Course Project: Milestone 4 - Individual Project Report

Vamsi Krishna Yadav Loya
1226303691
vloya1@asu.edu

Abstract

The project aims to develop a simplified version of auto-mated warehouse scenario done in Amazon warehouses where robots are used to deliver the products to the pickup stations to fulfil orders. This report gives a clear picture of problem description, problem statement, approach to solve the problem, results and analysis of the ASP code written, conclusion, and future scope of this project.

Problem Statement

In this project I am working on a robot automated warehouse to deliver products to the pickup stations where the warehouse is a rectangular grid of any size whereas robots can move horizontally and vertically through adjacent cells. Our main goal is to deliver the products in little time as possible. To achieve this the robots should cover the least number of cells to reach the desired destination. Here the time calculated to deliver the products is nothing but the number of moves that a robot takes.

Project Background

Introduction:

Answer Set Programming (ASP) is a high-level programming language that is well-suited for solving NP-complete problems. ASP is based on the logical programming paradigm, where rules and facts are used to deduce new facts.
To solve the given problem statement, I needed to have a thorough understanding of ASP and develop the competencies needed to solve the problem using Answer Set Programming. I studied the syntax and semantics of ASP, as well as its problem-solving techniques, such as the Block problem and Queen's problem.
I also researched external online materials, including research papers, online courses, and tutorials, to gain a better understanding of ASP and its applications in solving real-world problems.

Objective:

The main objective of the project is to:

1. Develop a thorough knowledge of the problem statement, then list the aspects that must be present in the design.
2. Write input and output predicates to define the **problem in clingo and implement a dynamic pro**gramming approach to handle the changing status of inputs over time.
3. Writing hard ASP constraints to meet the need of project to fulfill all orders is efficient and meets the requirements of problem statement.
4. Tested the implementation of ASP code to ensure whether it is functioning and can solve the given test cases.

Algorithm:

1. First, we created initial states of the warehouse, including the location of the nodes, highways, robots, shelves, and pickup stations.
2. The main goal is to find the shortest path for each order from pickup stations to the shelves containing the required products.
3. Assigning a robot based on its availability.
4. I used a dynamic programming approach to generate actions. Defining a state as a tuple of (robot location, shelf location, products to be delivered) and creating a dynamic programming table to store the optimal number of moves to reach each state.
5. Using dynamic programming helped in constructing a plan very easily for each robot, including moves to pick up and move shelves, and moves to deliver products to pick up stations.
6. Generating an output as a list of actions for each robot at that time.

Input predicates:

The project's input predicates comprise the beginning positions of the various items, such as the grid's nodes, highway cells, robots, shelves, pickup points, and goods. There are further limitations, such as the fact that robots can only move horizontally or vertically and not diagonally. Robots are flat surfaces that can only handle one shelf at a time; they are unable to switch shelves. A robot or a shelf cannot also be initially carried by another robot or positioned in a cell that has been designated as a highway.

• **init(object(node,'n'),value(at,pair('x','y')))** – The term 'n' in the predicate represents a point on a rectangular grid where 'x' and 'y' are the co-ordinates of the node 'n'.
• **init(object(highway,'h'),value(at,pair('x','y')))** – The term 'h' in the predicate represents a highway cell locat-ed at 'x' and 'y' co-ordinates.
The rectangular grid's highway cells are a subset of those cells and have the same coordinates as the regular cells. Given that a corresponding fact of the first kind ex-ists and identifies the grid cell in question as a highway cell, it may be assumed that the highway cells have some extra information or functionality that sets them apart from normal cells.

Output predicates:

The activities the robots do at each time step to deliver the products are the output predicates for the project. The activities involve the robot picking up a shelf at a certain location, shifting the shelf from one place to another, and delivering the merchandise to the pickup station.

Robots must supply the precise number of units necessary for each product in an order, even if the delivery is made in many portions. At time steps 't' denoted by positive integers beginning at 1, actions may be carried out.
Further criteria are stated along with the form of atoms for actions in the following: Each robot specified by an instance may, but need not, do one action each time step:

• **occurs(object(robot,'r'),move('dx','dy'),'t')** - At time t, the robot is moving shelves from position (X,Y) to position (X+dy, Y+dy), where dx and dy correspond to movement (0,1), (1,0), and (0,1). (1, 0).
• **occurs(object(robot,'r'),pickup,'t')** - At time t, the robot "r" picks up the shelf at coordinates "X" and "Y."

Methodology:

After developing the required competencies, I created the solution methodology for the problem. I defined the objects and mapped them with their unique coordinates, such as the grid, node, picking station, robot, order, product, and shelf. I also created constraints related to the state, such as the well-known law of inertia, uniqueness of locations, and restrictions on the movement of robots and shelves.

I validated the constraints and ensured that the shelf stayed on a single node or robot, the robot could not be present at multiple locations simultaneously, and the shelf's location was distinct. By following this approach, I was able to solve the problem of automating the warehouse with robots effectively using Answer Set Programming and clingo program.

The development of the solution involved implementing a dynamic programming approach to generate actions. I defined a state as a tuple of (robot location, shelf location, products to be delivered) and created a dynamic programming table to store the optimal number of moves to reach each state. This approach helped in constructing a plan very quickly for each robot, including moves to pick up and move shelves and deliver products to pick up stations.

Finally, I tested the implementation of the ASP code to ensure that it was functioning and could solve the given test cases.

Approach to solving the problem.

