Automated Warehouse Scenario Project
Project Milestone - 4

Dhanush Kukatla
dkukatla@asu.edu

Abstract

The world we live in is changing as a result of automation. Online shopping trends have continued to increase in recent times, making it necessary for the development of new technology to meet the demands. As a result, the goal of this project is to provide a solution for automating the warehouse. The optimal solution is described in this article for the instances provided and documentation of the problem is also included. This paper contains project background, solution approach, results and analysis, conclusion, and opportunities for future work.

Keywords – KRR, Clingo, ASP, Automated warehouse.

Problem Statement

Automated warehousing scenario project mainly consists of delivery of goods from shelves using robots for the fulfillment of orders. The warehouse is split into cubical grids such that a few grids are appointed as picking stations where orders for the items are placed, and highways where it is an absolute necessity that no shelves are placed on them.

The solution to the problem consists of an action plan for each robot for delivering products of the order to the respective picking station where the order was placed. Since time duration is measured for each step and each robot can only perform one task at a time.

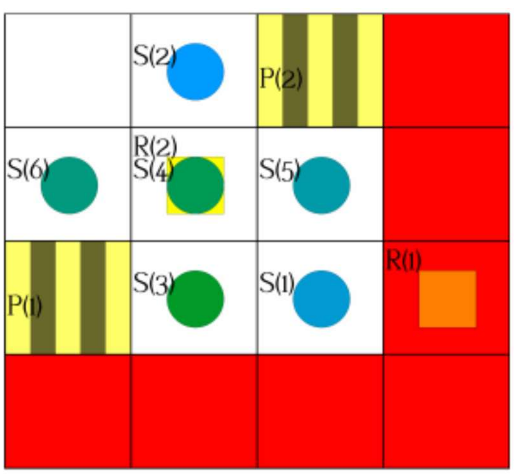

Figure 1: Warehouse grid cells representation

The objective of this problem is to deliver all orders in the least amount of time achievable. The robots should be able to travel within the cells (warehouse) without running into one another as they pick up packages, place down the shelves, deliver goods, or just when the robots are idle. There should be no shelves along the Grid cells that are marked as highways. The dimensions of the robots are sleek, and they can slide below the shelves to pick them up.

Project Background

The logistics and delivery-based industry is where most of the warehouse is dealt with, and to manage this warehouse with manpower it is very difficult. Not all warehouses are automated and self-manageable, but it will reduce the requirement of the manpower.

So, the solution to this problem is the Automated Warehouse project. This solution will be more efficient with negligible errors, cost, and time efficient. The are many specialized areas under Artificial Intelligence, one such is the Knowledge Representation and Reasoning.

KRR is associated with the representation of knowledge in high - level languages and its automated manipulation so that computers can make wise judgments based on the encoded knowledge.

The course work of the knowledge representation and reasoning course is well structured to learn the concepts. Some of the important concepts are:

- First order logic and knowledge representation
- Theory of Answer Set Programming(ASP)
- Reasoning about actions
- KRR with uncertainty
- Ontology languages

For this problem, I will be using Answer Set programming (ASP) along with Clingo (ASP tool) and the knowledge gained from the course.

My approach to solving the problem

I was very keen on solving this problem after hearing about it. After careful examination of the given problem, I felt it was not that easy. I started to bread down the problem into subtasks and as I continued, the problem felt to be laborious. Moving on, I penned down the constraints of the problem. The subtasks were, completing all the orders in minimum time, writing code for moving shelves, pickup shelves, putdown shelves and deliver shelves to the picking station. Before starting to solve the problem I went through clingo guide named *‘A User’s Guide to gringo, clasp, clingo, and iclingo’ : http://wp.doc.ic.ac.uk/arusso/wp-content/uploads/sites/47/2015/01/clingo_guide.pdf* and the course video lectures.

Another problem is a lack of information about the requirements required to demonstrate that the recommended technique is optimal. The main aim of this project is to deliver all the shelves in the shortest time. Initially I created atoms and the environment to deal with the problem. There are main constraints that have looked into, they are, robot can’t deliver more than one shelves at a time, no two robots can deliver the same shelve, the location of the robot doesn’t change at time T+1 if It doesn’t move, and similarly, the location of the shelve doesn’t change if the shelve is not picked up by the robot. The product on the shelf is still marked with same location if the shelf is not picked by the robot. There can’t be a pickup station or shelves present along the grid cells that are marked as highways.

There are state of the operations along the process, they are, when the shelf is getting picked or dropped off then state changes, and the state changes when product is getting delivered, and the during the movement of the robot is noticed then the state changes.

Results

The solution for this problem I tested with five different scenarios which were provided along with the problem itself. The main.asp file is where the logic of the problem is situated. The final.asp problem is the file where the matrix of warehouse grid is situated. The input.asp file is where the input from the instance files are being loaded and used for execution. These are the files that I will be using to solve the automated warehousing problem.

FILE NAME	OPTIMAL NO. OF STEPS	CPU TIME
Inst1.asp	64	1.063s
Inst2.asp	72	3.016s
Inst3.asp	31	2.141s
Inst4.asp	20	1.359s
Inst5.asp	31	3.094s

Table-1 : Optimal number of steps for each instance

The above table shows the CPU time and the number of optimizations taken place. Multiple iterations are done to find out the efficient solution, so that the final answer presented satisfies the requirements of the problem.

