Case Study: Intelligent Systems in Production

RL-Driven Warehouse Item Relocation Optimization

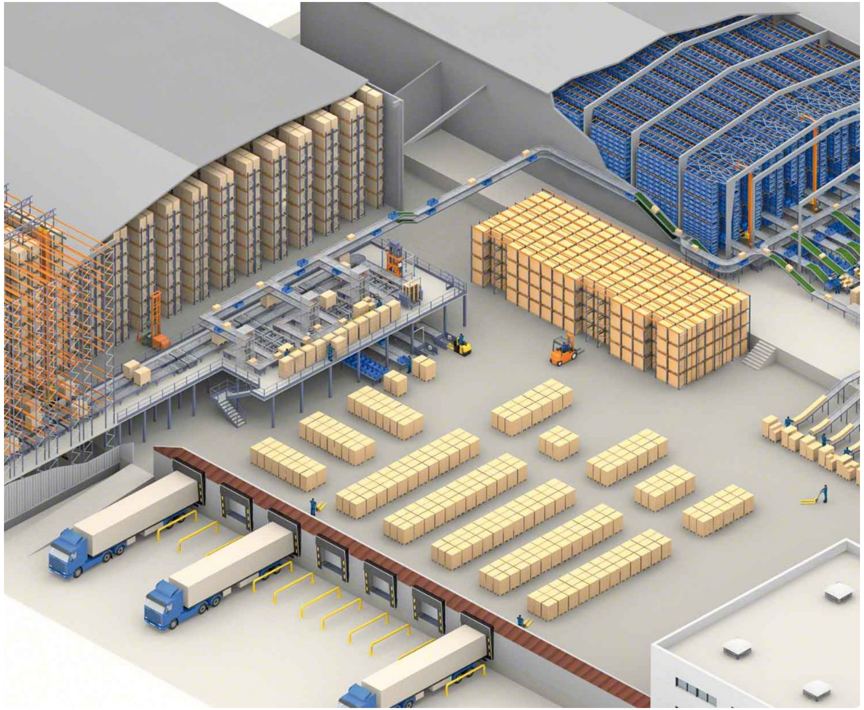
Prof. Hamidreza Heidari

Binil Sajeev, Abdelraheem Zekry, Orlando Manrique, Arthur Reis

Table of Contents

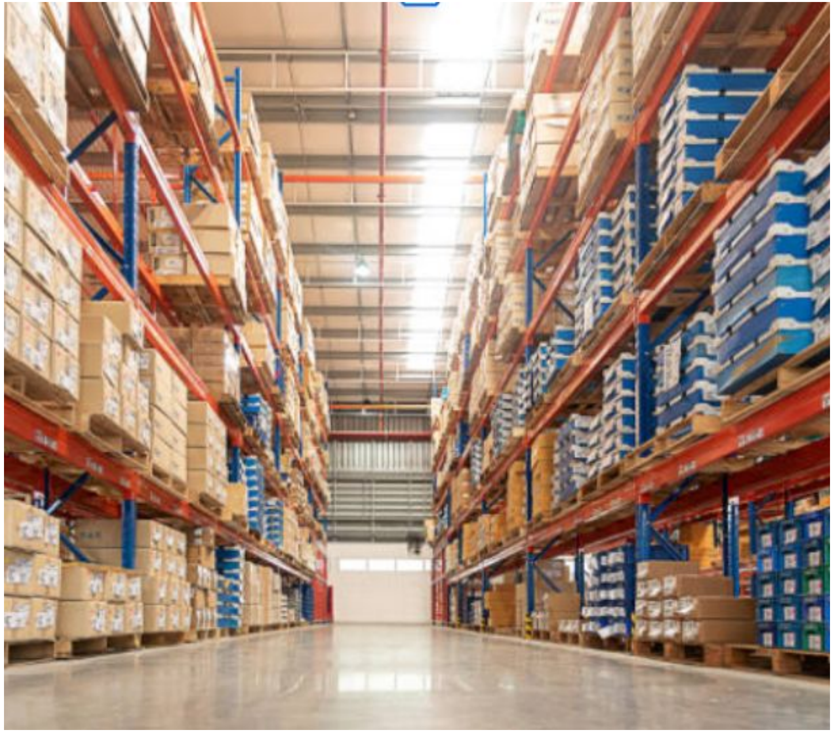
Context & Motivation

- A **warehouse** is a structured **storage facility** for **holding goods before distribution or production**.
- Organized into **zones, aisles, shelves, and bins** with defined capacity and access rules.
- Supports core operations: receiving, storing, picking, packing, shipping.
- Performance depends on how items are placed, affecting space use, travel time, and cost.



