

Automated Warehouse Scenario

Gaurav Kumar

gkumar28@asu.edu

Problem Statement

An automated warehouse is a logistic facility where automated equipment's are used to optimize the operations. This project also demonstrates one of the ideas where robots are used to fulfill orders in timely manner. Whenever order is placed, robots which are flat in size carry shelves with the ordered product and take them to matching picking stations.

An Automated warehouse is represented as a rectangular grid where robots can move horizontally and vertically in adjacent cell. Note that we have a constraint that a robot cannot move diagonally in the warehouse. As mentioned above robots are flat in size and can move underneath shelves and pick them up. Another constraint here is when robot carries a shelf it cannot fit under another shelf until it moves away from the current shelf.

The goal is to fulfill all the orders in timely manner in addition to fulfilling all the constraints. Robots must be prevented from having collision in any state be it Idle, moving, pickup, put down, deliver etc. This also implies that robots cannot switch cells in any step. Steps here is counted in terms of time (t_1 , t_2 , t_3 etc.).

To fulfil the problem statement as described above next section will highlight project background and will detail out the strategies which helped in completing this project.

Project background

Automated warehouse is a good use case to solve in today's world. People are more inclined towards online shopping than traditional approach. Big companies maintain warehouses which stores product to be delivered to users across the globe. Automating these warehouses would be a scientific feat and would provide a cost-effective solution to the consumers in timely manner.

Knowledge representation and reasoning (KR&R) is a field of artificial intelligence (AI) concerned with how knowledge can be effectively structured, organized, and utilized to facilitate intelligent decision-making and problem-solving. It involves capturing and representing knowledge in a form that can be understood and processed by computational systems. Reasoning mechanisms are then applied to this knowledge to draw conclusions, make inferences, and solve complex problems.

As part of KRR course work I gained expertise in propositional logic, First order logic, Answer set programming, Reasoning about actions, Ontologies etc. For this project I am using answer set programming and clingo which has been discussed in week3 of the course. To complete this project week3, week4, week5 course has helped a lot. Many concepts like – choice rules, cardinality bounds, constraints, predicates, aggregates, 1-1 function, onto function etc. is used in the project.

Implementation of this project will revolve around the steps learnt in **Reasoning about actions and planning** which are as below.

- sorts and object declarations
- effect and precondition of action
- domain independent axioms
- fluent and actions are exogenous
- uniqueness and existence of fluent values

A fluent is something which depends on the state of the world. E.g.: Robot location

Approach in problem solving

To complete this challenging project, tasks have further been divided into subtask and to maintain readability some of the code is moved to a different file and is being used by including those files in main file. Project is organized in to following directories:

- **simpleInstances**: It contains initialization files which are in .asp format and are input to the system. There

are in total 5 input files – inst1.asp, inst2.asp, inst3.asp, inst4.asp, inst5.asp.

- **FormatInput.asp**: This file format the input file from above directory to more readable and understandable format which will be used to implement constraints in project. Eg:

```
nodeAt(NDI,pair(X,Y)):init(object(node,
NDI),value(at,pair(X,Y))).
pair(X,Y):init(object(node,
NDI),value(at,pair(X,Y))).
```
- **RetrieveProduct.asp**: This file is responsible for returning values of number of objects and type.
Eg: % Number of Nodes

```
numNodes(ND):-
ND=#count{I:init(object(node,I),value(at,pair(X,Y)
))}.% Number of Products
numProducts(ND):-
ND=#count{I:init(object(product,I),value(on,pair(
X,Y)))}.
% Number of Robots
numRobots(ND):-
ND=#count{I:init(object(robot,I),value(at,pair(X,Y)
))}).
```
- **main.asp**: This is main file where all the code is written.
- **Description.pdf**: This file contains the problem statement and provide detail description of the system.
- **main.py**: Python file to initiate the execution of the program.

