Automated Warehouse

Kavya Chandrika Vempalli
kvempall@asu.edu

Abstract*—This project aims to optimize warehouse functionalities by introducing a system that empowers autonomous robots to transport goods from designated storage positions to assigned picking stations, aiming to achieve efficient order fulfillment using robots within the shortest feasible timeframe. The warehouse structure is illustrated as a rectangular grid where each grid point, termed as a node, is assigned a unique identifier. The warehouse comprises shelves, each uniquely numbered, serving as storage locations for diverse products. These shelves can occupy any node within the warehouse, except for designated highway nodes, each possessing its distinct identifier. The robots, characterized by a flat design, have the ability to move horizontally or vertically when not carrying a shelf and have the agility to navigate beneath shelves. However, once a robot lifts a shelf, it must traverse to an adjacent unoccupied node, forfeiting the capability to maneuver beneath other shelves. In scenarios where an adjacent empty node is unavailable, the robot must attempt to clear a path for its movement. Ensuring collision-free navigation of robots while simultaneously traversing the warehouse remains a critical aspect of this endeavor.

***Keywords*— orders, node, highway, pickingStation, robot, shelf, product**

I. Introduction

In the automated warehouse system, the primary task assigned to robots is the transportation of products to a specific picking stations for fulfilling customer orders efficiently. The layout of the warehouse is visualized as a rectangular space, and the robots are programmed such that they can navigate through neighbouring cells by moving either horizontally or vertically. Their primary responsibility involves carrying shelves that contain the necessary products to the designated picking stations.

Robots possess the capability to move underneath shelves and lift them as required. However, a robot carrying a shelf is unable to maneuver beneath another shelf. This restriction may lead to the need for rearranging shelves to create a clear path for the moving robots. The Key objective is to ensure the efficient completion of all orders, with time measured in discrete steps. Each robot is capable of performing only one action per step, which could involve shifting, lifting, delivering a shelf, or remaining idle.

Throughout the robots' movements within the warehouse, which include actions like shifting, lifting, delivering items, or remaining stationary, a stringent rule governs their behaviour: collisions between robots are strictly prohibited. This regulation ensures that no two robots can occupy the same cell or interchange positions in consecutive steps.

Solving the challenges presented by this complex scenario involves developing strategic plans for robots movements and shelf placements. This strategy aims to successfully fulfil orders while adhering to the strict rules and constraints imposed within the warehouse environment. It requires meticulous planning to orchestrate the actions of the robots and the placement of shelves in a manner that optimizes efficiency while preventing any potential collisions between the robots.

II. Description of Solution

The development of an automates warehouse system using Answer Set Programming (ASP) with CLINGO aimed to simulate a realistic environment for efficient order fulfilment. This section delves into the specifics of the project's approach, the stages of implementation, results and the challenges encountered. The project began by establishing foundational rules governing the warehouse layout, particularly the grid structure and designated highways. In this simulated environment, robots were programmed to navigate across the grid, allowing movements along both the X and Y axes. The critical objective was to facilitate the movement of shelves by the robots to specific picking stations. Rules governing shelf pickup, movement, and delivery were meticulously crafted to simulate the real-world constraints observed in warehouses.

The completion of tasks involved defining various actions for robots, including movement, lifting shelves, delivering items to picking stations, and waiting. Each action was restricted to occur within discrete time steps, mimicking the limitations faced by robots in an actual warehouse setting. The project's implementation was demonstrated through sample scenarios that encompassed different robot actions, shelf movements, and order fulfilments. Testcases were meticulously crafted to cover a spectrum of actions, ensuring the adherence of the system to predefined constraints and rules. A central aspect of the project was ensuring that robots operated without colliding, following the strict rules prohibiting the sharing of squares or concurrent movement into occupied squares.

Illustrative examples visualized the constraints of the grid, the placement of highways, and the specific locations of picking stations. These examples offered a clear representation of the logic used to dictate robot movements and shelf placements within the warehouse environment. Throughout the project's executing, Clingo served as the primary computational tool, executing the defined scenarios precisely as per the project specifications. Clingo's output provided valuable insights into the sequence of robotic activities and the action necessary to fulfil orders. The solver's stable models were instrumental in revealing the steps required to complete each instance, showcasing actions within defined time increments.

The solution derived from Clingo showcased the efficiency of the system in addressing the challenges posed by the warehouse environment. For instance, the minimum number of time steps required for order completion in various instances ranged from 4 to 10, showcasing the system's adaptability and efficiency in handling different scenarios. The project encountered several challenges. Understanding the intricate details of ASP and troubleshooting the code proved to be demanding. Debugging was challenging due to the absence of adequate tools, leading to difficulties in comprehending the framework fully. This project successfully simulated an automated warehouse scenario using ASP with CLINGO. Insights from Dr.Joohung Lee's course on Knowledge Representation and Reasoning played a pivotal role in the project’s development, providing a robust foundation for problem solving and programming. The solution demonstrated the efficient handling of product handling, shelving and delivery to designated stations within the warehouse, emphasizing efficiency and adherence to constraints.

