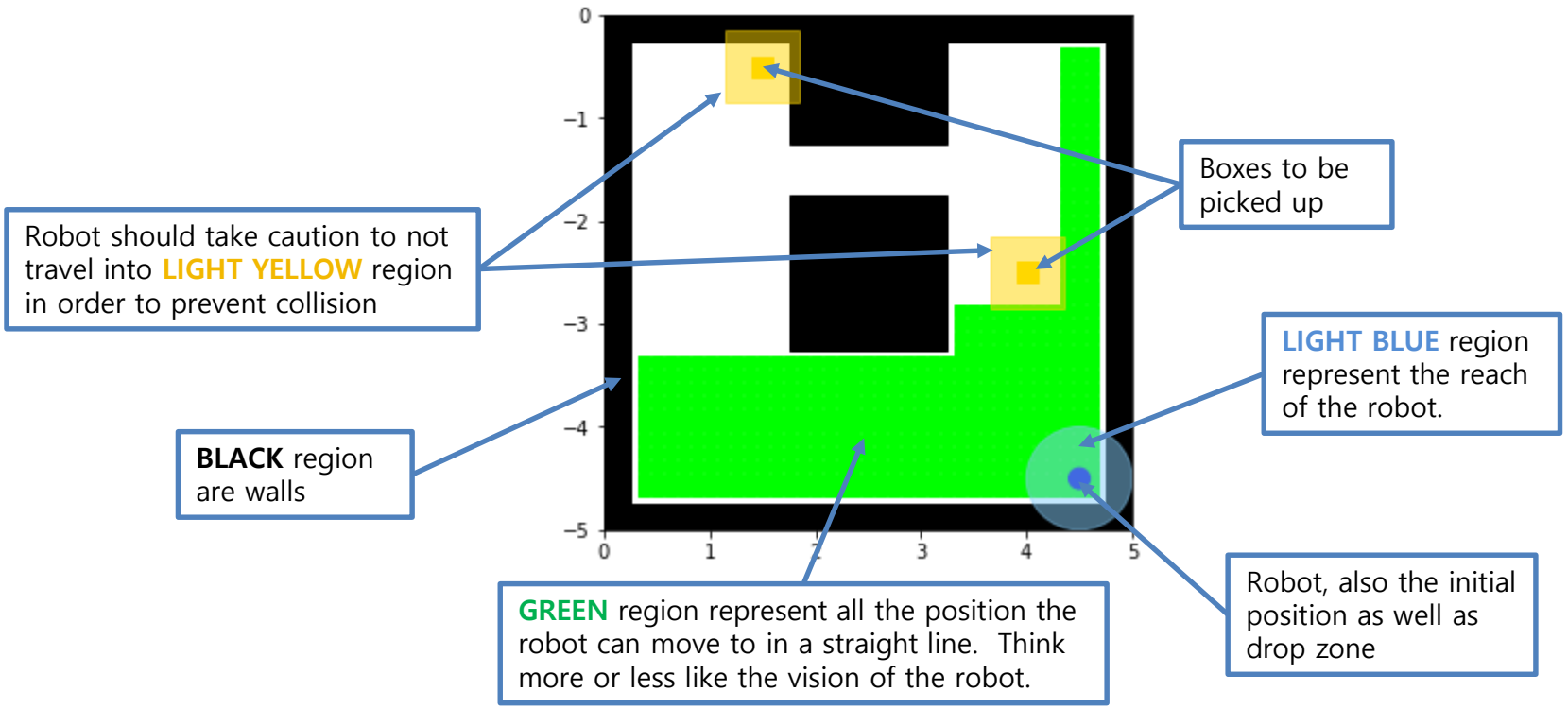
Robot Warehouse Navigation

Georgia Institute of Technology
CS 8803-001 Artificial Intelligence for Robotics
Summer 18
James Chan