With the above structure, project is easier to maintain. As mentioned **main.asp** is the file where all the logic, constraint, choice rules etc. are declared. Let's look in to detail of the **main.asp** to have the clear understanding towards approach of the problem.

main.asp file first need to include the other two files which are used to format input file and get the number of objects. This can be done using include directive. A constant **n** is also declared for number of steps. Next we need to look into the actions like – Robot movement, pick Shelf, put shelf, deliver product. These actions can be declared as below:

```
move (0,1;0,-1;-1,0;1,0).
{robotMove(R,move(DX,DY),T):move(DX,DY)}1:-
R=1..NR, numRobots(NR), T=0..TN,TN=n-1.
{pickUpShelf(R,SI,T):shelf(SI)}1:- R=1..NR, num-
Robots(NR), T=0..TN,TN=n-1.
{putDownShelf(R,SI,T):shelf(SI)}1:- R=1..NR, num-
Robots(NR), T=0..TN,TN=n-1.
{deliv-
er(R,OI,with(SI,PR,DQ),T):orderAt(OI,object(node,N
D),contains(PR,OQ),T), produc-
tOn(PR,object(shelf,SI),with(quantity,PQ),T),
```

```
DQ=1..PQ}1:- R=1..NR, numRobots(NR),
T=0..TN,TN=n-1.
```

All of the above actions have certain constraint as per the project guidelines. We can look into some of those. For the robot movement: % **Robot cannot move outside of the grid**

```
:-robotAt(RI,object(node,ND),T), robot-
Move(R,move(DX,DY),T),nodeAt(ND,pair(X,Y)),
X+DX<1.
```

Similar to above we need to declare constraints for other actions. When constraints for all the actions are declared next step would be to think of state constraints. We have multiples state constraint for robot and the Shelf for different scenarios like- No two robots can be on same node, no two shelves on same robot, shelf and picking station cannot be highway etc. E.g.: % **No robot on 2 nodes**

```
:-2{robotAt(R,object(node,ND),T):node(ND)}, ro-
bot(R), T=0..n.
```

Next step would be declaring effects for the actions like- Effect of robot movement, robot picking up a shelf, robot putting down a shelf, delivering a product. E.g.: % **Effect of moving robot.**

```
robotAt(R,object(node,NEW_ND),T+1):- ro-
botAt(R,object(node,ND),T), nodeAt(ND,pair(X,Y)),
nodeAt(NEW_ND, pair(X+DX,Y+DY)), robot-
Move(R,move(DX,DY),T).
```

Next steps would be to provide law of inertia and goal state. As per the problem statement we have to fulfill the orders in minimum amount of time and steps. This can be achieved as following: **numActions(N):-**

```
N=#sum{1,O,A,T:occurs(O,A,T)}.
timeTaken(N-1):-N=#count{T:occurs(O,A,T)}.
#minimize{1,O,A,T:occurs(O,A,T)}.
#minimize{T:occurs(O,A,T)}.
```

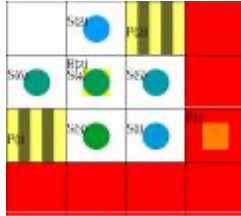
Finally, we have to declare the multiple show directive to see the result in output. E.g.: **#show occurs/3.**

Main Result and Analysis

There are five input files given in the project for which we have to find optimal result. Below is the summarized detail of information present in input file: **inst1.asp**

```
% location of node:
init(object(node,1),value(at,pair(1,1))).
% location of highways:
init(object(highway,4),value(at,pair(4,1))).
% picking station
init(object(pickingStation,1),value(at,pair(1,3)))
%info about number of robots and their location
init(object(robot,1),value(at,pair(4,3))).
%info about shelf locations
init(object(shelf,1),value(at,pair(3,3))).
%info about product shelf location and quantity
```

Inst1.asp is a 4*4 grid scenario with two robots, six shelves and three orders. The diagram below taken from description.pdf depicts the initial scenarios for inst1.asp file.



