Automated Warehouse

*Portfolio Report for CSE 579 as a partial fulfillment for Master of Computer Science at ASU

Haibin Liang
Biodesign Institute
Arizona State University
Tempe, Arizona, United States of America
hliang7@asu.edu

***Abstract*—The project I have chosen to complete is "automated warehouse". This individual report is to provide a statement of the problem, a summary of the work I have done, issue or challenges and the approach I used to solve them.**

***Index Terms*—Answer set programming, Knowledge Representation and Reasoning, Automated warehouse.**

I. Introduction

Warehouse automation is the strategies to automate the movement of inventory into, within, and out of warehouses to customers with minimal human assistance [1].For centuries, warehouse operations are labour intensive. There is an advance in finding optimal strategies for planning and control of warehouse systems in the last few decades [2]. In the modern day supply chain system, the development of automated warehousing greatly reduce the labour, time and space cost for wide variety of business operation [3]. The optimization of automation includes, eliminating labor-intensive duties that involve repetitive physical work and manual data entry and analysis [4].Automated warehousing are used in many E-commerce firms such as Amazon,Walmart, Target, Best Buy..etc. The fully-automation or semi-automation of warehousing provides cost effective, time efficiency and quality improvement of the services for these business [5].

In this project, I will propose a solution to automated warehousing to fulfill request of customers by having an knowledge-based algorithm to control the robots to fulfill the order required while adhering to the constraints.

I will define set of constrains in Answer Set Programming to reduce the chance of collision, pick up and deliver the items at the right location and in a right logistic. In conventional programming, such constrains will be more technically challenging. However, the use of ASP, which is specially designed for examine the conflict of logic, finding the number of iteration, time-space-compute, will be a more efficient way to give solution to the problem stated.

We use rectangular grid as a representation of the warehouse. Within the boundary of warehouse, the robot is able to move horizontally or vertically between adjacent cells. To complete the given tasks, the robot is required to carry the shelves with the required products to designated picking stations. The robots have the the following characteristics, they are flat and can move under the shelves to pick it up and move. Figure 2 shows a robot similar to those described in the problem. If the robot carrying a shelf not fitting under another shelf, the shelves have to be moved out of the way in advance.

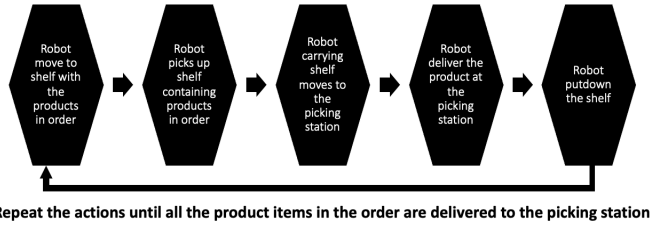

Figure 1. A schematic of the process for the robots to complete the task. The goal of this problem is to have an algorithm directing the robot to fulfill all orders in as little time as possible. The time is counted in steps and each robot may perform (or not to perform) one action per time step.

As the robots moving around, picking up, or putting down shelves, delivering products or idling, there should not be any collisions. The grid cells are designed as highways, and no shelves should be put on such cells.

II. Description of solution

This is a very challenging project. It involve a lot of consideration in multiple tasks and constrains. There's a lot going on at every time step. The course materials in module five highlights the knowledge I needed to know about this project. Therefore, I revised the videos and knowledge in week 5 for several times, especially the block-world problems before I started this project. And then, I followed the recommendation stated in the project walk-through video, that I tried to work on the specific robot actions one at a time.

A. Setting constrains on the robots' movement

The movement of the robots are the backbone of this project. Without these actions, the other actions can hardly be executed. Moreover, it will not be possible for me to test some self-created instance to test the algorithm and grab the picture of where I am at. Therefore, I have chosen "setting the robots' action constraints" as my first steps to solve the problem. First, I need to set constrains on the movements and locations of the robot The following are the constrains I have set are:

1) Robots should not move out from the boundary of the grid.
2) Robots can only perform one action (or no action) at a time
3) Any one of the robots cannot be on multiple nodes. This restrict the robot to be present in one location only at any instance.
4) Multiple robots cannot be on same nodes. This avoid collision of the robots.
5) Robots' location cannot be swapped.

While explained in plain English, the effect of movement will be: The new location of the robot will be at (x+dx,y+dy) at the next time step (t+1), where (x,y) and t was the location and time point of the robot in the previous time step.

B. Setting constrains on the robots' pickup actions

After the robots are moving in according to the constrains set, including they would not collide with each other and does not move out of the grids, the next step is to define their pick up actions. I have to ensure the robots are picking up the shelf in a sensible way following the constrains given. The following are the constrains I have set for the pick up action of the robots:

1) A robot is not allowed to pickup a shelf if the shelf is already being picked up by the robot.
2) A robot is not allowed to pickup another shelf if it has already picked up a shelf at that moment.
3) Multiple robots are not allowed to pick up the same shelf at the same moment.

