Automated Warehouse Scenario
Individual Project Report

Lucy Zhang
Ira A. Fulton Schools of Engineering, Arizona State University
lzhan370@asu.edu

Abstract

This report presents an optimized, automated solution for warehouse order fulfillment using Clingo, an answer set programming system. The model represents an automated warehouse as a grid featuring distinct highways, picking stations, and movable robots and shelves. Each robot can transport a single shelf laden with products for order completion. To prevent collisions, I've instituted specific rules. The goal is to devise a plan that minimizes steps to fulfill all orders. The strategy involves encoding warehouse status and order details into a structured format and defining possible robot actions like movement, shelf retrieval, product delivery, and shelf placement. The report showcases the model's effectiveness in enhancing order fulfillment efficiency in diverse warehouse environments.

Problem Statement

The focus of this research is to optimize the process of order fulfillment within the context of an automated warehouse system, wherein robots are assigned tasks to pick up and deliver products. The warehouse is structured as a rectangular grid punctuated with shelves, acting as obstacles, and distinct areas designated as highways and picking stations as in Figure 1. Each robot can execute only a single action per unit of time and must maneuver to avoid collisions. These robotic entities are constrained from executing diagonal movements or bypassing any cells.

The orders are delineated in a specific structured format, necessitating the robots to deliver specified products in exact quantities. Multiple shelves may store identical products, and the robots are permitted to relocate a shelf, provided no other shelf obstructs its path. In instances where the desired shelf is blocked by another, the robot must initially move the obstructing shelf away.

The crux of the challenge lies in developing an answer set programming model utilizing Clingo, which can generate an optimal plan for robotic actions. This plan must ensure the fulfillment of all orders, minimize the number of steps taken, and adhere to the constraints specific to the warehouse setting, thereby maximizing the warehouse's overall efficiency.

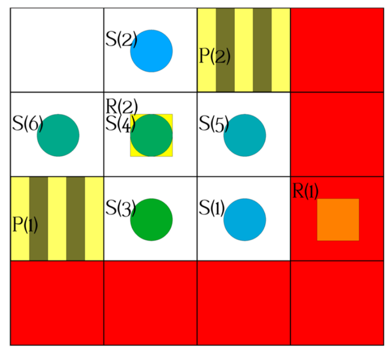

Figure 1: Representation of a 4x4 grid with 2 robots (R), 6 shelves (S), 2 picking stations (P), and 7 highway cells in red.

Project Background

The primary objective of this project was to discern the minimal number of steps required for robots to complete tasks within the provided constraints. In this context, the initial stride towards this goal was to understand and leverage the principles of Knowledge Representation and Reasoning (KRR) in conjunction with Clingo. By comprehending concepts within KRR, such as logic, knowledge representation, and first-order logic, alongside the practical application of Clingo, I was able to build up a skill set that would permit me to model intricate scenarios effectively, deliberate on the optimal course of action, and find the most efficient solutions. The fifth week's lecture, which focused on reasoning about actions and simple transition systems, proved particularly relevant to this project, as these ideas directly correspond to the movements and operations of the robots within the context of the warehouse scenario.

A comprehensive understanding of the problem and the given constraints of the project was my secondary achievement, attained through a meticulous perusal of the description. My most significant accomplishment was the thorough enumeration of all constraints and the initialization of the states of each object.

The journey towards the successful execution of this project wasn't without its share of challenges and complications. One of the predominant hurdles was the identification of the optimal solution, which was complicated by the existence of a multitude of potential solutions that were equally valid. This challenge was further amplified by the requirement to minimize the overall time for order fulfillment, thus introducing an additional level of complexity to the problem. Another significant obstacle was the amount of effort required to identify and rectify bugs within the script.

Approach to Solve the Problem

The project started with a thorough familiarization with the tools required, specifically Clingo and Answer Set Programming (ASP), which were critical for the successful execution of the project. A comprehensive review of concepts in Knowledge Representation and Reasoning (KRR) was also carried out, establishing a robust theoretical base for the project.

The next step was to understand in depth the complexities of the automated warehouse problem at hand. This understanding allowed for the identification of the key objectives of this project, with the primary goal being to fulfill all orders in as little time as possible, subject to certain constraints. Constraints include:

- Warehouse is a rectangular grid, each cell defined as a node.
- Robot can move forward, backward, left, and right.
- Robot cannot have diagonal movement.
- Robot cannot jump or skip over any cell.
- Same product may locate in more than one shelves.
- Robot picks up the entire shelf.
- Robot that carries a shelf cannot be on the same cell with another shelf.
- Robot needs to move away the shelf that blocks desire shelf.
- Robot can be on any cell if not carry a shelf.
- No more than one shelf can be on the same cell at the same step.
- Robot can perform any number of actions per step.
- Robot action can only be move, pick up, put down, or remain idle.
- No more than one robot can be on the same cell at the same step.
- Robots cannot swap position.
- Grid cells are highways.
- No shelves can be put down on highway.

