DOI: 10.53469/jrse.2025.07(10).02

Leveraging Answer Set Programming for Warehouse Automation: A Fundamental Approach to Enhancing Supply Chain Efficiency

Varadharaj Varadhan Krishnan

Olathe, USA
kirshnan@gmail.com
https://www.linkedin.com/in/bhubalanmani/
https://orcid.org/0009-0006-1143-9360

Abstract: ***With growth in customer demand and market competition it is more important than ever to improve the speed and accuracy of all interconnected functions in the supply chain[1]. In every domain involving supply chain, effective storage and retrieval of products in a warehouse in critical to the success of overall supply chain. Effective warehouse automation not only improves overall supply chain efficacy, but it also reduces various costs like warehouse operations cost, inventory holding cost, cost incurred due to damaged product. With advancement in technology, we have seen warehouse operations evolve from manual handling to multiple sophisticated robots working simultaneously to complete the work[1]. Irrespective of advancement, the basic requirement from automation remains same. In this paper, I try to present a basic automation of a warehouse which would address the basic requirements from an automation like safe handling of product and quicker fulfillment of order. Any sophisticated system can be built upon this fundamental building blocks.***

Keywords: Warehouse Automation, Answer Set Programming, Supply Chain Efficiency, Optimization, Robotics, Logistics, Supply Chain Analytics

1. Introduction

This paper develops an effective approach to automation system to fulfill orders in minimum possible duration. Multiple real world challenges and requirements were considered in this study. Specifically, which involves handling high value finished goods that needs to be shipped to customer. In addition to the fundamental asks from an automated system like speed, accuracy and increased volume, any automation should make sure that the design is free from certain conflicting actions which would either result in damage to the products or create safety concerns or damage the system itself. The goal is to develop a working model of an automated material handling system in a finished goods warehouse which will fulfill the order in the minimum possible time[5]. In additional to meeting the basic requirement from the design, this paper also makes sure that none of the movements cause any damage to the product or the system itself. The goal would be to effectively utilize the automation for effective order promising without any issues in the possible minimum time. Key conflicting actions that were avoided were inability to move underneath the shelves while carrying load, inability to drop shelves on highways, or robots swapping positions at same time. This problem is particularly interesting to me as I've spent close to 20 years in Supply Chain area and my employer recently built a new automated warehouse dedicated for a new major automobile manufacturing client.

My Supply Chain Management background specifically in planning helped me to easily interpret the requirement and articulate the problem well relating to real world working model. I was able to quickly come up with required state and action constraints to have an effective working model. Before getting into advanced topics, this study starts off with basic foundational concepts like functions, constraints, optimization, reasoning about actions using examples like Hamiltonian Cycle, Ancestors problem, Blocks worlds examples, Monkeys and Bananas, Sudoku and chess. I've used Asprilo visualizer[7] to visualize actions and troubleshoot the violation of constraints. The tool was very handy and saved lot of time by quickly identifying the actions that logically shouldn't happen in real world. Further, I've referenced the online material and books that were helpful in completing this study.

2. Description of Solution

The paper approaches this section in two parts. First is the approach to the solution, followed by the actual solution.

a) Approach to Solving the Problem

My approach to get to the final working model was to start with building the movements and actions and then add constraints as required. The thought process was to have a working model that would fulfil the orders without any constraints then start adding individual constrains[2].

I broke down the action into 3 main categories as listed below:

- Moving the robot to instructed final grid. With this I was able to focus on 'move' action of the robot and get the robot to move from initial position to the picking stations.
- Moving the robot loaded with shelf containing the product from initial position to the picking station. With this I was able to focus on 'pickup' and 'putdown' actions on the robot.
- Delivering the product from shelf to fulfill the order at the picking station. With this I was able to focus on making sure the quantity gets decrements from shelf and the out-standing quantity on order is also reduced as deliveries are made.

Once I was able to make the robot to make all the actions as listed above for all provided 5 instances without any constraints, I started to build constraints for each of the action[3]. I've listed few of the constraints below. Few of the listed constraints were built to stop movements that did not make sense while troubleshooting with the visualizer[4].

