Automated Warehouse Scenario

[redacted]

Abstract

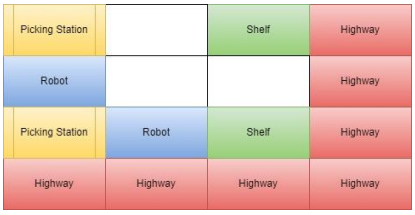
Automation has become a vital process in today's world. Artificial intelligence and robotics are the major technologies that are closely associated with automation. Tasks that we once thought would always require the human minds are now being replaced by using these technologies. Knowledge based tasks such as processing information, drawing inferences and conclusions, making decision have also been taken over by these technologies. One of the key areas where automation is booming these days is in the e-commerce domain. Many e-commerce businesses rely on robots to handle heavy laborious logistics like packaging of products, picking and moving them, managing inventories which is extremely tedious, time consuming and error prone when done by humans. Robots also increase time efficiency. In this project we implement automation using robots in warehouses to pick up and deliver products to fulfill consumer orders as quickly as possible.

Problem Statement

We are considering a scenario where robots are in charge of identifying, picking up and delivering products that are stacked in a warehouse. The primary task is to develop an algorithm which makes the robot identify the shelf which contains the desired product and carry that shelf over to a picking station in a time efficient manner. The overall set-up of the warehouse is such that it is made up of several rectangular grids. Some of these grids have shelves placed in them which are used to store the products, some other grids are identified as picking stations, while the other grids can be used by the robots to move from one place to another to access the shelves and the picking stations. The grids that are used by the robots to move around can be designated as highways and no shelves may be placed in these grids. The robots are allowed to move either horizontally or vertically along the grids. The robots are designed in such a way that they are flat and can travel underneath shelves to move from one grid to another. They pick up the required shelves by moving under them and lifting them. Once a robot picks up a shelf, they will not be able to travel underneath other shelves to move towards the picking stations. To ensure faster picking and delivering of shelves, we may deploy multiple robots and picking stations. The main goal here is to ensure that the robots do not collide with each other while moving around the grid, picking up or delivering products or while waiting idly for a pickup order. The algorithm that we are to develop needs to find the most optimal path that a robot needs to take to complete the given order in the least possible time.

Progress Made

As the basic step for this project, I installed and configured CLINGO in my local machine. Next, I have learnt the foundations of Answer Set Programming (ASP) and how to translate the facts written in ASP to CLINGO by using the right syntax. The next step was understanding and breaking down the problem statement into workable steps. The first major step towards the project was to construct the layout of the warehouse in the form of grids and establish this grid as the working environment for the given scenario. Once the grid was setup, I encoded the properties of each of the objects in the grid, which are the shelves, picking stations and robots. As the next step, I positioned the picking stations in the grid and allotted some of the grid cells as highway cells where only the robots can move and no picking station or shelf can be positioned. Next, I have assumed an initial position of the robots and shelves and have hard coded it to the environment. The figure below depicts the environment that currently exists.



Challenges Encountered

The major challenge that I am currently facing is dealing with the collision of robots. As there is more than one robot in the environment it is important to ensure that they don't collide. The constraint where the robots are allowed to move only vertically or horizontally makes it harder to move the robots without colliding into each other. It also increases the time taken by the robot to reach its destination since it has to avoid the path/cell of the other robot. Another problem is that the code takes a long time to execute as it is not optimized.

Plan to Resolve Issues

To solve the challenge of robot collision I will have to introduce some additional constraints that check for the positions of other robots. My algorithm should be able to come up with the next most optimal path that ensure there is no collision. One way this can be dealt with is by assigning priorities to the orders. The robot that is handling the high priority order is allowed to choose the most optimal path while the other robot can choose the next most optimal path. In a case where both robots have an equal priority task, they can both take their second most optimal path so that both tasks are completed in a fair amount of time rather than one task being completed very efficiently and the other task taking a very long time. In order to fix the issue with the code having longer execution times, I will need to research on simpler ways to transform and execute the ASP facts in CLINGO.

Itemized Tasks Completed

- Learning the basics of ASP and understanding its working.
- Installing and configuring CLINGO.
- Executing several simple and complex CLINGO problems to understand and get the hang of the syntax.
- Writing facts in ASP for the initial setup of the environment.
- Translating the ASP facts to CLINGO.
- Created the necessary constraints required for the given problem statement.
- Converted the constraints to CLINGO using proper syntax and verified its correctness.

Itemized Tasks To Be Completed

- Creating further constraints that are required to avoid the collision of two robots.
- Research and identify how effective assigning priorities to tasks will be.
- Finding the best logic to derive the optimal path for each of the robots.
- Work on decreasing the time taken to execute the code by using optimal algorithms.
- Ensure correctness of the program such that the final program is satisfiable in CLINGO.
- Generate the final documentations required for the project.