III. Results

The completion of this project involved several stages, each crucial to achieving the intended warehouse simulation. Initially, the establishment of rules governing the grid and highway laid the foundation for the subsequent development. Robot movements were programmed along both the X and Y axes, enabling them to traverse the warehouse grid effectively. Furthermore, procedures for picking up and putting down shelves were meticulously crafted to simulate the realistic handling of items within the warehouse scenario. The culmination of tasks involved ensuring the delivery of items to the designated picking stations while addressing associated limitations and constraints.

To illustrate the project’s functionality and validate its implementation, sample scenarios were utilized, covering various actions such as robot movements along different axes and handling shelf pickup and delivery to specific stations. Rules were devised to prevent robot collisions and enforce the Commonsense Law of Inertia. This law focused on maintaining the status of items, robot locations, shelf carrying actions, and order statuses within the simulated environment. The derived solutions from the project's execution using Clingo, an Answer Set Programming tool, were successful in executing the warehouse scenario precisely according to the project specifications. The execution encompassed all actions, instances, and constraints established within the project's framework.

Instance	**Timesteps**	**Number of Actions**
1	9	19
2	10	17
3	6	10
4	4	10
5	6	10

Table 1: Minimum Time-steps and number of actions required for solving instances.

The Clingo solver's stable models offered a comprehensive visualization of the robotic activities necessary for fulfilling orders. These activities were observed as singular operations occurring within defined time increments. Each alteration in time steps corresponded to a change in the atomic representation, reflecting the sequence of actions executed within the program. The results were quantitatively analysed to determine the minimum time steps and the number of actions required to solve each given instance. Table 1 presents the minimum time steps and the corresponding number of actions needed for solving the sample instances. Notably, as the time steps increased, the solver's execution time also increased. Instances one and two required more time to complete compared to the subsequent instances.

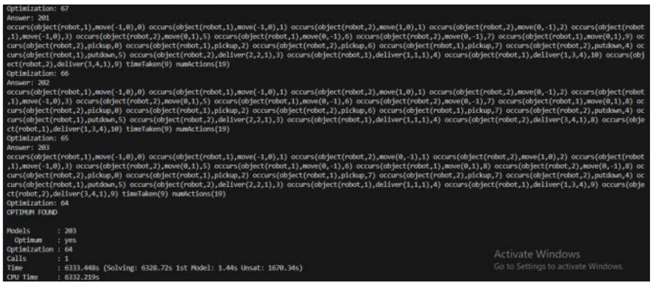

Image1: Instance1 Output

Image2: Instance2 Output

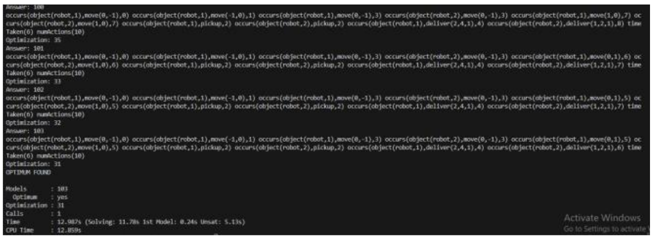

Image3: Instance3 output

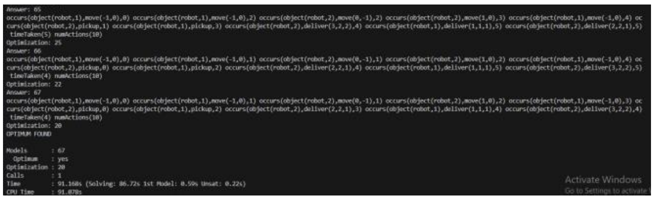

Image4: Instance4 output

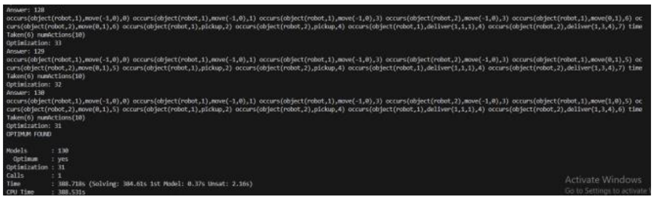

Image5: Instance5 output

Screenshots representing the output for each instance were generated to provide a visual representation of the execution results. These screenshots (Image1 to Image5) showcased the simulation outcomes for the respective instances, highlighting the robot movements, shelf placements, and overall order fulfilment actions within the warehouse environment. Throughout the project’s execution several issues were identified. Lack of clarity in understanding certain instances’ code within the input format of the problem statement posed challenges. Additionally, grappling with the intricate details of ASP and troubleshooting the code presented significant difficulties. The absence of adequate debugging tools compounded the challenge, along with a need for a complete understanding of the framework.

