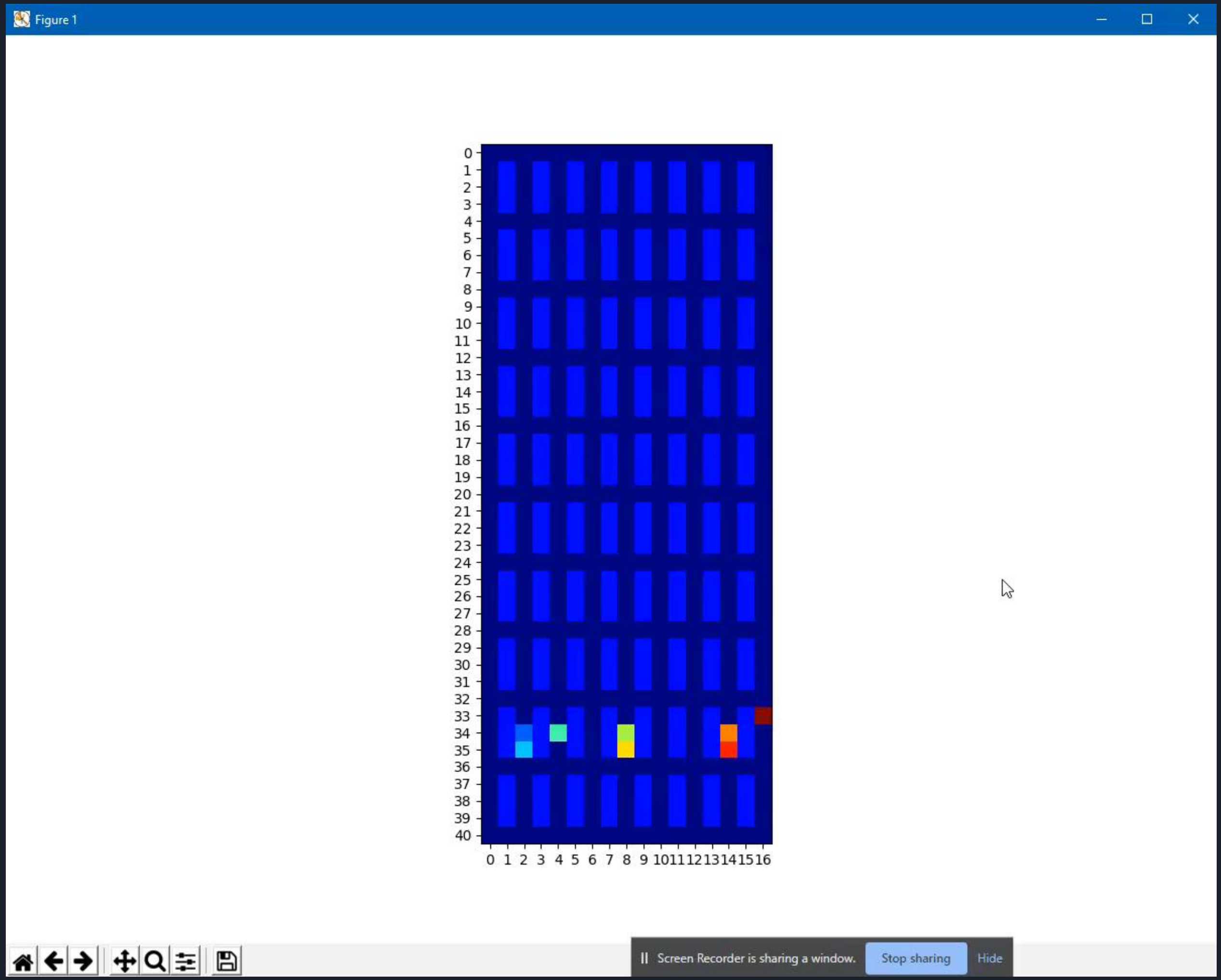
Results – Case 1 (All to the left)



Results – Case 2 (All to the right)



Results – 3 (Split left and right)



Future Work

- Human interference can be considered.
- Extra feature based on the state of a robot can be added.

  If a robot is in state 0 i.e. it is not carrying any load, then it does not need to go underneath the shelf and pick it up and that is why, there would be no stationary obstacles for that particular robot, it would only have the dynamic obstacles which are the other robots in operation. This approach can help to reduce traffic on main lines as there would be only loaded robots on the main lines and robots which are not loaded can roam anywhere in the workspace, so more robots can be used in the warehouse. This will also improve the overall efficiency of the system.

References:

- J van den Berg, J Snoeyink, M Lin, D Manocha, “Centralized Path Planning for Multiple Robots: Optimal Decoupling into Sequential Plans”, Robotics: Science and Systems, 2009
- P Svetska, MH Overmars, “Coordinated path planning for multiple robots”, Robotics and Autonomous Systems, Vol. 23, 125-152, 1998
- M Saha, P Isto, “Multi-Robot Motion Planning by Incremental Coordination”
- R D’Andrea, “A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead”, IEEE Transactions on Automation Science and Engineering, Vol. 9, No. 4, October 2012
- W Honig, S Kiesel, A Tinka, JW Durham, N Ayanian, “Persistent and Robust Execution of MAPF Schedules in Warehouses”, IEEE Robotics and Automation Letters, Vol. 4, No. 2, 2019
- RKA Sakir, A Rusdinar, S Yuwono, AS Wibowo, Silvirianti, NT Jayanti, “Movement Control Algorithm of Weighted Automated Guided Vehicle Using Fuzzy Inference System”, International Conference on Control and Robotics Engineering, 2017
- D Milner, “Swarm Robotics Algorithms: A Survey”, May 2007
- Michal Cáp, Peter Novák, Alexander Kleiner, Martin Selecký, “Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots”
- Jun-Han-Oh, Joo-Ho Park and Jong Tae-Lim, “Centralized Decoupled Path Planning Algorithm for Multiple Robots Using the Temporary Goal Configurations
- Fatih Okumus, Adnan Fatih Kocamaz, “Comparing Path Planning Algorithms for Multiple Mobile Robots

Thank You!