As a python script is used to initiate the program which eventually calls the clingo and main.asp to execute the asp code. The following program runs for the value of n=5. Here is how it looks at the terminal:

Similarly, we can execute the program for other input files and retrieve the output. For **inst2.asp** following is the output.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[+] 729x +[] +[] +[] +[] +[] +[]
gauravkumar@Gauravs-MBP Automated-Warehousing-master % python main.py

code Instance File
1: --:
1 inst1.asp
2 inst2.asp
3 inst3.asp
4 inst4.asp
5 inst5.asp

Please enter the code for the instance file: 2
cingo version 5.6.2
Reading from main.asp ...
Solving...
Answer: 1
occurs(object(robot,1),move(0,-1,0),0) occurs(object(robot,2),move(0,1,0),0) occurs(object(robot,1),move(-1,0,1),0) occurs(object(robot,2),move(-1,0,2),0) occurs(object(robot,1),move(0,-1,4),0) occurs(object(robot,1),move(0,1,5),0) occurs(object(robot,2),move(1,0,5),0) occurs(object(robot,1),move(0,1,6),0) occurs(object(robot,2),move(0,-1,6),0) occurs(object(robot,1),move(-1,0,7),0) occurs(object(robot,2),move(-1,0,8),0) occurs(object(robot,2),move(0,-1,8),0) occurs(object(robot,2),move(1,0,9),0) occurs(object(robot,2),pickup,1) occurs(object(robot,1),pickup,3) occurs(object(robot,2),pickup,7) occurs(object(robot,2),putdown,4) occurs(object(robot,2),deliver(1,1,1),3) occurs(object(robot,1),deliver(1,3,1),10) occurs(object(robot,2),deliver(2,2,1),10) occurs(object(robot,1),deliver(1,1,1),9) timeTaken(10) numActions(21)
Optimization: 76
Answer: 2
occurs(object(robot,1),move(0,-1,0),0) occurs(object(robot,2),move(0,1,0),0) occurs(object(robot,1),move(-1,0,1),0) occurs(object(robot,2),move(-1,0,2),0) occurs(object(robot,1),move(0,-1,4),0) occurs(object(robot,1),move(0,1,5),0) occurs(object(robot,2),move(0,1,5),0) occurs(object(robot,1),move(0,1,6),0) occurs(object(robot,2),move(0,-1,6),0) occurs(object(robot,1),move(-1,0,7),0) occurs(object(robot,2),move(-1,0,8),0) occurs(object(robot,2),move(0,-1,8),0) occurs(object(robot,2),move(1,0,9),0) occurs(object(robot,2),pickup,1) occurs(object(robot,1),pickup,3) occurs(object(robot,2),pickup,7) occurs(object(robot,2),putdown,4) occurs(object(robot,2),deliver(1,1,1),3) occurs(object(robot,2),deliver(1,3,2),10) occurs(object(robot,2),deliver(2,2,1),10) timeTaken(10) numActions(20)
Optimization: 75
Answer: 3
occurs(object(robot,1),move(-1,0,0),0) occurs(object(robot,2),move(0,1,0),0) occurs(object(robot,1),move(0,-1,1),0) occurs(object(robot,2),move(-1,0,2),0) occurs(object(robot,1),move(0,1,3),0) occurs(object(robot,2),move(0,-1,5),0) occurs(object(robot,2),move(1,0,6),0) occurs(object(robot,1),move(-1,0,7),0) occurs(object(robot,2),move(0,-1,8),0) occurs(object(robot,1),move(-1,0,9),0) occurs(object(robot,2),pickup,1) occurs(object(robot,2),pickup,7) occurs(object(robot,2),putdown,4) occurs(object(robot,1),pickup,2) occurs(object(robot,2),pickup,6) occurs(object(robot,2),deliver(1,1,1),3) occurs(object(robot,2),deliver(1,3,2),10) occurs(object(robot,2),deliver(2,2,1),10) timeTaken(10) numActions(18)
Optimization: 73
Answer: 4
occurs(object(robot,1),move(-1,0,0),0) occurs(object(robot,1),move(-1,0,1),0) occurs(object(robot,2),move(1,0,1),0) occurs(object(robot,2),move(0,-1,2),0) occurs(object(robot,1),move(-1,0,3),0) occurs(object(robot,1),move(1,0,5),0) occurs(object(robot,2),move(0,1,5),0) occurs(object(robot,2),move(-1,0,7),0) occurs(object(robot,2),move(-1,0,8),0) occurs(object(robot,2),move(0,1,9),0) occurs(object(robot,2),pickup,0) occurs(object(robot,1),pickup,2) occurs(object(robot,2),pickup,6) occurs(object(robot,2),putdown,4) occurs(object(robot,2),deliver(2,2,1),3) occurs(object(robot,1),deliver(1,1,1),4) occurs(object(robot,2),deliver(1,3,2),10) timeTaken(10) numActions(17)
Optimization: 72
OPTIMUM FOUND

