Case Study Intelligent Systems in Production: RL-Driven Warehouse Item Relocation Optimization

Student Name	Email
Arthur Noroes Reis	arthur.noroes-reis@stud.th-deg.de
Binil Sajeev	binil.sajeev@stud.th-deg.de
Orlando Manrique	orlando.manrique-olmos@stud.th-deg.de
Abdelraheem Zekry	abdelraheem.zekry@stud.th-deg.de

I. Introduction

The rapid growth of e-commerce and changing customer expectations have pushed warehouses to operate with higher efficiency. For example, nowadays customers expect shorter delivery times, no errors and maximum quality possible. Therefore, logistics must be performed with optimal space utilization, and shorter picking times.

Nevertheless, currently most warehouses still apply traditional slotting and storage methods, which often rely on fixed rules and struggle to adapt to shifting demand and limited floor space, leading to bottlenecks and potential errors when volume increases massively. Several studies point out that item relocation and picking activities represent a large part of warehouse operating costs, especially when products are not arranged according to demand frequency (Duque-Jaramillo et al., 2024).

At the same time, research on discrete-event simulation shows that process-based modeling can help predict congestion, travel times, and resource usage (Matloff, 2008). Furthermore, cutting-edge approaches combine simulation with optimization techniques such as genetic algorithms (Torlaschi, 2020) or digital twins (Arslan, 2025), offering better insight into how layout changes affect performance without modifying any parameter in the physical world.

Moreover, advances in artificial intelligence have created new opportunities for learning-based decision making in warehouse picking and item relocation. Reinforcement learning has also gained attention because it can learn sequential relocation policies that adapt to changing SKU demand over time. This concept can be applied in this case study since storage is a dynamic process where product demand changes over time (be it seasonality or trends). Thus, it is possible to model a warehouse throughout different points of time, defining the states which serve as baseline for the agent to take actions.

The ability to continuously improve storage decisions can offer meaningful performance gains. The goal of this literature review is to explore these methods and identify how

reinforcement learning could be applied to improve storage density, movement cost, and picking efficiency in a simulated warehouse environment.
Building on these considerations, it becomes evident that warehouse optimization is not a static challenge but a dynamic one—where item placement, accessibility, and retrieval efficiency must continuously adapt to fluctuating demand. Reinforcement learning, with its capacity to learn optimal strategies through experience, presents a promising avenue for enabling such adaptability. This perspective naturally leads to the central inquiry of this study:

How can reinforcement learning improve warehouse storage density and picking accessibility through adaptive item relocation under varying demand patterns?

Keywords:
Warehouse optimization, item relocation, picking, slotting, travel distance, movement cost, warehouse simulation, reinforcement learning, machine learning, SimPy, SKU demand, smart logistics.

II. Literature Selection Criteria

Only sources in English were consulted. The literature collected consists of research papers and books retrieved using the keywords stated above. The research publications were also filtered by publication date, including sources no older than 15 years old. Regarding books, we only selected sources that were published or revised later than 2000.

III. Summary of Findings

The analysis of reviewed literature highlights the critical factors that influence warehouse efficiency and the optimization of item relocation. Key parameters consistently emphasized include warehouse layout and structure, storage state variables, operational actions, temporal and process factors, performance metrics, reward functions, and input data. Warehouse layout and structure—encompassing the number of stacks, tiers, shelves, aisles, and zones—directly affect spatial constraints, travel distance, and relocation feasibility, as shown in Liu et al. (2025) and Waubert de Puiseau et al. (2022). Storage state variables, including occupancy, item types, retrieval priorities, and blocking indicators, define the state space for reinforcement learning agents and influence relocation and retrieval logic. Operational actions, such as retrieval, relocation, storage, and movement, map to RL actions and also inform ILP- and TSP-based sequencing frameworks, as exemplified in Duque-Jaramillo et al. (2024). Temporal factors, including simulation steps and demand variations, while less critical for static relocation, provide the foundation for dynamic scheduling in adaptive environments, as illustrated in Cestero et al. (2022). Performance metrics such as total relocations, travel distance, transport cost, and space utilization offer objective criteria for optimization and reward design, with optional indicators like FIFO compliance providing additional guidance for priority-based retrieval.