The effect of picking up a shelf will be: At the new time point (t+1), the shelf location (x,y) containing the product will be the location of the robot, which is picking up the shelf.

C. Setting constrains on the robots' putting down actions

The next step I did, is to define the actions , of which the robots putting down the shelves.The following are the constrains I have set for the putting down action:

1) A robot is not allowed to put down a shelf if it has not picked up any shelf at that moment.
2) Multiple robots are not allowed to put down the same shelf simultaneously
3) Robots are not allowed to put down shelves on the highway.

The effect of putting down a will be: At the new time point (t+1), the shelf location (x,y) will be the location of the robot, which is putting down the shelf.

D. Setting constrains on the robots' delivering actions

Next, I also defined the the actions , of which the robots delivering the products.The following are the constrains I have set:

1) Robots are only allowed to deliver the products if they are at the picking station.
2) Robots are only allowed to deliver if they have the shelf with the product.
3) The quantity of products that the robot delivers cannot be more than the number of products required in the order.
4) The quantity of products that the robot delivers cannot be more than the number of product on all shelves.

The effect of delivering will be: At a time point 't', the robot 'r' will deliver 'u' units of product 'i' within order 'o'. The location of the robot at the previous time point 't-1' would be the location of picking station of order 'o'. After delivering, the quantity of product 'i' on the shelf carried by the robot 'r' and the number of unit of product 'i' within order 'o' will both be reduced by 'u' at that time point 't'.

E. Setting constrains on shelves

After finishing setting the constrains on the perspective of the robot, I moved on to set the constrains on shelves. The following are the constrains I have set for the shelves:

1) A shelf cannot be on multiple robots' hands simultaneously.
2) There should not be multiple shelves on the same robot's hand.
3) There should not be multiple shelves at the same location (the same node).
4) A shelf can only be on a single location (one node).

F. Setting the states constrains

1) Picking station cannot present on the highway
2) The location of the shelves cannot be on the highway

G. Setting the law of inertia

After setting all the constrains, I have to set the laws of inertia, which are the common sense of the scenario. These includes:

1) If the robot did not move, the location of it will not change.
2) If the shelf was not being picked up, the location of it will not change.
3) If the shelf was not being put down by the robot, the location of it will be the same as the robot which is lifting it.
4) If the orders were not delivered, the quantity of the orders will remain unchanged.
5) If the products were not delivered, the quantity of the products will remain unchanged.

H. Setting the goal state of the whole task

Finally, I have to set the and to tell the algorithm goal state of the whole task. It will be the state to be reached so the robots can stop their actions. In plain words, the goal of the task should be: all the products in all orders are delivered to the designated picking stations respectively. If this apply, the robots can stop all their actions, including, moving, picking up, putting down and delivering. The optimal model is found by minimizing the time steps taken to achieve the goal.

III. Results

IV. Main results and analysis

The program was coded in Answer Set Programming and run by Clingo. I solved the problem step by step and the descriptions below shows the results and analysis in each of the steps.

1) **Setting constrains on the robots' movement.** My goal for this part is to have robot located at a given spot within a given shape of grid. Also, the robots should move within the grid and not colliding with each other. To test it I created an instance of 4x4 grids with highways and set the goal for the robots to move from there starting locations to locations of destination I have set. I was initially stuck at preventing the robot to move out of the grids.I checked my code again and found the mistake in defining the comparative constraints in limit of distance move by the robot to the row and column size and the involved. I finally be able to restrict the robot to move within the boundary. One of the ways for me to test this was to set the starting location of the robot at the corner of the grid and surround it by highway. If the constrains work, it should not give any stable models.
2) **Setting constrains on the robots' picking up and putting down shelves actions.** The goal for this part is to have the robots correctly pickup, and putting down the shelves at the right locations.I set few constrains so the robots perform the actions in a sensible way, e.g. not picking up two shelves at a time. The following segments of output is a demonstration of output related to this part of actions.

```
occurs(object(robot,2),pickup,0)
occurs(object(robot,1),pickup,2)
occurs(object(robot,2),pickup,6)
occurs(object(robot,1),pickup,7)
occurs(object(robot,2),putdown,4)
occurs(object(robot,1),putdown,5)
....
```

3) **Implementation of the robots' delivering.** The goal for this part is to have the robots correctly deliver the shelf with the product to the right spot. The number of product units should hold the right arithmetic in the whole process (i.e. the sum of units of a product 'i' included in orders does not exceed the sum of units of 'i' stores on shelves). The following segments of output is a demonstration of output related to this part of actions.

```
occurs(object(robot,2),deliver(2,2,1),3)
occurs(object(robot,1),deliver(1,1,1),4)
occurs(object(robot,1),deliver(1,3,4),9)
occurs(object(robot,2),deliver(3,4,1),9 ....
```

This is a very challenging step. There are multiple constrains involved, meanwhile, we should handle the arithmetic involved correctly.