For 'move' action, below are few of the constraints I modelled.

- Robots cannot move outside the grid.
- Robots cannot swap places.
- A loaded robot cannot move underneath the shelf.
- Cannot enter the picking stations when not loaded.

For 'pickup' and 'drop' action, below are few of the constraints I modelled.

- A robot cannot carry 2 shelves.
- A robot can pick up shelf only when at the location of the shelf.
- Robots cannot drop the shelves on highway.
- Cannot enter the picking stations when not loaded.

For 'deliver' action, below are few of the constraints I modelled.

- A robot can deliver only at the picking station.
- A robot cannot deliver more than required.
- A robot cannot deliver more than available.
- A robot cannot deliver a wrong product.
- A robot can deliver only from the shelf that is on the robot.

In addition to this, I modelled state constraints and few constraints on combination of actions. Below are few constraints that were modelled.

- A shelf cannot be on the highway.
- 2 robots cannot be at the same location.
- 2 shelves cannot be at the same location.

Further, I logically segregated the constraints as mandatory and flexible constraints based on the impacts of violating the constraints. Flexible constraints are the ones which can be tweaked or relaxed based on other changes. I've given more details on this in the results and analysis section.

Once I had all the pieces, the overall broader scope of the study was broken down to specific subject areas to approach this problem.

- Sort and object declaration.
- State constraints.
- Effect and preconditions of actions.
- Action constraints.
- Domain independent axioms with subsections fluents are initially exogenous, uniqueness and existence of fluent values, actions are exogenous and commonsense laws of inertia.

I enabled these constraints iteratively in a logical manner and used asprilo visualizer to make sure the actions and movements logically make sense and doesn't cause any violation.

Finally when I had a working model that works for all test cases, I tweaked few of the constraints to simulate few scenarios based on real world questions. I've given more details on this with additional details in the following section.

b) *Main Results and Analysis*

I will be presenting analysis for one of the simpler test cases including the interpretation of results and simulation done by tweaking the constraints. I'll also be discussing how the response of model to changes in constraints plays into the real-world decision process.

I've presented the results of all other test cases at the end of the section.

I've taken instance 5 for illustration, simulation and detailed analysis. I've presented 2 simulated cases below which are results of relaxation made to the constraints based on the improvements done to the working environment. This case needs fulfilment of just 1 order with mix of different products required.

I'll be using illustration from asprilo visualizer to demonstrate the model. Below are few of the legends that will be helpful in visualizing the model.

Below figure shows the initial state of the warehouse including the position of robots, shelves and location of picking station. The grids are numbered from R1 through R4 for rows and C1 through C4 for columns. At any given time robots can move one grid up or down in the same vertical axis or move one grid right of left in the same horizontal axis. The robots can move only one step at a time and can perform only one action at the time. Cannot move and pickup, putdown or deliver at the same time. A loaded robot cannot navigate through the grid which is occupied by a shelf.