Context & Motivation

- Modern warehouses must balance **storage density**, **retrieval efficiency**, and **operational cost**.
- **Manual / rule-based slotting** cannot adapt to **dynamic inventory profiles** or demand changes.
- Result: **sub-optimal layouts**, **excessive relocations**, and **inefficient space utilization**.



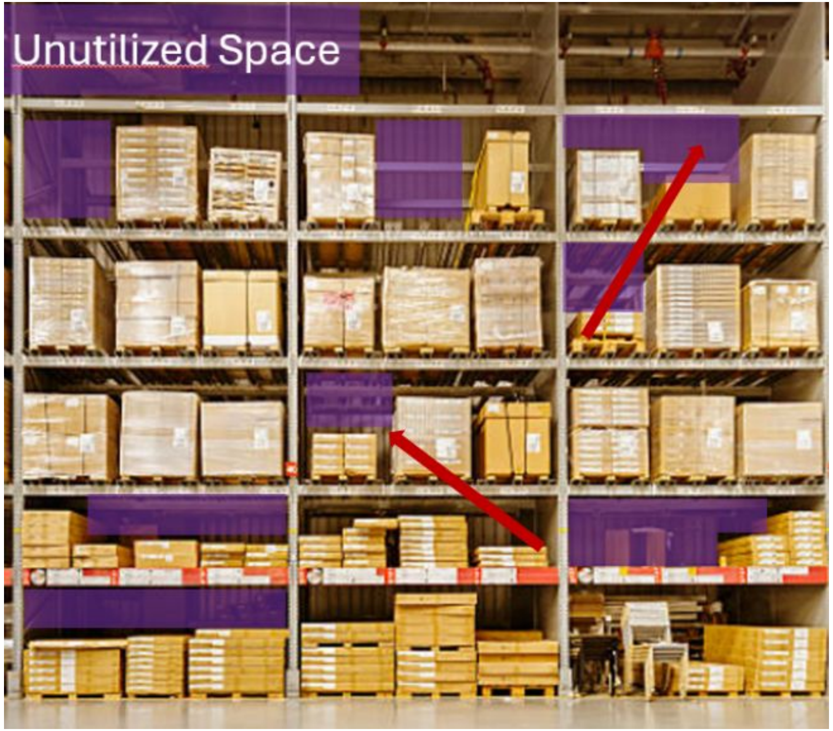
Problem Statement

Project Goal

- Develop an **RL-based optimization system** that identifies **efficient item relocation strategies** for warehouse configurations.

Objective

- **Maximize** usable **storage space** while **minimizing movement cost**, enabling pre-optimized layouts before peak operations.



Main Parameters

Warehouse Structure

- Layout: stacks, tiers, aisles, zones
- Entry/exit positions (distance & cost drivers)

Storage State Variables

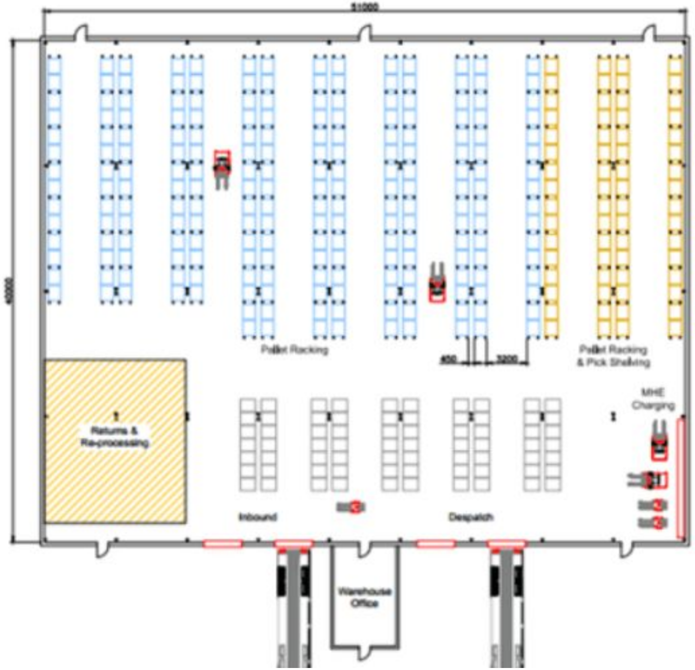
- Slot occupancy & item type
- Retrieval priority / blocking relationships

Operational Actions

- Retrieve, relocate, store
- Movement between stacks/zones

Temporal Factors

- Time-step per operation
- Arrival / retrieval rates



Evaluation and Metrics

Performance Metrics

- Total relocations
- Retrieval time / travel distance
- Transport cost
- Space utilization (%)

Objective / Reward Components

- Penalties: relocations, invalid moves
- Bonuses: efficient access, correct order
- Weighted efficiency index

Methodology

- Data generation (synthetic dataset).
- Environment design, reward function, constraints.
- Development of a Reinforcement Learning algorithm to interact with the environment.
- Simulation Framework: SimPy setup, scenarios, and validation assumptions.