Building on these parameters, reinforcement learning emerges as a particularly effective approach for warehouse item relocation. Unlike static heuristics, ILP, or rule-based slotting methods, RL allows agents to learn adaptive, sequential relocation policies that optimize both space utilization and movement efficiency while accommodating variable item types and warehouse layouts. Reward functions can combine penalties for inefficient moves with bonuses for correct retrieval sequences, guiding agents toward cost- and time-efficient strategies. Empirical data, including historical operations, zone capacities, and pallet characteristics, supply realistic training environments that enhance the generalizability and reproducibility of learned policies. Overall, the integration of structural, operational, and performance-focused parameters within an RL framework provides a robust, data-driven methodology for optimizing warehouse relocation strategies, enabling continuous improvement, adaptability to changing conditions, and measurable efficiency gains across both simulated and real-world warehouse systems.

IV. Discussion

During the analysis of the reviewed literature, particular attention was given to the factors most influential in determining warehouse performance. Two aspects emerged as especially significant: the key parameters and performance indicators used to evaluate warehouse efficiency, and the methods or approaches employed to achieve optimization. Examining these elements allowed for a clearer understanding of how different studies define success, what variables they prioritize, and which methodological frameworks yield the most consistent or innovative results.

A. Comparison and Rationale for Choosing Reinforcement Learning (RL) for Warehouse Item Relocation Optimization

1. Existing Methods for Warehouse Optimization

A variety of techniques have been used to optimize warehouse layouts and relocation problems before the emergence of RL-based methods.
Early studies relied on rule-based heuristics such as the *ABC classification* and *Cube-per-Order Index (COI)* (Duque-Jaramillo et al., 2024), which prioritize items by demand frequency and volume.
Later, metaheuristic algorithms were introduced to search large configuration spaces more efficiently.
For example, Tabu Search was used by *Chen, Langevin & Riopel (2011)* to minimize relocation and travel costs by iteratively improving feasible layouts while avoiding previously explored solutions.
Genetic Algorithms were applied in simulation frameworks such as *Torlaschi (2020)* to evolve near-optimal designs for automated warehouses.
Similarly, Mixed-Integer Linear Programming (MILP) and Sorting-based Slotting approaches (Duque-Jaramillo et al., 2024; Viveros et al., 2021) achieved gains in specific, static configurations.

While these techniques provide valuable baselines, they share a major limitation: they operate on fixed datasets or static assumptions.
Each time demand or product profiles change, the model must be recalibrated or rerun.
This limits scalability in dynamic environments where order patterns fluctuate frequently.

2. Why Reinforcement Learning (RL) Is a Promising Alternative

Reinforcement Learning (RL) differs fundamentally by allowing an **agent to learn through interaction** with a simulated warehouse environment, receiving rewards or penalties after each relocation decision.
RL can therefore adapt continuously to new patterns, evaluate long-term effects of each action, and autonomously refine strategies that balance space utilization, accessibility, and movement cost.

The table below summarizes key findings from the reviewed studies and highlights both the **advantages of RL** and the **limitations of alternative methods**.

Research Paper	**Why RL is Better (as Claimed/Shown)**	**Disadvantages of ML / Rule-based / Heuristic Methods**
Liu et al. (2025) — Q-learning for the Block Relocation Problem	For large, complex instances, Q-learning outperformed conventional exact/heuristic methods and learned sequential relocation policies that adapt dynamically to different layouts.	Exact/heuristic methods work well on small cases but struggle to scale; performance advantages fade as problem complexity grows.
Wu, Chiu & Wu (2024) — Deep RL for Task Assignment & Shelf Reallocation	Deep RL enables real-time, adaptive decision-making under uncertainty, improving throughput and reducing delays.	Heuristic baselines underperform in high-dimensional, time-varying coordination problems and lack adaptability.
Koo, Lee & Kim (2022) — Dynamic Storage Using Deep RL	DDQN adapts storage assignments to fluctuating demand, reduces handling/travel costs, and generalizes to stochastic environments.	Static slotting (e.g., ABC rules) fails to adapt to changing demand; conventional optimization struggles under variability.
Aljohani, Reggelin & Schenk (2023) — Hybrid Simulation + RL	Simulation-trained RL policies improved throughput and space utilization versus heuristics, learning robust strategies from stochastic data.	Heuristics cannot handle uncertainty or randomness effectively and tend to overfit to static conditions.
Röhrig & Böse (2021) — RL for Container Relocation	RL learned efficient relocation sequences that reduced unnecessary moves by 15–25% and scaled across yard sizes.	Rule-based strategies miss optimal long-term sequences and incur extra relocations as complexity increases.
Troch et al. (2023) — RL for Storage Location Assignment	Q-learning improved efficiency (~20%) and generalized across layouts, outperforming static allocation rules in adaptive scenarios.	Frequency-based or random allocation methods fail in dynamic environments and lack adaptability to demand changes.