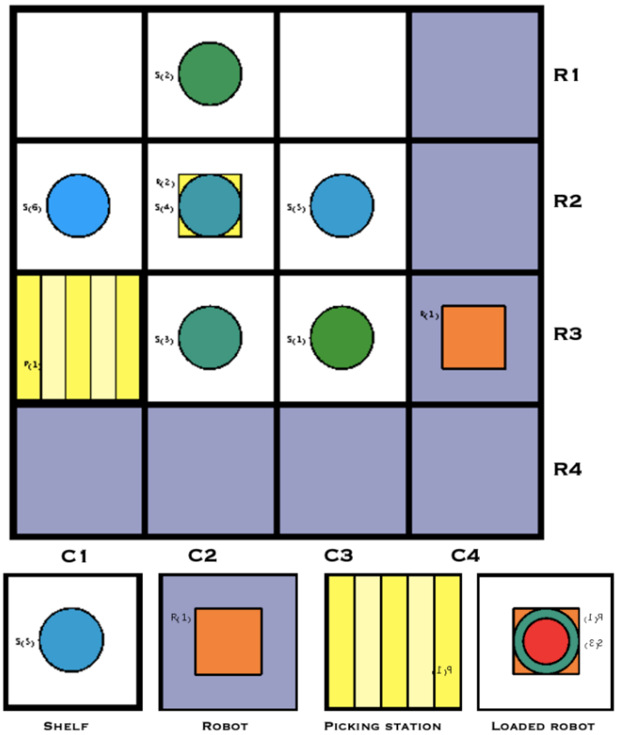


Figure 1: Initial State and Legend

Case-1 is to simulate a scenario where the firm has made investments in improving the working environment. The firm has built a sophisticated picking station which can accommodate multiple robots to make deliveries at the same time. This simulation was achieved by relaxing the state constraint which was restricting more than 1 robot at the picking station.

I've taken a simple case for illustration, simulation and detailed analysis. I've presented 2 simulated cases below which are results of relaxation made to the constraints based on the improvements done to the working environment. This case needs fulfilment of just 1 order with mix of different products required.

A key takeaway from this simulation is how infrastructure enhancements directly impact process efficiency. By allowing multiple robots to make deliveries simultaneously, the model eliminates unnecessary waiting times and optimally utilizes the available workforce. The ability to deploy multiple robots at the picking station significantly reduces idle time and ensures that fulfillment happens in a synchronized manner. As seen in the simulation, this results in a more distributed workload, where each robot is engaged efficiently, preventing bottlenecks that typically arise from restrictive movement constraints.

Additionally, this scenario highlights the importance of scalability in automation-driven environments. By relaxing the constraint on the picking station, the system dynamically adapts to increased order volumes without requiring fundamental changes to robot behavior. This reinforces the principle that small yet strategic infrastructure upgrades can drive exponential improvements in throughput. However, firms must also analyze whether allowing multiple robots at a single station introduces operational risks, such as congestion or coordination challenges, and whether an intelligent scheduling mechanism is required to maximize the benefits of this investment. These simulations provide critical insights that decision-makers can use to evaluate whether parallelizing order fulfillment aligns with their long-term operational goals.

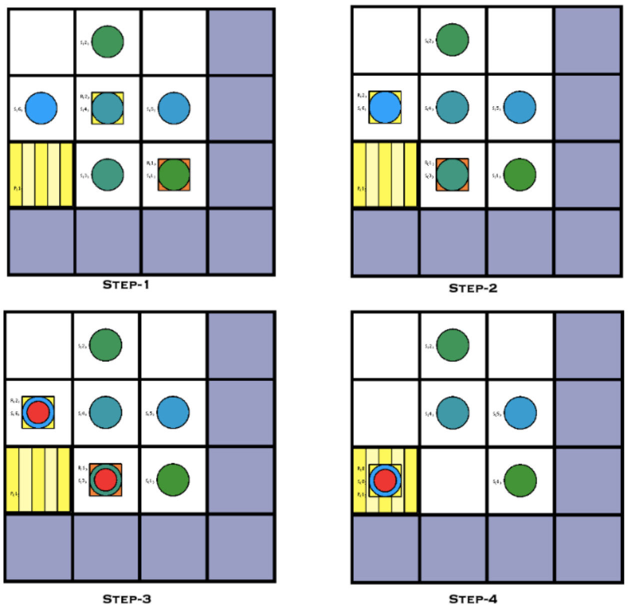


Figure 2: Simulation-1

Case-2 is to simulate a scenario where the firm has made investments in improving the technology and installing a more sophisticated robot which can carry 2 shelves at a time. This simulation was achieved by relaxing the constraint which was restricting robot from carrying more than 1 shelf at a time. Interesting observation form this is, the optimized model chose to use just 1 robot instead of using both the robots. It was able to fulfil the order in 6 timesteps and 8 total actions. Even though there is an increase in timesteps, total number of actions come down. Also, a single robot is able to complete the order with not doubling up the timesteps.