These constraints were meticulously documented, providing a clear understanding of the problem's boundaries and parameters for the solution. Key objects integral to the project were declared successfully: 'robot', 'highway', 'shelf', and 'picking station'. Each declaration not only defined the initial position but also elaborated on the functionality of these objects within the warehouse system.

The project then moved onto finalizing all given constraints to ensure all rules and limitations were accurately represented in the ASP model. An assessment was carried out to add any additional constraints required to increase the realism or efficiency of the system. This completed model was then subjected to rigorous testing and debugging to address any issues or errors in the ASP code.

Upon achieving a working solution, the focus shifted towards optimization. The aim was to ensure the most efficient use of resources and minimize the time for order fulfillment, delivering a highly optimized solution for the automated warehouse problem.

Main Results and Analysis

The results of each provided instance exhibit unique outcomes concerning the minimum steps and total movements required, due to the distinctive scenarios presented by each instance. These outcomes are clearly delineated in Table 1. All instances share the common features of having two robots, a 4x4 grid size, and identical highway locations. However, they diverge with respect to other characteristics. Instance 1 encompasses two picking stations, six shelves, four products, and three orders. In contrast, Instance 2 involves two picking stations, five shelves, three products, and two orders. Instance 3 is characterized by a single picking station, six shelves, four products, and two orders. Similarly, Instance 4 features two picking stations, six shelves, two products, and three orders. Finally, Instance 5, unique in its requirements, hosts one picking station, six shelves, four products, and only one order to be fulfilled.

Inst	Min Steps	Movements
1	13	24
2	11	22
3	7	11
4	10	20
5	6	11

Table 1: Minimum steps and total movements for the optimum output of each instance.

As denoted in Table 1, the fastest order fulfillment was accomplished in Instance 5, with a minimal six steps and a total of eleven movements. The efficiency can be attributed to the fact that there was only one order to be completed, thus lessening the workload on the robots. Notably, the robots' performance was enhanced by their capability to perform multiple actions in one timestep - for instance, moving to a shelf cell and picking up the product within the same timestep. Hence, despite the varying complexities of the instances, the robots demonstrated commendable performance throughout.

```
occurs(object(robot,2),move(-1,0),1) occurs(object
(robot,1),move(-1,0),1) occurs(object(robot,1),mov
e(-1,0),2) occurs(object(robot,2),move(0,1),3) occ
urs(object(robot,2),move(0,-1),5) occurs(object(ro
bot,1),move(-1,0),5) occurs(object(robot,2),move(1
,0),8) occurs(object(robot,2),move(1,0),9) occurs(
object(robot,1),move(1,0),9) occurs(object(robot,1
),move(0,-1),10) occurs(object(robot,2),move(0,-1)
,11) occurs(object(robot,1),move(1,0),12) occurs(o
bject(robot,1),move(0,-1),14) occurs(object(robot,
2),move(1,0),14) occurs(object(robot,2),pickup,2)
occurs(object(robot,1),pickup,3) occurs(object(rob
ot,2),pickup,10) occurs(object(robot,1),pickup,11)
 occurs(object(robot,2),putdown,7) occurs(object(r
obot,1),putdown,8) occurs(object(robot,2),deliver(
1,3,4),4) occurs(object(robot,1),deliver(1,1,1),7)
 occurs(object(robot,2),deliver(3,4,1),12) occurs(
object(robot,1),deliver(2,2,1),15)
Optimization: 13
```

Figure 2: Optimal output for Instance 1

```
occurs(object(robot,2),move(0,1),1) occurs(object(
robot,1),move(0,-1),1) occurs(object(robot,1),move
(-1,0),2) occurs(object(robot,2),move(-1,0),3) occ
urs(object(robot,1),move(1,0),4) occurs(object(rob
ot,2),move(0,-1),5) occurs(object(robot,1),move(0,
1),5) occurs(object(robot,1),move(0,1),6) occurs(o
bject(robot,2),move(1,0),7) occurs(object(robot,1)
,move(-1,0),7) occurs(object(robot,1),move(-1,0),8
) occurs(object(robot,2),move(1,0),9) occurs(objec
t(robot,1),move(0,-1),9) occurs(object(robot,1),mo
ve(-1,0),14) occurs(object(robot,2),move(0,-1),14)
 occurs(object(robot,2),pickup,2) occurs(object(ro
bot,1),pickup,3) occurs(object(robot,2),pickup,8)
occurs(object(robot,2),putdown,6) occurs(object(ro
bot,2),deliver(1,1,1),4) occurs(object(robot,1),de
liver(1,3,2),15) occurs(object(robot,2),deliver(2,
2,1),15)
Optimization: 11
```