Models: 4
Optimum: yes
Optimization: 72

```

For **inst3.asp** following output is produced:

```
PROBLEMS          OUTPUT          DEBUG CONSOLE          TERMINAL          zsh + ~ ▣ ▢ ▹ ▸ ▹ ▸ x
```

```
gauravkumar@Gauravs-MBP Automated-Warehousing-master % python main.py
```

code	Instance File
1	inst1.asp
2	inst2.asp
3	inst3.asp
4	inst4.asp
5	inst5.asp

Please enter the code for the instance file: 3
clingo version 5.6.2
Reading from main.asp ...
Solving...
Answer: 1
occurs(object(robot,1),move(-1,0),0) occurs(object(robot,1),move(0,-1),1) occurs(object(robot,2),move(-1,0),2) occurs(object(robot,1),move(0,-1),3) occurs(object(robot,2),move(0,-1),3) occurs(object(robot,2),move(1,0),4) occurs(object(robot,1),move(0,1),5) occurs(object(robot,2),move(1,0),5) occurs(object(robot,2),pickup,1) occurs(object(robot,1),pickup,2) occurs(object(robot,1),deliver(2,4,1),4) occurs(object(robot,2),deliver(1,2,1),6) timeTaken(6) numActions(12)
Optimization: 33
Answer: 2
occurs(object(robot,1),move(0,-1),0) occurs(object(robot,1),move(-1,0),1) occurs(object(robot,1),move(0,-1),3) occurs(object(robot,2),move(0,-1),3) occurs(object(robot,1),move(0,1),5) occurs(object(robot,2),move(1,0),5) occurs(object(robot,2),pickup,1) occurs(object(robot,1),pickup,2) occurs(object(robot,1),deliver(2,4,1),4) occurs(object(robot,2),deliver(1,2,1),6) timeTaken(6) numActions(10)
Optimization: 31
OPTIMUM FOUND
Models : 2
Optimum : yes
Optimization : 31
Calls : 1
Time : 0.060s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.01s)
CPU Time : 0.060s

```
gauravkumar@Gauravs-MBP Automated-Warehousing-master %
```

Conclusion

For **inst4.asp** following is the optimal output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
gauravkumar@Gauravs-MBP Automated-Warehousing-master % python main.py

code | Instance File |
:---:|:-----:|
1 | inst1.asp |
2 | inst2.asp |
3 | inst3.asp |
4 | inst4.asp |
5 | inst5.asp |

Please enter the code for the instance file: 4
clingo version 5.6.2
Reading from main.asp ...
Solving...
Answer: 1
occurs(object(robot,1),move(-1,0),0) occurs(object(robot,1),move(-1,0),1) occurs(object(robot,2),move(0,-1),1) occurs(object(robot,2),move(1,0),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 : 1
Optimum : yes
Optimization : 20
Calls : 1
Time : 0.081s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.072s

gauravkumar@Gauravs-MBP Automated-Warehousing-master %
```