Cestero et al. (2022) — Storehouse RL Environment	Demonstrated that RL agents can learn to minimize travel distance and improve picking efficiency by interacting with a simulated warehouse.	Traditional rule systems require manual design per layout and lack flexibility for diverse configurations.
Chen, Langevin & Riopel (2011) — Tabu Search for Relocation (Heuristic Baseline)	Serves as a baseline showing that heuristic methods are limited in scalability compared to RL approaches.	Tabu search improves initial solutions but degrades in performance as problem complexity increases, unlike learned RL policies.

3. Limitations and Challenges of Reinforcement Learning

Although RL provides significant advantages, several studies also identify its **limitations**, which must be considered for balanced evaluation:

- **High computational demand and training time:** Liu et al. (2025) note that convergence may be slow, particularly for very large state-action spaces.
- **Dependence on simulation quality:** Aljohani et al. (2023) emphasize that RL performance heavily depends on how accurately the simulated environment represents real operations; poor modeling can cause overfitting.
- **Reward design complexity:** Koo et al. (2022) report that defining an appropriate reward function balancing travel cost and utilization is non-trivial.
- **Stability issues in deep RL:** Wu et al. (2024) mention that deep Q-networks require careful parameter tuning to maintain stable learning.
- **Interpretability and integration challenges:** Cestero et al. (2022) acknowledge that translating learned policies into actionable warehouse control rules remains an open challenge.

4. Summary

In conclusion, while traditional heuristics (e.g., Tabu Search, Genetic Algorithms, MILP) offer computationally efficient yet static solutions, Reinforcement Learning provides a **data-driven, adaptive, and sequential framework** suited to dynamic warehouse environments.
Despite its higher computational cost and implementation complexity, RL's ability to **learn continuously and optimize long-term performance** makes it a compelling approach for modern item-relocation optimization.

B. Key Parameters and Performance Indicators

A critical aspect of warehouse optimization and item relocation using reinforcement learning is the identification of parameters and performance indicators that define the system state, guide decision-making, and measure outcomes. Across the reviewed literature, several studies have formalized these parameters within structured environments, specifying constraints, priority rules, and operational metrics that influence the efficiency of retrieval and relocation processes.

The following table summarizes the key parameters and performance indicators reported in selected studies and highlights their potential application for warehouse optimization. This synthesis provides a foundation for modeling warehouse configurations, defining action spaces for reinforcement learning agents, and establishing objective functions that balance storage density and operational efficiency.

Research Paper	Key Parameters and Performance Indicators	Potential Application
Wu et al., 2024 – "Deep Reinforcement Learning for Task Assignment and Shelf Reallocation in Smart Warehouses"	Warehouse layout: 2D grid - Number of robots and positions - Robot carrying status - Inventory: item types, quantities, shelf capacities - Orders and tasks with locations and timings - Operational constraints: collisions, movement limits - Objectives: task time, relative cost, makespan, total cost	- Warehouse representation: grid and shelf positions for RL state. - Item/shelf parameters: define environment state. - Movement rules: inform relocation actions. - Task sequencing: guides prioritization in RL. - Performance metrics: can be adapted as reward signals balancing efficiency and movements.
Leon et al., 2023 – "A Hybrid Simulation and Reinforcement Learning Algorithm for Enhancing Efficiency in Warehouse Operations"	- Warehouse layout and storage locations (4 racks in the case study) - Item types and arrival probabilities (A–D) - Rack capacities (50 items) - Actions: storage location assignment - Observations: current stock per rack + incoming item type - Reward: travel distance of pickers (inverse of change from previous step) - RL algorithms: A2C, PPO, DQN - Simulation environment: FlexSim, connected to Python via sockets	- State representation: rack inventory and incoming item type as RL state. - Action mapping: discrete rack selection for each incoming item. - Reward design: travel distance optimization guides RL policy. - RL algorithm selection: A2C/PPO/DQN can be benchmarked for your dynamic storage allocation. - Simulation integration: FlexSim as a realistic warehouse environment to train RL agents for task assignment and storage decisions. - Benchmark comparison: random vs. greedy policies allow evaluation of RL learning efficiency.