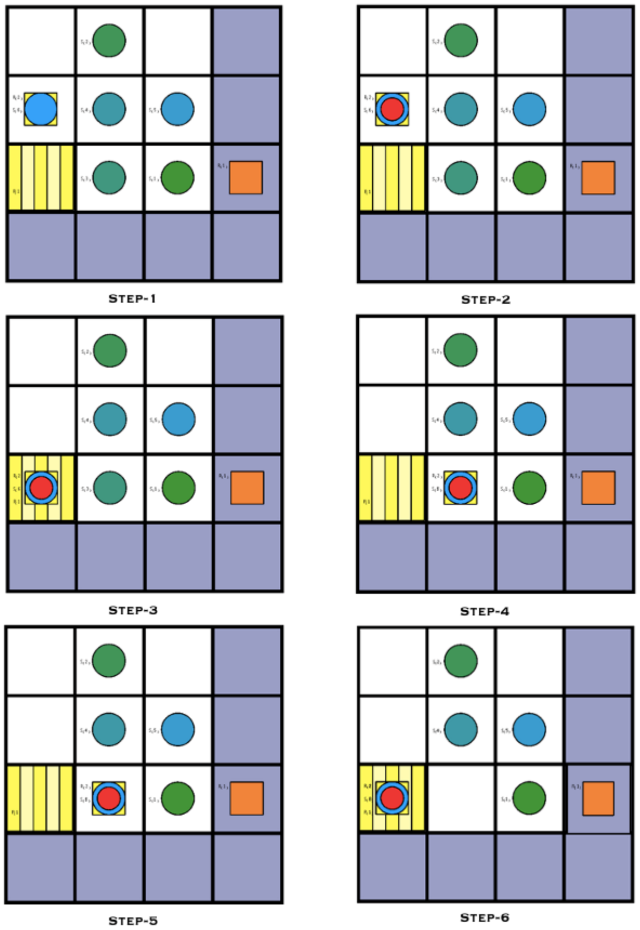


Figure 3: Simulation-2.

These simulations give powerful information which can be interpreted to arrive at important business decisions based on the requirement and constraints. How do we achieve the minimum timestep when your takt time to fulfil orders based on demand is fixed or how do you bring down your investment and operating cost by still meeting the order fulfilment goals?.

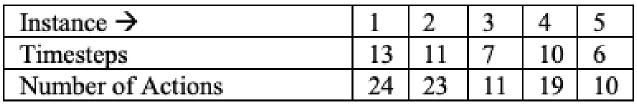
Another crucial insight from this simulation is the trade-off between robot utilization and efficiency. By enabling a more advanced robot capable of handling two shelves at a time, the system naturally gravitates towards using a single robot rather than distributing tasks between multiple units. This highlights the self-optimizing nature of constraint-based decision models when given the flexibility, the model inherently finds the most efficient way to fulfill orders with the least effort. The reduction in total actions, despite a slight increase in timesteps, reinforces the principle that task consolidation can often be a more effective strategy than parallel execution, particularly in structured environments like warehouse automation.

Additionally, this scenario underscores the importance of balancing technology investments with operational goals.

While a single robot approach has clear advantages in terms of reducing total movements, firms must also assess whether the increased takt time aligns with service level agreements (SLAs) and customer expectations. If demand spikes unpredictably, relying solely on one high-capacity robot might introduce bottlenecks. Therefore, a hybrid approach where the system dynamically decides when to activate a second robot based on demand variability—could be a more adaptive and scalable solution. This simulation presents a strategic decision point for firms aiming to optimize between technology-driven efficiency and operational responsiveness.

3. Results

The results show that the total moves and time increases with the increase in number of orders to be fulfilled. The mix of products for each model also has an impact on the total duration to fulfill the order. But the mix of products has higher impact on the duration and number of steps.



Instance →	1	2	3	4	5
Timesteps	13	11	7	10	6
Number of Actions	24	23	11	19	10

Figure 4: Results

4. Conclusion

Declarative Knowledge Representation and Reasoning (KRR) frameworks [6] have demonstrated their significant utility in tackling complex search problems such as design, configuration, planning, classification, and diagnosis. Among these, Answer Set Programming (ASP) stands out due to its expressive representation language and its ability to efficiently model and solve intricate problems. As a result, ASP has gained widespread recognition in both academia and industry, with its adoption steadily increasing in response to the growing demand for advanced problem-solving technique.