Optimization Methods Evaluation

- Discrete-event simulation allows experimentation in a ***safe and cost-effective*** environment.
- Useful to analyze and compare several ***hypothetical scenarios*** and understand the impact of different variables.
- The model of phenomena of study must be as ***accurate and interpretable*** as possible in comparison to the real world.

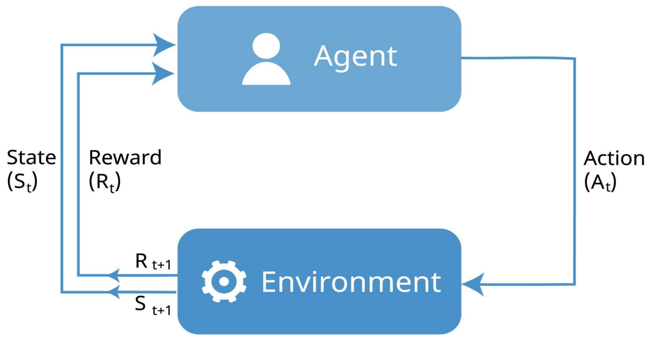
A short example

- Total storage 20 m^3, no restrictions.
- 2 SKUs → SKU1: 0.3 x 0.3 x 0.3; SKU2: 0.2 x 0.4 x 1.2
- Initial stock → SKU1 100 units; SKU2 130 units.
- Demand → SKU1 Normal(8,2.5); SKU2 Exp(0.3)
- Simulation time 36 months.
- KPI: available space in %

REINFORCEMENT LEARNING

- RL is a branch of Machine Learning where an agent learns by doing.
- The agent isn't told what to do. It interacts with its environment, trying actions and observing outcomes. So it doesn't have to be retrained.
- It receives rewards or penalties based on how good or bad its actions are.
- Over time, it creates optimal policy that maximizes total rewards & minimizes mistakes.

Reinforcement Learning



WHY RL OVER ML & HEURISTIC METHODS?

Reinforcement Learning (RL)	**Rule-Based / Heuristic Methods**
● Learns automatically from data and improves over time. ● Considers many factors at once (fit, demand, bins, zones). ● Adapts to changing demand patterns in the warehouse. ● Makes smart, selective relocations based on long-term benefits. ● Can discover better strategies that humans don't think of.	● Must be manually tuned and adjusted by humans ● Usually follow simple rules (eg: demand only). Break down when demand or layout changes. ● Decisions are short-term and not sequential. ● Stuck with predefined, no learning.

WHY RL → Supported by Research

Research Study	RL Outcome	Limitation of Heuristic / Rule-based methods
Liu et al., 2025	25% fewer relocations	Heuristics collapse on large instances
Koo et al., 2022	14% less travel distance	ABC-slotting fails with changing demand
Wu et al., 2024	32% more throughput	Rules can't handle dynamic congestion
Aljohani et al., 2023	15% more robust under uncertainty	Heuristics overfit static scenarios
Troch et al., 2023	20% shorter walking distance	Heuristics miss non-intuitive strategies
Röhrig & Böse, 2021	30% fewer container moves	Tabu Search requires retuning & scales poorly

HOW RL WORKS

MDP

- Model as a Markov Decision Process (MDP) Mathematical framework that RL uses in decision making

MDP COMPONENTS

- **State:** Current item locations
- **Environment:** The external world with which the agent interacts.
- **Action:** Move item i from slot A → B (relocation)
- **Policy:** Strategy the agent develops to decide which actions maximizes cumulative reward.
- **Reward:** Feedback from the environment tells the agent how good or bad last action was.
- **Episode:** Number of relocation steps to reach goal per simulation

HOW RL WORKS IN OUR CASE

Project goal

- To maximize space utilization while minimizing unnecessary movement operations in a smart warehouse.

RL APPROACH

- Build a warehouse simulator
- RL agent aim to relocate products
- Use reward shaping & action masking to ensure valid moves

EVALUATION OF MODEL

- Optimization baselines & the previous condition the warehouse was
- Storage utilization, Travel time in warehouse, Relocation Cost

RL MODEL - Utilizing Warehouse Demo

Goal: Create an RL agent that decides which part goes into which bin to make the best use of space while avoiding unnecessary movements.

