Automated Warehousing Scenario

Written by

Yash Vakil*, Akshay Kumar*, Anish Oswal*, Pranay Singhal*

*Arizona State University

Abstract

Warehousing in logistics management is a crucial component for overseeing transportation and storage, and retrival of inventories. Organizations are working to come up with compelling and effective management to reduce manpower. This paper describes our solution to the particular problem of the fulfillment of orders in a distribution center.

Problem Statement

Automated Warehousing Scenario involves delivering of products from racks using robots for the fulfillment of orders. The warehouse is divided into cells such that a few cells are designated 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 of the problem consists of an action plan of each robot for delivering products of the order to the respective picking station where the order was placed.

The problem statement begins with the underlying description of the present situation of the warehouse consisting of the location of the robot, shelves, picking stations and highways.

For computing the solution, various constraints - few of those are mentioned below - need to be satisfied

- The location of robots and shelves need to be unique i.e two objects cannot be on the same location at the same time and an object cannot be in two locations at the same time.
- Products of the same type may exist on different shelves.
- Robots can pick up and put down a shelf but can only carry one shelf at a time.
- Robots travel under shelves but can no longer move under a shelf it it is already carrying another shelf.
- Robots are restricted to only move vertically or horizontally to the next cell.
- A robot is restricted to perform only one action at a time.

Project Background

The problem statement is part of the ASP Challenge 2019, and thus needed to be implemented in an ASP language. clingo is an ASP system that offers a straightforward and powerful way to solve combinatorial problems, plus our prior knowledge from the course seemed to add to the decision of using clingo.

For realizing the solution we revisited our previous programming assignments of the block world, and monkey and banana problem to understand how to approach the problem. We also glossed through the clingo documentation to get a stronger understanding of the language before writing the code.

Team Approach

We conceptualized the genuine working of the issue by sketching the working of the system given in the initialization files on a paper to get an idea of the various actions and the constraints over the problem. Alongside this, we looked into the initialization files to understand the input and how to convert it into a simple format for easier implementation. The methodology we pursued was iterative to comprehend the problem in a better manner.

We began with no restrictions, what we did was execute the movement of robots in the given framework with the goal state as a particular node. From that point onward, we began adding constraints and actions related to all the objects of the system. With racks, we started by coding the 'pickup' and 'putdown' actions of the robots with an end state as a node with a shelf on top of a robot. This methodology helped us in troubleshooting the code and finding the ideal answer for each case. At first, we were confronting a ton of issues with the imperatives. Debugging the codes was perhaps the hardest thing we needed to manage as a group.

Something else we did as a group was allowing each member to code their own ideas during the initial stages of the project so that we were open to other ways a person can approach the problem. It helped us a great deal in thinking of numerous arrangements and contending over the rightness of these solutions. We additionally had regular team meetings to manage the project.

```
Answer: 6
occurs(object(robot,1),move(-1,0),0) occurs(object(robot,1),move(-1,0),1) occurs(object(robot,2),move(1,0),1) occurs(object(robot,2),move(0,-1)
,2) occurs(object(robot,1),move(-1,0),3) occurs(object(robot,2),pickup,0) occurs(object(robot,1),pickup,2) occurs(object(robot,2),deliver(2,2,1
),3) occurs(object(robot,1),deliver(1,1,1),4) occurs(object(robot,2),deliver(3,2,2),4) timeTaken(4) numActions(10)
Optimization: 20
OPTIMUM FOUND

Models       : 6
  Optimum    : yes
Optimization : 20
Calls        : 1
Time         : 0.439s (Solving: 0.29s 1st Model: 0.01s Unsat: 0.16s)
CPU Time     : 0.431s
```

Figure 1: The optimal output of the program for a specific instance.

Result and Analysis

We have tested our code for all the instance files provided in the project.We have minimized the time taken to, and the total number of actions taken by the robots to deliver the required order to given picking station.

- Small parts of rules needed multiple updations at times during the project development cycle due to their high complexity.
  Example-As soon we increased the robots and orders requirement there were some random movements which we had to debug taking most of our time.
- All the instances ran successfully in optimal time with the robot reaching the goal states. Optimized plan is achieved by minimizing the time stamps and action taken by robots collectively.
- Firstly we observe that as we increase the number of robots, the time taken for delivery reduces due to simultaneous delivery. Having multiple robots on the grid allows us to utilize as many empty cells as possible on which the robots can navigate and deliver the orders without any accident.
- Similarly, As we increase the number of orders, the time taken by robots also increased. This is due to single shelf carrying constraint over the robots due to which it can carry only a maximum of one shelf.
- Ultimately, even if the number of robots or shelves increase, there were no collisions observed. Eventually all instances were satisfied and the robot was reaching the goal state in minimum number of steps.

