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

INTRODUCTION

- Warehouse Operations Overview
- Storage Efficiency
- Item placement - how to balance:
    - Storage Density
    - Retrieval Efficiency
    - Operational Costs
- Limitations of other methods
- RL Model applications:
    - Complex/Large problems
    - Generalization

INTRODUCTION

- Warehouse Operations
- Storage Efficiency (Densification)
- Item placement challenges: slotting system.
- Traditional manual methods.
- Dynamic inventory and demand patterns.
- Complexity: too many SKUs.

PROBLEM STATEMENT

Research questions

- How can space utilization be maximised?
- How can unnecessary movements be minimised?
- How to increment the efficiency and operational cost in an automobile parts warehouse?

Case Study Objectives

- Develop a smart relocation system
- Develop a simulated warehouse model.
- Apply RL/ML models on the optimisation problem.
- Adjust the parameters of the simulation through iterations.

RELATED SCIENTIFIC WORKS

Study Title	Research Focus:	Methodology	Key Findings	Relevance
Liu et al. (2025). A Q-learning-based algorithm for the block relocation problem	Q-learning algorithm for the Block Relocation Problem (RBRP-dup) to minimize relocations in stacked storage systems.	Modeled BRP as an Markov Decision Problem; applied Q-learning with heuristic-based initialization and rule-filtered actions; compared to heuristic and exact methods on benchmarks.	RL method converges faster, scales better for large instances, and achieves competitive or superior solutions versus heuristics.	Demonstrates how reinforcement learning can optimize storage operations similar to warehouse slotting or relocation tasks.
Wu, Shao-Ci & Chiu, Wei-Yu & Wu, Chien-Feng. (2024). Deep Reinforcement Learning for Task Assignment and Shelf Reallocation in Smart Warehouses.	Use of deep Q-network to assign robot tasks and reallocate shelves in a smart warehouse, considering inventory and shelf locations to improve efficiency.	Simulated a warehouse; Deep Q-Network (DQN) models assign tasks and shelf returns; compared to heuristic Regret-Marginal Cost Algorithm (RMCA).	The DQN-based approach outperforms RMCA in most tested scenarios: lower Cycle Processing Time, higher throughput, and faster runtimes. It scales with number of robots and warehouse size, though congestion and scale still degrade performance.	Shows how to integrate inventory and shelf return logic into task assignment. Also gives a template for state representations, rewards, and comparison against heuristic baselines.

RELATED SCIENTIFIC WORKS

Study Title	Research Focus:	Methodology	Key Findings	Relevance
Arslan, E. (2025). Optimizing Human-Centric Warehouse Operations: A Digital Twin Approach Using Dynamic Algorithms and AI/ML.	Integrate digital twin with AI/ML to optimize manual warehouse operations (picking, routing, slotting).	Developed a digital twin framework layered architecture; used clustering, dynamic algorithms, forecasting models (SVM, decision trees, LSTM) in a prototype setting; ran “before vs after” experiments.	Improved KPIs in warehouse management. Aprox. 28.6% reduction in picking time; inventory turnover improved 20%; forecasting accuracy rose from 85% to 92%; labor cost reduction in 15%.	Shows practical digital twin and ML applied in human-centric warehouses. It presents a benchmark of improvements and a template for combining simulation, analytics, and adaptation.

EXPECTED OUTCOMES

- A **functional prototype (MVP)** capable of generating optimized warehouse configurations based on defined parameters and constraints.
- A **synthetic dataset** modeled after realistic warehouse conditions, ensuring data privacy and compliance while maintaining validity.
- A **visual interface or dashboard** for analyzing warehouse states and tracking relocation improvements.
- A **Feedback Loop** where the results of the optimized parameters are used as input for the model, starting an iterative process.
- A **documented framework** that can later be extended or integrated with real warehouse data for production-level deployment.

TECH STACK + CHALLENGES

Tech Stack

- Programming Language: Python
- Simulation: SimPy, Anylogic
- Optimization & AI: PyTorch (Reinforcement Learning), TensorFlow, custom heuristic solvers.
- Data Handling: NumPy, Pandas.
- Visualization: Matplotlib, Plotly, or a web-based 2D warehouse viewer (e.g., Dash, Streamlit, or React front-end).

Challenges

- Limited Access to Real-World Data
- Combinatorial Complexity
- Trade-off Tuning.
- Generalization: risk of over and underfitting.

METHODOLOGY

- **Step 1:** Warehouse Representation
- **Step 2:** Obtain Synthetic Data
- **Step 3:** Optimization Engine
- **Step 4:** Fitness / Objective Function
- **Step 5:** Ensure Reproducibility

  GitHub Repository

THANK YOU

Binil Sajeev, Abdelraheem Zekry, Orlando Manrique, Arthur Noroes