For **inst5.asp** following is the optimal output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
gauravkumar@Gauravs-MBP Automated-Warehousing-master % python main.py

code | Instance File |
:---:|:-----:|
1 | inst1.asp |
2 | inst2.asp |
3 | inst3.asp |
4 | inst4.asp |
5 | inst5.asp |

Please enter the code for the instance file: 5
clingo version 5.6.2
Reading from main.asp ...
Solving...
Answer: 1
occurs(object(robot,1),move(-1,0),0) occurs(object(robot,2),move(0,1),0) occurs(object(robot,1),move(0,-1),1) occurs(object(robot,1),move(-1,0),2) occurs(object(robot,1),move(-1,0),3) occurs(object(robot,2),move(-1,0),3) occurs(object(robot,1),move(0,1),5) occurs(object(robot,2),move(0,1),5) occurs(object(robot,2),pickup,2) occurs(object(robot,1),pickup,4) occurs(object(robot,2),deliver(1,1,1),4) occurs(object(robot,1),deliver(1,3,4),6) timeTaken(6) numActions(12)
Optimization: 33
OPTIMUM FOUND

Answer: 2
occurs(object(robot,1),move(-1,0),0) occurs(object(robot,1),move(-1,0),1) occurs(object(robot,1),move(-1,0),3) occurs(object(robot,2),move(-1,0),3) occurs(object(robot,1),move(1,0),5) occurs(object(robot,2),move(0,1),5) occurs(object(robot,1),pickup,2) occurs(object(robot,2),pickup,4) occurs(object(robot,1),deliver(1,1,1),4) occurs(object(robot,2),deliver(1,3,4),6) timeTaken(6) numActions(10)
Optimization: 31
OPTIMUM FOUND

Models : 2
Optimum : yes
Optimization : 31
Calls : 1
Time : 0.082s (Solving: 0.02s 1st Model: 0.00s Unsat: 0.01s)
CPU Time : 0.081s

gauravkumar@Gauravs-MBP Automated-Warehousing-master %
```

It was a great experience implementing something learnt throughout the course. The project was challenging initially but revisiting some of the core concept helped in understanding and implementing the project. Doing this project also helped in knowing the potential of KRR and how it can be utilized in solving real world problems. Potential of KRR is unlimited and there would definitely be advanced research happening in future. I am enthusiastic to apply the concepts learnt in this course and project in future KRR projects.

Opportunities for future work

There are many real-world scenarios where KRR stands out and can be used to tackle the problems. Some of the fields where KRR can be explored in future are as below:

- **Explainable AI and Reasoning:** As AI systems become more sophisticated, there is an increasing demand for explainable AI, particularly in critical domains like healthcare and finance. Opportunities exist for researchers and practitioners to work on developing methods and tools for explainable reasoning in AI systems, ensuring transparency and accountability.
- **Automated Reasoning and Theorem Proving:** Automated reasoning and theorem proving are central to KRR, facilitating logical inference and knowledge validation. Future work may involve advancing the state-of-the-art in automated reasoning techniques, developing more efficient algorithms, and exploring new reasoning paradigms.
- **Contextual and Temporal Reasoning:** Enhancing the ability of KRR systems to reason with context and temporal information is another promising area. Work opportunities may involve developing models and techniques to incorporate context-aware and temporal reasoning capabilities into KRR systems.

References

- ASU CSE579: Knowledge Representation and Reasoning: week3, week4 and week5
- <https://github.com/yashvakil/Automated-Warehousing>
- https://web.stanford.edu/~vinayc/logicprogramming/html/answer_set_programming.html
- https://eur-ws.org/Vol-804/04_LANMR11.pdf
- <https://www.mdpi.com/1999-4893/16/4/216>