Conclusion and Opportunities

The project taught us a lot of things overall especially in the fields of teamwork and ASP. The team being divided into sub-teams led us to working parallelly and validate our work from others. This opened us to criticism and helped us in understanding and working with teamwork. Due to extensive usage of propositional logic required for the project, we understood the basic as well as the complex concepts of answer set programming and learnt about the language in a greater depth. We also got a hands-on experience about a very practical scenario where answer set language would be the most optimal language to use to find solutions to an automated robot scenario.

On completion of the project, the script was able to solve all the given instances successfully with the minimum number of steps taken. This solution for all of the given instances was the most optimal solution found by our program. The robot traversed through the given workspace avoiding any collisions and also successfully completing all the orders in minimum time. This ensured the overall optimality of the solution as well as the concluded the development cycle of the project.

The applications of the project are widespread and be used in a variety of fields. Opportunities for an automated warehouse can be applied to a lot of industries which handle product warehouses and orders. This system of robots handling the product orders efficiently is fast, efficient and reliable when compared to manual labour.

Automatic processes such as storage, retrieval, relocation and delivery of goods becomes a lot more flexible with the application of robots as they can be programmed to complete complex tasks and even work together to complete orders in a much faster fashion in specific scenarios. The scenario can be modified to be integrated with autonomous agents such as delivery drones which can provide shipment via door-to-door services. This saves a lot of mechanical effort as well as time which can be utilized elsewhere. This robust environment has a lot of advantages. Due to such features and a futuristic solutions for not just this but all kinds of problems, the usage of robots in warehouses and other fields should be highly promoted as well.

Contributions

- **Akshay Kumar**
  From the starting of the project, Akshay has spent a lot of time in the movement action of robots. He has also looked into the movement shelves following pickup action made by the robots. He has also been part of the regular discussion that we had every week to keep the track of progress. He made quite an effort in dealing with the movement of multiple robot and shelves which was causing a lot of conflicts with the constraint given in the problem.It even generated a few irrelevant outcomes consuming a lot of his time.

  Akshay and Yash were part of the Team A which looked into the action and delivery of the product to the picking station. Akshay's part was to look into the action constraint of the robot and shelves. It involved the movement of robots and shelves with the constraint on robots.

- **Anish Oswal**
  Initially, Anish's specific contributions to the project were eccentric to designing of a project plan while creating and designating the tasks among the fellow team members. The overall team of four was divided in two teams with respect to his approach on the initial project plan. Being a part of the team designing the state constraints for the robots and the other objects such as shelves and the picking stations, he collaborated with Pranay on programming of the constraints. Occasionally, all the team members helped each other for complex parts of the constraints so as to maintain a smooth flow of work. Along with Pranay, he also validated the action constraints designed by the other subteam which helped towards faster completion of the project as the code was properly tested at each stage of the development cycle.

  Specific duties such as designing rules pertaining to the movement of the robot, avoiding collision of the robots in the warehouse workspace were some of his essential duties in contributing to the conclusion of the project.

- **Pranay Singhal**
  Pranay's role was to manage the overall project progress and integrate the different threads of the sub teams which were being prepared into the master file. After analyzing the problem, the core team was divided into 2 subteams.

  Pranay and Anish were part of the Team B which overlooked the constraint problems for various agents. Along with Anish, he contributed to the part of the state constraint. He successfully tested the program for several test instances to provide proper direction to the teams.Apart from that, the common contribution was to help others with complex rules and design them as a team.

  His specific duties included designing state constraints for the shelves, map them with the constraints of robots and pickup stations along with designing rules to find the optimal path in minimum time stamps. In the end, the team in the whole prepared the presentation where Pranay handled the conclusion and future work section.

- **Yash Vakil**
  Yash offered a lot of time at the beginning of the project to sketch and conceptualize the diverse scenarios of the problem statement.

  After explaining the various possible test cases to the rest of the group, to help with parallelizing of tasks within his subgroup, he implemented the necessary logic for generating the four actions i.e. robot movement, picking up of a shelf, putting down of shelf and delivering of product. Along these lines, he also implemented the effects these actions have on the state of the objects. Following that, he continued with the task of actualizing the constraints for shelf 'pickup' and 'putdown' by the robots.

  Apart from these errands, his other work as a team member was to debug the code to identify bugs and find out whether the solution yielded by the program was, in fact, a solution or not. He also implemented the logic for finding an optimal path from the solutions by minimizing the time taken and the number of actions executed by the robots.

References

Technische Universität Wien (Austria) and the University of Genoa (Italy), *Answer Set Programming Challenge (2019)*, https://sites.google.com/view/aspcomp2019

Gebser, M.; Kaminski, R.; Kaufmann, B.; Ostrowski, M.; Schaub, T.; and Thiele, S. *A User's Guide to gringo, clasp, clingo, and iclingo*, The University of Texas at Austin,