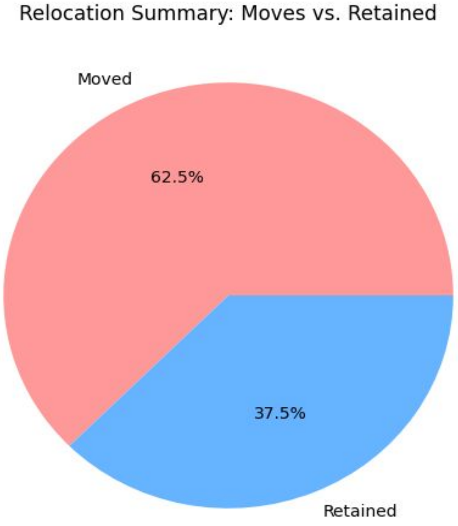
Setup: Created a virtual warehouse environment simulation,The model uses synthetic data that contains parts & bins dimensions inside the warehouse

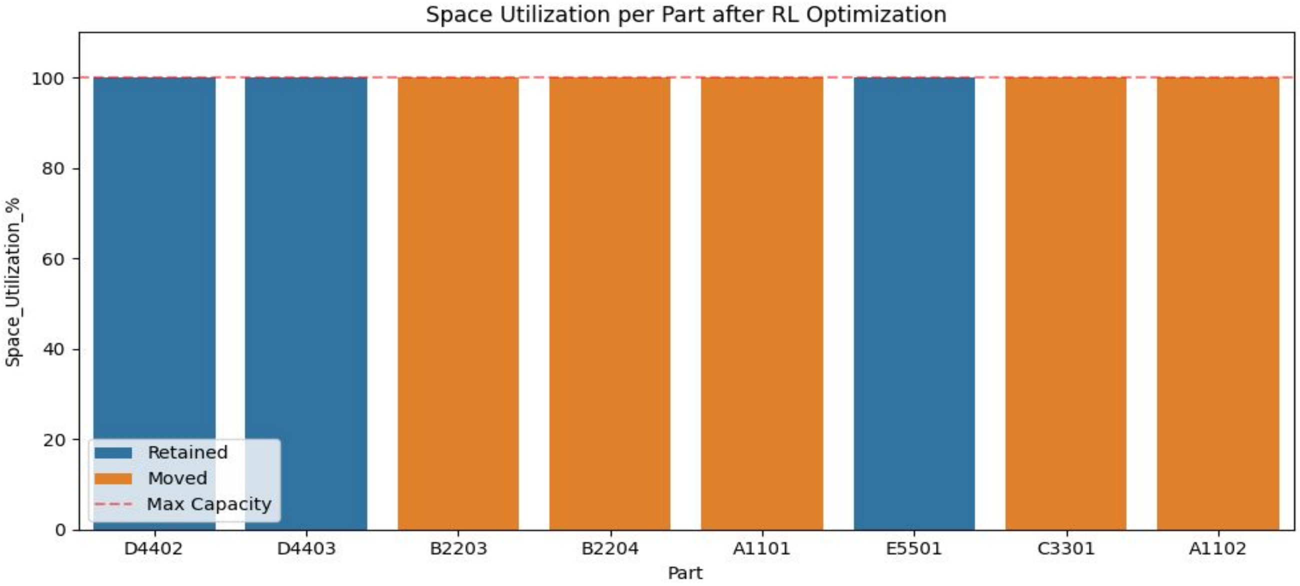
How the model works?

- Looks at every part number & storage number then calculates its Total Volume
- The "Environment" simply matches every part with every Location using criterias such as Physical Fit of part in the bin & maximum space utilization of each bin
- Model Receives **High Reward**: If the part fits perfectly and fills up the bin (High Utilization). Receives **Penalty**: If the part doesn't fit (physically impossible).

RL MODEL - Utilizing Warehouse Demo

Results: Shows how efficiently each part fits into its assigned bin after the optimization process. And 62.5% of parts were relocated to achieve 100% of space utilization





CONCLUSION

- Defined the warehouse relocation problem, constraints, and optimization goals.
- Designed the RL framework using an **MDP** with states, actions, rewards, and transitions.
- Built a virtual warehouse simulator using synthetic parts and bin data.
- Implemented a baseline **RL agent** with reward/penalty shaping and generated relocation & utilization results.
- Compared RL with heuristic methods, supported by findings from related research.

THANK YOU

Binil Sajeev, Abdelraheem Zekry, Orlando Manrique, Arthur Reis