I have encountered quite a lot of challenges throughout the process. For example, I was initially stuck at preventing the robot to move out of the grids. Also, my personal assumption was made that the constrains made to the robots were enough to make every actions sensible and able to fulfil the general constrains in the problem statement. However, I later on discovered that there are more constraints I should set to prevent situations like, multiple shelves at the same location and one shelf having multiple locations. I was also stuck at the part of delivering. I also spent a lot of time in making sure the sum of units of a product 'i' included in orders does not exceed the sum of units of 'i' stores on shelves.

instance	Number of models	Optimal model time steps	Number of actions
1	1	10	19
2	4	11	17
3	4	7	10
4	1	5	10
5	2	7	10

The above table shows my result of running the 5 instances. The optimal model is the one with minimal time steps taken to complete the task. The number of actions indicate the total number of actions taken by all the robots in the given instance.

V. Lessons learned

Although this project is challenging and made me struggle at some points, it served as a great practice for me to learn answer set programming and being able to apply it to a real life situation. Through the projects and assignments I have done earlier, I gained a profound picture and insight to answer set programming and knowledge-based reasoning. It is often easier to think in human beings' logic than making a machine/robot "logical". As I programmed the problem, I always missed some constrains. However, I also got the strategies of approaching the problem by breaking down the problem to several parts, writing the constrains in plain English, followed by turning the constrains to workable codes. My coding habit previously rely on trail-and-error, however, this seems so inefficient in this type of programming. There were too much case to consider and it is so easy for you to "miss" the error even if the code runs. I turned my approach to tackle the behaviors of the robot one by one, (e.g. movement, pickup, put down, deliver...) and would not proceed before making sure the previous stages were in correct orders.

Warehousing is a very practical and widely used scenario of Answer Set Programming (ASP). However, ASP can be also a very powerful tool in solving other complex real world problem. Such problems exists in wide variety of aspects, including commercial, space exploration, zoology, and bioinformatics..etc. I have listed some of the examples below:

1) Automated Product Configuration, which enables web-based product configuring [6].
2) Decision Support for Space Shuttle, which ASP would be capable of planning and diagnosing problems related to the operation of the Space Shuttle [6].
3) Inferring Phylogenetic Trees, which ASP can help in constructing a phylogeny in living organism and aid the studies in ecology and evolution systems [7].
4) Bio-informatics, the declarative nature of ASP allows modeling and solving some well-known and challenging

classes of problems in the general domain of bioinformatics, including genomics, structural and system studies. [8]

ASP has some distinctive advantages, including stable models and static—logic programming provides the level of elaboration-tolerance to support model modifications and the implementation of new knowledge. The use of traditional methods, such as linear programming, is often inadequate in such areas of applications. This is why ASP has a great potential in Artificial Intelligence and other domains.

REFERENCES

[1] K. Azadeh, R. De Koster, and D. Roy, "Robotized and automated warehouse systems: Review and recent developments," *Transportation Science*, vol. 53, no. 4, pp. 917–945, 2019.

[2] F.-l. CHANG, Z.-x. LIU, X. Zheng, and D.-d. LIU, "Research on order picking optimization problem of automated warehouse," *Systems Engineering-Theory & Practice*, vol. 27, no. 2, pp. 139–143, 2007.

[3] V. Lifschitz, *Answer set programming*. Springer Berlin, 2019.

[4] M. Gebser, R. Kaminski, and T. Schaub, "Complex optimization in answer set programming," *Theory and Practice of Logic Programming*, vol. 11, no. 4-5, pp. 821–839, 2011.

[5] T. Eiter, G. Ianni, and T. Krennwallner, "Answer set programming: A primer," in *Reasoning Web International Summer School*. Springer, 2009, pp. 40–110.

[6] G. Brewka, T. Eiter, and M. Truszczyński, "Answer set programming at a glance," *Communications of the ACM*, vol. 54, no. 12, pp. 92–103, 2011.

[7] G. Wu, J.-H. You, and G. Lin, "Quartet-based phylogeny reconstruction with answer set programming," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 1, pp. 139–152, 2007.

[8] A. Dal Palù, A. Dovier, A. Formisano, and E. Pontelli, "Exploring life: answer set programming in bioinformatics," in *Declarative logic programming: theory, systems, and applications*, 2018, pp. 359–412.