Problem	**Optimal Solution**
Inst1.asp	**Command line**: clingo main.asp simpleInstances\inst1.asp -c n=10 **Optimal output:** Answer: 1 event(obj(rob,1),activity(-1,0),0) event(obj(rob,1),activity(-1,0),1) event(obj(rob,2),activity(0,-1),1) event(obj(rob,2),activity(1,0),2) event(obj(rob,1),activity(-1,0),3) event(obj(rob,2),activity(0,1),5) event(obj(rob,1),activity(0,-1),6) event(obj(rob,2),activity(0,-1),7) event(obj(rob,1),activity(0,1),8) event(obj(rob,2),pickup,0) event(obj(rob,1),pickup,2) event(obj(rob,2),pickup,6) event(obj(rob,1),pickup,7) event(obj(rob,2),putdown,4) event(obj(rob,1),putdown,5) event(obj(rob,2),delivery(2,2,1),3) event(obj(rob,1),delivery(1,1,1),4) event(obj(rob,2),delivery(3,4,1),8) event(obj(rob,1),delivery(1,3,4),9) timeTaken(9) numActions(19) Optimization: 64
Inst2.asp	**Command line**: clingo main.asp simpleInstances\inst2.asp -c n=13 **Optimal output:** Answer: 16 event(obj(rob,1),activity(-1,0),0) event(obj(rob,1),activity(-1,0),1) event(obj(rob,2),activity(1,0),1) event(obj(rob,2),activity(0,-1),2) event(obj(rob,1),activity(-1,0),3) event(obj(rob,2),activity(0,1),5) event(obj(rob,1),activity(1,0),6) event(obj(rob,2),activity(-1,0),7)

	event(obj(rob,2),activity(-1,0),8) event(obj(rob,2),activity(0,1),9) event(obj(rob,2),pickup,0) event(obj(rob,1),pickup,2) event(obj(rob,2),pickup,6) event(obj(rob,2),putdown,4) event(obj(rob,2),delivery(2,2,1),3) event(obj(rob,1),delivery(1,1,1),4) event(obj(rob,2),delivery(1,3,2),10) timeTaken(10) numActions(17) Optimization: 72
Inst3.asp	**Command line**: clingo main.asp simpleInstances\inst3.asp -c n=13 **Optimal output:** Answer: 20 event(obj(rob,1),activity(0,-1),0) event(obj(rob,1),activity(-1,0),1) event(obj(rob,1),activity(0,-1),3) event(obj(rob,2),activity(0,-1),3) event(obj(rob,1),activity(1,0),5) event(obj(rob,2),activity(1,0),5) event(obj(rob,2),pickup,0) event(obj(rob,1),pickup,2) event(obj(rob,1),delivery(2,4,1),4) event(obj(rob,2),delivery(1,2,1),6) timeTaken(6) numActions(10) Optimization: 31
Inst4.asp	**Command line**: clingo main.asp simpleInstances\inst4.asp -c n=13 **Optimal output:** Answer: 10 event(obj(rob,1),activity(-1,0),0) event(obj(rob,1),activity(-1,0),1) event(obj(rob,2),activity(0,-1),1) event(obj(rob,2),activity(1,0),2) event(obj(rob,1),activity(-1,0),3) event(obj(rob,2),pickup,0) event(obj(rob,1),pickup,2) event(obj(rob,2),delivery(3,2,2),3) event(obj(rob,1),delivery(1,1,1),4) event(obj(rob,2),delivery(2,2,1),4) timeTaken(4) numActions(10) Optimization: 20
Inst5.asp	**Command line**: clingo main.asp simpleInstances\inst5.asp -c n=13 **Optimal output:** Answer: 22 event(obj(rob,1),activity(-1,0),0) event(obj(rob,1),activity(-1,0),1) event(obj(rob,1),activity(-1,0),3) event(obj(rob,2),activity(-1,0),3) event(obj(rob,1),activity(1,0),5) event(obj(rob,2),activity(0,1),5) event(obj(rob,1),pickup,2) event(obj(rob,2),pickup,4) event(obj(rob,1),delivery(1,1,1),4) event(obj(rob,2),delivery(1,3,4),6) timeTaken(6) numActions(10) Optimization: 31

Table-2: Efficient solution for each instance input

Conclusion

The project is all about solving the real-world problem of automating a warehouse. I have provided an efficient solution for this problem. At first I found the solution to just satisfy all the fundamentals, that is to complete all the orders. Later I enhanced the code to find the efficient solution for the given problem. that are the files that I will be using to solve the automated warehousing problem.

There are a lot of this I learnt to deal with this question and also learnt many other things that can be later implemented in real-world applications. I even debugged and analyzed each and every component of the code. There are many components that affect the optimal solutions in different ways. The time constraint is one such, it plays a major role as we need to finish all the deliveries in the least amount of time.

Because it supports recursive definitions, optimization instructions, weight restrictions, aggregates, default negations, and external atoms, ASP has proven useful in tackling difficult problems. ASP is being used in a wide range of areas under Artificial Intelligence and other domains as well. Therefore, I present the best optimal solution for the automated warehousing problem to deal with this problem.

Future Work

This project is a fictionalized version of the real-world scenario of automated warehouses like eBay, Amazon, and many more companies. This challenge may be updated and extended in the future to include more limitations and features, bringing it one step closer to the real-world operation of automated warehouses. Furthermore, this project can be extended in many ways like,

- The project must incorporate additional structural reliability and boundaries.
- Numerous warehouses might be included in the scenario.
- Automate a warehouse along with some human involvement.
- Both stable and adaptive Shelf restocking options can be offered.

References

A User's Guide to gringo, clasp, clingo, and iclingo : *http://wp.doc.ic.ac.uk/arusso/wp-content/uploads/sites/47/2015/01/clingo_guide.pdf*

Routing Driverless Transport Vehicles in Car Assembly with Answer Set Programming–Extended Abstract : *http://www.kr.tuwien.ac.at/events/taasp18/papers/TAASP_2018_paper_10.pdf*

Knowledge Representation and Reasoning, CSE 579, Joo-hyung Lee – Lecture Slides and videos