Figure 3: Optimal output for Instance 2

```
occurs(object(robot,1),move(-1,0),2) occurs(object
(robot,1),move(0,-1),3) occurs(object(robot,1),mov
e(0,-1),5) occurs(object(robot,2),move(1,0),6) occ
urs(object(robot,2),move(0,-1),7) occurs(object(ro
bot,1),move(1,0),7) occurs(object(robot,1),move(0,
1),12) occurs(object(robot,2),pickup,3) occurs(obj
ect(robot,1),pickup,4) occurs(object(robot,1),deli
ver(2,4,1),6) occurs(object(robot,2),deliver(1,2,1
),12)
Optimization: 7
```

Figure 4: Optimal output for Instance 3

```
occurs(object(robot,2),move(0,-1),1) occurs(object
(robot,1),move(-1,0),1) occurs(object(robot,1),mov
e(0,-1),2) occurs(object(robot,2),move(-1,0),3) oc
curs(object(robot,1),move(-1,0),3) occurs(object(r
obot,2),move(0,1),5) occurs(object(robot,1),move(0
,-1),5) occurs(object(robot,1),move(1,0),6) occurs
(object(robot,2),move(0,1),6) occurs(object(robot,
2),move(1,0),7) occurs(object(robot,2),move(-1,0),
12) occurs(object(robot,2),pickup,2) occurs(object
(robot,1),pickup,4) occurs(object(robot,2),pickup,
11) occurs(object(robot,1),pickup,11) occurs(objec
t(robot,2),putdown,4) occurs(object(robot,1),putdo
wn,7) occurs(object(robot,1),deliver(3,2,2),12) oc
curs(object(robot,1),deliver(2,2,1),15) occurs(obj
ect(robot,2),deliver(1,1,1),15)
Optimization: 10
```

Figure 5: Optimal output for Instance 4

```
occurs(object(robot,2),move(-1,0),3) occurs(object
(robot,1),move(-1,0),3) occurs(object(robot,1),mov
e(-1,0),8) occurs(object(robot,2),move(0,1),8) occ
urs(object(robot,2),move(0,1),11) occurs(object(ro
bot,1),move(-1,0),11) occurs(object(robot,2),move(
1,0),12) occurs(object(robot,2),pickup,7) occurs(o
bject(robot,1),pickup,9) occurs(object(robot,2),de
liver(1,3,4),9) occurs(object(robot,1),deliver(1,1
,1),12)
Optimization: 6
```

Figure 6: Optimal output for Instance 5

Conclusion

This research study effectively modeled an automated warehouse system, demonstrating the implementation of answer set programming using Clingo to optimize the process of order fulfillment. The proposed model successfully dealt with complex system dynamics, factoring in various elements such as highways, picking stations, movable robots and shelves. The computational model was able to generate an optimal plan for robotic actions, fulfilling all orders in the minimum number of steps and under specific constraints. The model's efficiency was substantiated across multiple distinctive scenarios, with robots demonstrating high performance and the capability to execute multiple actions within a single timestep.

The study underscored the potential of answer set programming in warehouse automation, managing to expedite the process of order fulfillment. By using intelligent agents such as robots to handle complex tasks and optimize actions, the study provided a glimpse into the future of warehouse management, where such systems could greatly improve efficiency and productivity.

Opportunities for Future Work

While the study has provided valuable insights, there are several avenues for further research and refinement. For instance, the current model could be adapted to incorporate more realistic scenarios, such as handling larger warehouse sizes, increased number of robots, shelves, and varying order complexities. Moreover, the model could be expanded to consider additional operational constraints, such as robot battery life or time windows for order fulfillment.

In terms of the optimization, future work could explore different strategies for reducing the number of steps, such as shortest path algorithms or swarm intelligence techniques for collaborative work among multiple robots. Additionally, it would be interesting to investigate the impact of adding more complex actions that the robots could perform, like stacking shelves or prioritizing certain types of orders.

Lastly, the introduction of machine learning techniques into the current model could open up new possibilities. For instance, reinforcement learning could be used to allow the robots to learn from their past actions and continuously improve their performance over time. Such enhancements could make the model more robust, adaptable, and even more effective at minimizing steps for order fulfillment.