Despite its strengths, ASP still faces challenges, particularly in terms of community support and accessible knowledge resources. As technology advances and computational capabilities continue to evolve, I foresee ASP gaining even greater traction, benefiting from improved toolsets and a more robust ecosystem. However, for ASP to fully realize its potential, there is a need for stronger industry adoption, better documentation, and a more engaged support community.

From a personal perspective, this project has provided invaluable insights into the power of ASP in warehouse automation and supply chain optimization. While I encountered challenges due to the limited knowledge base, I also recognized the vast untapped potential of this approach. Moving forward, I am committed to deepening my expertise in ASP, contributing to its evolving ecosystem, and applying it to increasingly complex and impactful real-world problems. With continuous advancements in technology and growing industry interest, the future of ASP looks promising and I am excited to be part of its journey.

Declarative Knowledge Representation and Reasoning (KRR learning, will open new avenues to tackle even more sophisticated problems. Cross-disciplinary applications are the key to unlocking the full potential of ASP in various industries, from logistics to healthcare. As the demand for more intelligent and adaptive solutions rises, I am confident that ASP will continue to be at the forefront of problem-solving innovation.

References

[1] Falkner, A., Friedrich, G., Schekotihin, K. et al. Industrial Applications of Answer Set Programming. Künstl Intell 32, 165–176 (2018). https://doi.org/10.1007/s13218-018-0548-6.

[2] Erik T. Mueller,Chapter 15 - Commonsense Reasoning Using Answer Set Programming, Editor(s): Erik T. Mueller,Commonsense Reasoning (Second Edition),Morgan Kaufmann,2015,Pages 249-269,ISBN 9780128014165.

[3] Baral, Chitta (2003). Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press. ISBN 978-0-521-81802-5.

[4] Subrahmanian, V.S.; Zaniolo, C. (1995). "Relating stable models and AI planning domains". In Sterling, Leon (ed.). Logic Programming: Proceedings of the Twelfth International Conference on Logic Programming. MIT Press. pp. 233–247. ISBN 978-0-262-69177-2.

[5] Gebser, M., Obermeier, P., Otto, T., Schaub, T., Sabuncu, O., Nguyen, V., & Son, T. C. (2018). Experimenting with robotic intra-logistics domains. CoRR, abs/1804.10247.

[6] Andres, B. Kaufmann, O. Matheis, and T. Schaub. Unsatisfiability-based optimization in clasp. In A. Dovier and V. Santos Costa, editors, Technical Communications of the Twenty-eighth International Conference on Logic Programming (ICLP'12), volume 17, pages 212–221. Leibniz International Proceedings in Informatics (LIPIcs), 2012. 75, 94.

[7] Banbara, B. Kaufmann, M. Ostrowski, and T. Schaub. Clingcon: The next generation. Theory and Practice of Logic Programming, 17(4):408–461, 2017.58.

[8] Biere, M. Heule, H. van Maaren, and T. Walsh, editors. Handbook of Satisfiability, volume 185 of Frontiers in Artificial Intelligence and Applications. IOS Press, 2009. 9, 60

[9] Fandinno, J., Mishra, S., Romero, J., & Schaub, T. (2023). Answer Set Programming Made Easy. In Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems (Vol. 13160, pp. 133–150). Springer

[10] Takeuchi, R., Banbara, M., Tamura, N., & Schaub, T. (2023). Solving Vehicle Equipment Specification Problems with Answer Set Programming. In PADL (Vol. 13880, pp. 232–249). Springer

[11] Son, T. C., Pontelli, E., Balduccini, M., & Schaub, T. (2023). Answer Set Planning: A Survey. Theory Pract. Log. Program., 23(1), 226–298

[12] Cabalar, P., Dieguez Martin, Schaub, T., & Schuhmann, A. (2022). Metric Temporal Answer Set Programming over Timed Traces. In LPNMR (Vol. 13416, pp. 117–130). Springer.