The project successfully simulated an automated warehouse scenario, demonstrating effective robot actions and shelf placements within defined constraints. Despite encountering challenges, the project provided valuable insights into automated warehouse simulation using Answer Set Programming, paving the way for potential improvements and further research in this domain.

IV. Lessons Learned

After The execution of the Automated Warehouse Scenario project necessitated the development of multifaceted skills and a deep comprehension of state-of-the-art technologies. The progression through different stages facilitated the cultivation of expertise across pivotal domains.

A. Utilization of Answer Set Programming (ASP) and Clingo:

The fundamental technology utilized in this endeavor centered on Answer Set Programming (ASP) in conjunction with Clingo. This encompassed harnessing Clingo's robust capabilities to resolve intricate combinatorial problems while representing knowledge effectively. Mastery in ASP syntax, comprehension of Clingo's unique execution process, and adeptness in utilizing Clingo to navigate complex problem-solving.

B. Embracing Knowledge Representation and Reasoning:

The project dived into the fundamentals of Knowledge Representation and Reasoning, specifically applying these principles to automate warehouse logistics. Profound understanding of formalisms for knowledge representation, transformation of logical deductions into constraints, and implementation of reasoning mechanisms for informed decision-making.

C. Implementation of the Davis–Putnam–Logemann–Loveland(DPLL) Algorithm:

The project capitalized on solvers based on the DPLL algorithm, enhancing efficiency in resolving propositional logic formulae. In-depth comprehension of the DPLL algorithm and its application in solving intricate logical problems.

D. Addressing Weak and Hard Constraints in Problem-Solving:

Addressing challenges associated with weak and hard constraints underscored the significance of managing constraints effectively to formulate robust solutions. Competence in handling weak and hard constraints while ensuring logical coherence in problem-solving strategies.

E. Code Debugging Precision and Optimization Strategies:

The implementation process demanded the adept handling of debugging challenges, emphasizing a meticulous approach to identify and resolve issues. Additionally, strategic optimization strategies were deployed to minimize program execution time-steps. Expertise in debugging intricate code segments, incorporating comprehensive comments for clarity, and systematic optimization techniques to enhance program efficiency.

F. Strategic Problem-Solving Methodologies and Planning:

Strategic and systematic problem-solving methodologies were crucial in addressing challenges pertaining to robot movements, collision prevention, and path planning. Strategic analysis, breakdown of complex problems into manageable segments, and parallel planning of sub-problems for effective solutions.

G. Application of Optimization Techniques:

Efforts to minimize time-steps necessitated the application of optimization techniques, leveraging Clingo's minimize function to achieve the minimum feasible time-steps. Application of optimization strategies, understanding trade-offs, and systematic reduction of time-steps while upholding the validity of the action plan.

H. Practical Experience with Minicoda and Anaconda Powershell:

The project involved utilizing Minicoda within Anaconda Powershell to code the action plan in Clingo ASP. Proficiency in utilizing Minicoda, executing commands within Anaconda Powershell, and coding within the specified environment.

I. Collaborative Knowledge Assimilation:

Recognizing the significance of collaborative knowledge assimilation from diverse sources, including class lectures, notes, online references, and prior programming tasks. Effective collaboration, assimilation of diverse knowledge resources, and the synthesis of information for comprehensive problem-solving.

J. Future Endeavors and Continuous Enhancement:

The project's culmination emphasized future avenues, such as expanding the warehouse scenario, integrating additional constraints, and exploring novel features. Acknowledgment of the iterative nature of software development, outlining concrete requirements, and delineating future trajectories for ongoing enhancement.

References

[1] J. Lee, "Monkey and banana problem explained," Lecture presented in CSE 579 - Knowledge Representation and Reasoning,Arizona State University.

[2] J. Lee, "Block words puzzle explanation," Lecture presented in CSE 579 - Knowledge Representation and Reasoning, Arizona State University

[3] J. Lee’s Lectures on Stable Models and Constraints from Module 3 and Module 4, Knowledge Representation and Reasoning, Arizona State University.

[4] R. Peixoto, L. Dias, M. S. Carvalho, G. Pereira and C. A. S. Geraldes, "An automated warehouse design validation using discrete simulation," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 2016, pp. 199-204, doi: 10.1109/ITSC.2016.7795554.

[5] H. Chu, Y. Zhang, J. Xiao, S. Zi and W. Wang, "An overview location navigation and path planning for automated guided vehicle," AIIPCC 2021; The Second International Conference on Artificial Intelligence, Information Processing and Cloud Computing, Hangzhou, China, 2021, pp. 1-6.