Overview

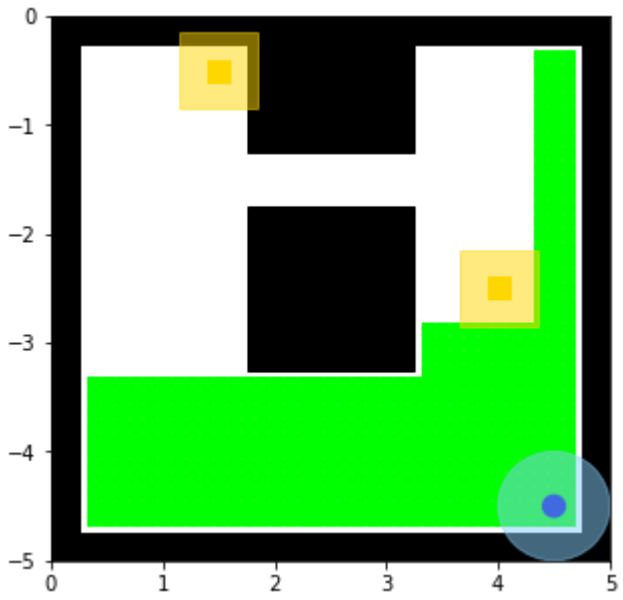
Problem Statement:

Devise a program to navigate a robot within a simulated warehouse. The goal of the robot is to collect all the boxes and bring them back to the drop zone. Any collision with walls or boxes are considered failure. The quicker (shorter path) the robot can collect all the boxes, the higher score the student receives. The program must work in various warehouses and box arrangements. Exploration is not necessary – the robot already have knowledge of the locations of the boxes ahead of time. The robot can only carry one box at a time.



Solution

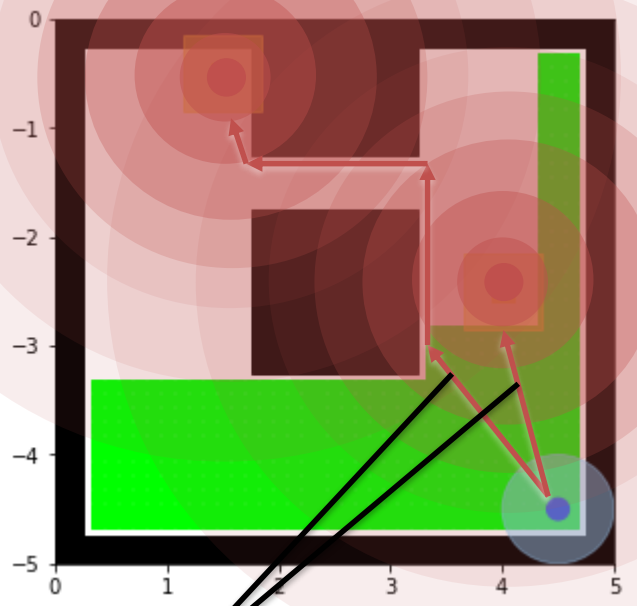
To calculate the optimal path, the map of the warehouse is discretized into grids and A* is utilized to search for the path with the shortest cost. The use of numpy and carefully selected heuristic function (continuous linear distance from the box) contributes significantly to the efficient performance of the program. Visualization generated with matplotlib.