Understanding the problem statement and identifying the requirements to automate the warehouse with robot.
Comprehending the assignment by studying the descriptive video and the description document.
Developing the competencies needed to complete the assignment, including studying Answer Set Programming, Clingo Programs, and problem-solving techniques like the Block Problem, Queen's problem. Building the solution methodology for the problem, which involves creating constraints, actions, and states for the Transition system problem, like the Block problem.
Defining the objects and mapping them with their unique coordinates, such as the grid, node, picking station, robot, order, product, and shelf. Creating constraints related to the state, such as the well-known law of inertia, uniqueness of locations, and restrictions on the movement of robots and shelves.
Validating the constraints and ensuring that the shelf stays on a single node or robot, the robot cannot be present at multiple locations simultaneously, and the shelf's location is distinct. By following this approach, the problem of automating the warehouse with robots can be effectively

solved using Answer Set Programming and clingo program.

Main Results

I have written some test cases in the input instances (.txt) files to check whether the code is working correctly or not and the code is executed using the command.

clingo KRRproject.txt instances/input_instance1.asp 0

Output is generated, below is the output screenshot and I have used table to represent the models and optimization for the test cases written.

Screenshot:

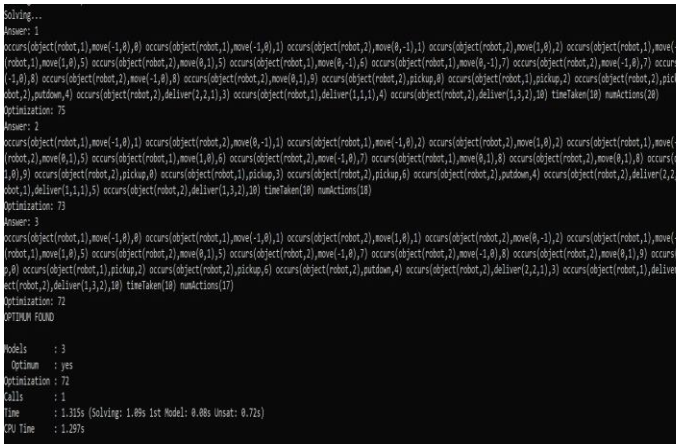

Table 1: ASP Models

Test cases	Model	Time	CPU-Time
input_instance1	**3**	**1.31s**	**1.29s**
input_instance2	**6**	**1.53s**	**1.33s**
input_instance3	**17**	**1.85s**	**1.78s**

Table 2: ASP Optimization

Test cases	Optimization	Time	CPU-Time
input_instance1	**72**	**1.31s**	**1.29s**
input_instance2	**20**	**1.53s**	**1.33s**
input_instance3	**31**	**1.85s**	**1.78s**

Analysis

The implementation of the project is completed successfully with the above results. The program has successfully executed the input given in the initial states, including the location of the nodes, highways, robots, shelves, and pickup stations. The program has also successfully found the shortest path for each order from pickup stations to the shelves containing the required products. The dynamic programming approach helped in constructing a plan very easily for each robot, including moves to pick up and move shelves, and moves to deliver products to pick up stations.

The program's input predicates included the beginning positions of the various items, such as the grid's nodes, highway cells, robots, shelves, pickup points, and goods. The program's output predicates were the activities the robots did at each time step to deliver the products. The activities involved the robot picking up a shelf at a certain location, shifting the shelf from one place to another, and delivering the merchandise to the pickup station.

The program used a dynamic programming approach to generate actions. It defined a state as a tuple of (robot location, shelf location, products to be delivered) and created a dynamic programming table to store the optimal number of moves to reach each state. Using dynamic programming helped in constructing a plan very easily for each robot, including moves to pick up and move shelves, and moves to deliver products to pick up stations.

The program has created output as a list of actions for each robot at that time. The activities the robots did at each time step to deliver the products were the output predicates for the program.

Conclusion

The project has successfully developed a simplified version of an automated warehouse scenario done in Amazon warehouses where robots are used to deliver the products to the pickup stations to fulfill orders. The program has been executed successfully, and it meets the requirements of the problem statement. The program's dynamic programming approach has helped to generate actions easily, including moves to pick up and move shelves and moves to deliver products to pick up stations. The program's input predicates, and output predicates were also successfully executed, which included the beginning positions of the

various items, such as the grid's nodes, highway cells, robots, shelves, pickup points, and goods.

Self-assessment

This report provides a clear and detailed explanation of the problem statement, background information on Answer Set Programming, the objective of the project, the algorithm used, input and output predicates, methodology, and approach to solving the problem. I did my job of explaining my way of approach to the problem and I did learn a lot of things working for this project mainly writing hard constraints in ASP, and during the process of learning deep into ASP I read some research papers which helped me in getting clear understanding of this language and how I can apply this to real world problems.

Future scope

The automation of warehouse operations using robots has become an essential trend in the logistics industry, and the future scope of this project is vast. One potential area of development is the implementation of machine learning algorithms to optimize warehouse operations by predicting order volumes and creating efficient delivery routes for the robots.

Another area of development is the integration of advanced sensor technology, such as lidar and sonar, to enable robots to navigate more accurately and avoid collisions with other robots and objects in the warehouse.

In conclusion, the development of automated warehouse scenarios using robots has significant potential in streamlining operations and enhancing efficiency in the logistics industry, and the use of Answer Set Programming and dynamic programming is an effective tool in developing solutions for this problem.