Calculate path

Heuristic function

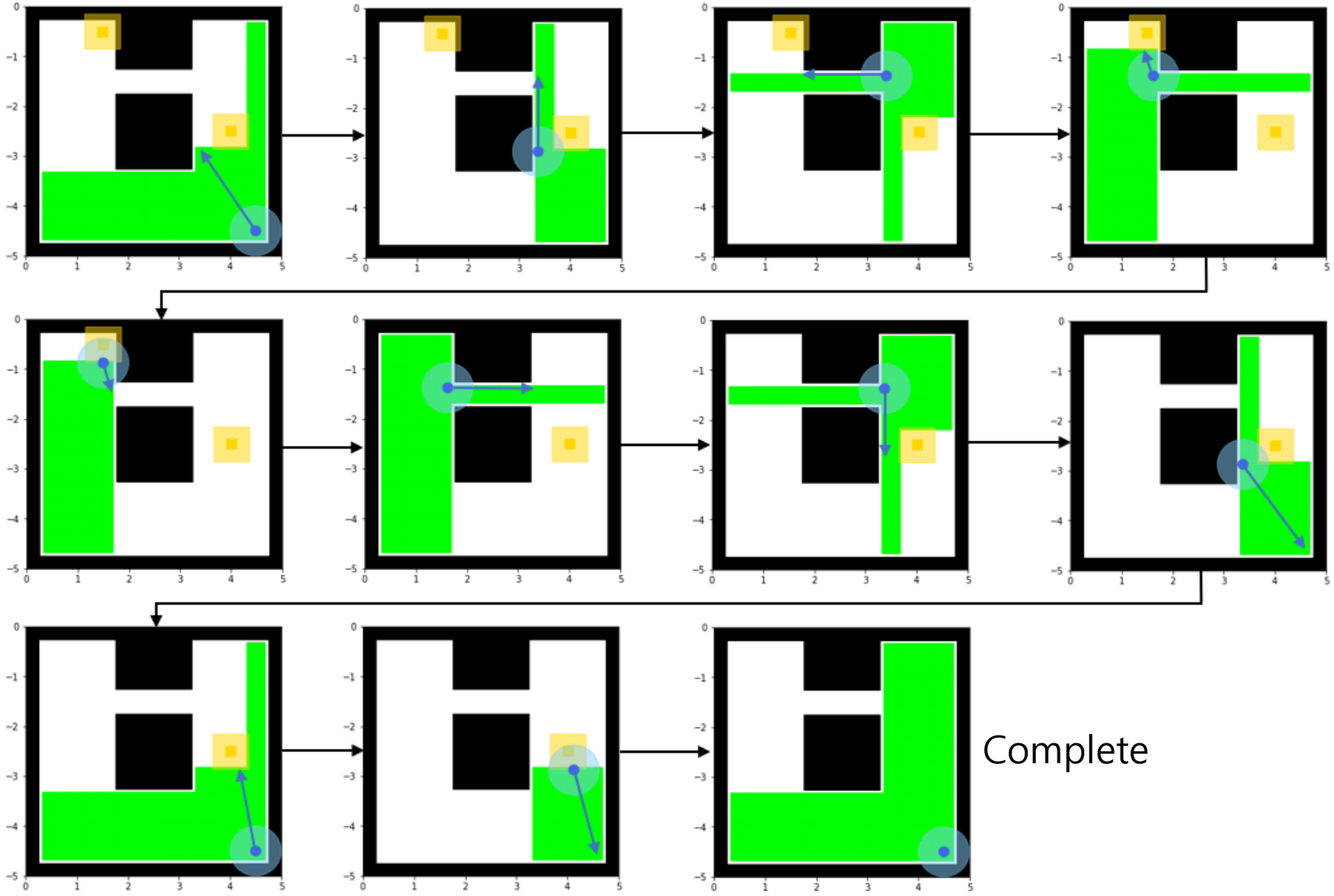
Heuristic function



Optimal/near-optimal path

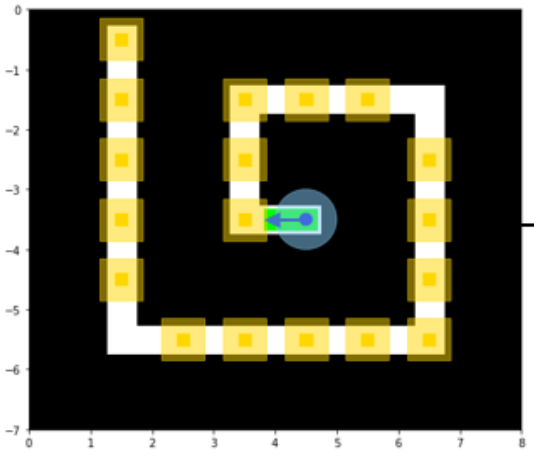
Trial Run – Warehouse #1

Start

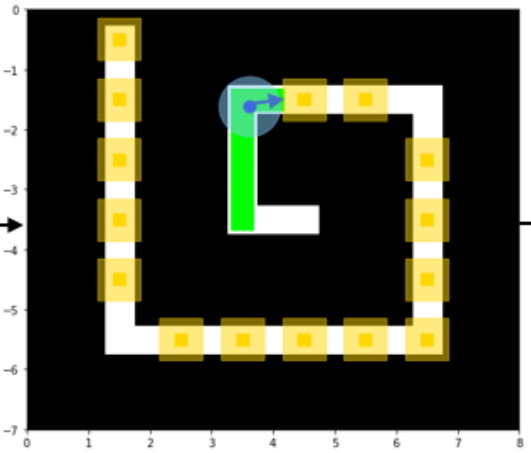


Complete

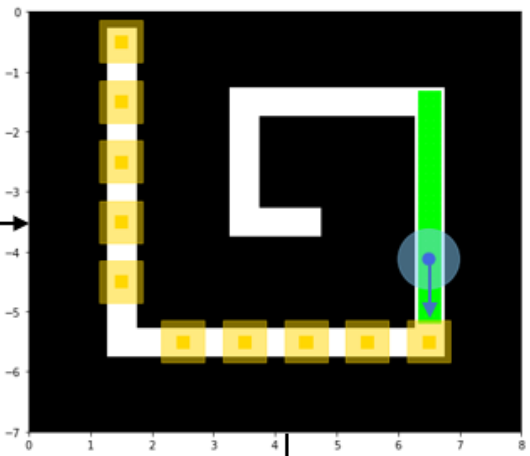
Trial Run – Warehouse #2



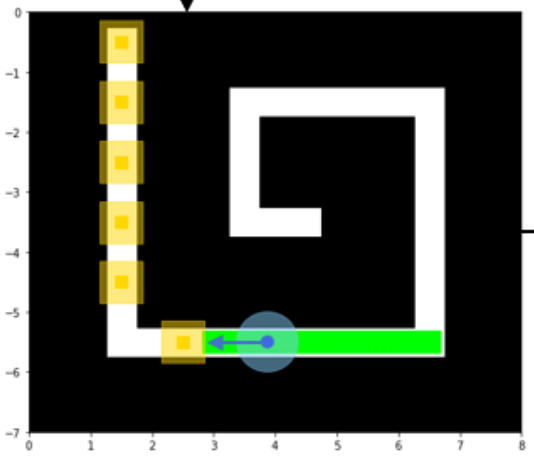
Start



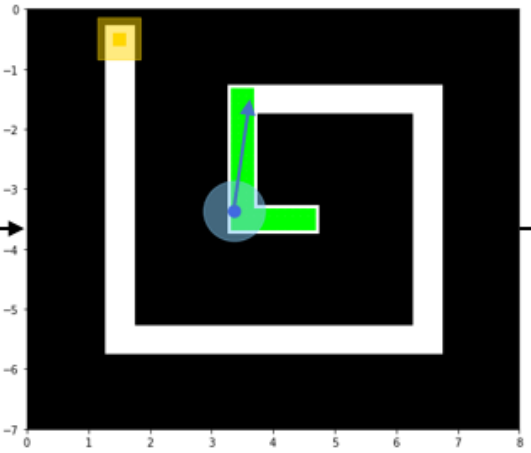
17th Step



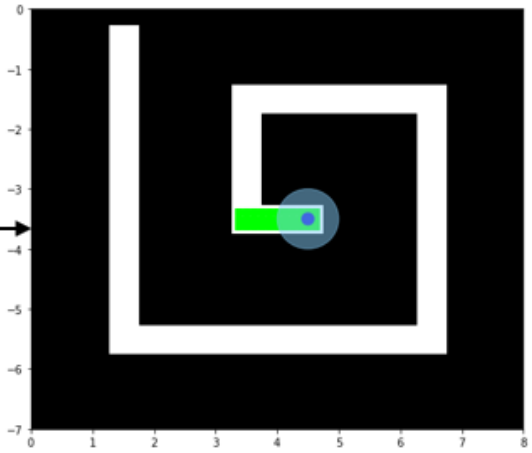
61th Step



124th Step



200th Step



Complete

Conclusion

The robot was able to discover near-optimal, and in many cases, optimal path. Finer resolutions of gridding may be used to improve the probability of discovering optimal solution at the cost of higher time complexity. Planning algorithms such as value iterations or policy iteration also have merit in solving this problem.