Waubert de Puiseau, C., Nanfack, D. T., Tercan, H., Löbbert-Plattfaut, J. & Meisen, T. (2022). Dynamic Storage Location Assignment in Warehouses Using Deep Reinforcement Learning. Technologies, 10(6), 129.	-Warehouse model: Semi-automatic high-bay warehouse with 3 storage zones (A, B, C), two ASTS systems, conveyor belts between racks, single entry point. -Simulation: Python (OpenAI Gym) tracking transport cost as a time-proportional unitless metric. State: Zone utilization, goods type (500 types), entry/re-entry flag. -Action: Assign pallet to zone A, B, or C. -Reward: Negative transport cost (A = 1, B = 2, C = 10 units). -Data: 12,100 historical operations from 2021–2022.	- Zone structure and utilization data can define spatial constraints in storage simulations. -Goods type and entry/re-entry information can support demand-aware slotting models. -Transport cost metrics can guide RL reward shaping for efficiency optimization. -Historical operation logs provide a baseline for benchmarking model performance and reproducibility.
Cestero et al. (2022) – "Storehouse: A Reinforcement Learning Environment for Optimizing Warehouse Management"	-Metrics: delivery rate, average item age, FIFO violations. -Layout: grid-based warehouse with entry and delivery points, restricted cells. -Items: multiple material types generated stochastically (Poisson process). -State: 3D tensor of feature grids (boxes, age, agents, entry/delivery status). -Actions: agent movement on grid, pick/drop behavior, invalid moves penalized. -Reward: −1 for invalid, −0.9 for idleness, up to 0 for FIFO-compliant deliveries.	- The delivery-rate metric supports designing RL reward functions rooted in throughput, useful for warehouse relocation or slotting to prioritize frequently moved items. - Age of material & FIFO violation metrics can be adopted to capture accessibility and freshness/access-order in item placement—helpful when modeling retrieval accessibility in item relocation tasks. - The grid-based warehouse model and detailed state-feature design (box grid, age grid, restricted cells, agent grid) provide a blueprint for encoding warehouse layout, constraints, and agent state in RL relocation systems. - The actions and reward definitions provide a template for designing relocation moves, penalty structure for invalid moves, and incentives for efficient picks/drops in a simulated warehouse RL framework.
Duque-Jaramillo et al. (2024). "Warehouse Management Optimization Using a Sorting-Based Slotting Approach"	-Two-phase model: (1) SKU priority via ABC and COI; (2) slot assignment and operation-time evaluation. -Warehouse grid: Sections × Rows × Columns × Levels; 24 sequencing options; six slot-assignment sequences (SAS). -Travel time: Manhattan distance; speed depends on load; proximity to entry/exit prioritized.	-Used to optimize slot allocation in large pallet-based warehouses. -Applicable to distribution centers needing adaptive slotting and route planning to minimize travel time. -Supports digital twin or WMS simulation for priority-based order picking efficiency.

	-ILP model: minimizes total operation time per period; decision variables for slot and quantity assignment. -Data assumptions: uniform slots, identical pallets, known SKU demand.	

Building on the insights from the reviewed literature, the following table translates the key concepts, methodologies, and performance indicators into the specific parameters and metrics that will be applied in this project. While previous studies focused on general warehouse layouts, storage assignment, or task allocation using reinforcement learning and simulation, our project adapts these principles to a block- and container-based warehouse environment with dynamic relocation operations. The parameters listed below capture both the structural and operational characteristics of the warehouse, the decision-making actions available to the agents, and the metrics needed to evaluate performance in line with the research objectives.

Category	Parameter / Indicator	Purpose / How It Connects to Literature
Warehouse Layout & Structure	- Number of stacks / tiers / shelves - Number of aisles and bays - Zone or block configuration (A–C or similar) - Entry/exit point positions	Defines spatial constraints and travel distance, based on Liu et al. (2025) and Waubert de Puiseau et al. (2022). Enables cost/time computation and relocation feasibility.
Storage State Variables	-Occupancy per stack/tier/zone - Item type per slot - Item retrieval priority / label - Blocking vs. target containers	Determines RL state space (as in CRP papers); influences relocation and retrieval logic.
Operational Actions	- Retrieval - Relocation - Storage (insertion) - Movement (to/from stack, to destination)	Captures discrete warehouse actions; maps directly to RL actions and ILP/TSP-based sequencing from Duque-Jaramillo et al. (2024).

Temporal / Process Factors	- Simulation time step (per move or operation) - Arrival / retrieval rate - Demand variation (static vs. dynamic)	Enables dynamic scheduling, as in Cestero et al. (2022) and Liu et al. (2025); determines environment dynamics for RL.
Performance Metrics	- Total relocations (CRP efficiency) - Average retrieval time per order - Total travel distance / time per cycle - Transport cost (zone-based) - Average material age (optional, FIFO indicator) - Space utilization rate (%)	Primary optimization objectives and RL reward design components, drawn from Waubert de Puiseau (transport cost), Cestero (delivery rate, FIFO), and Duque-Jaramillo (operation time).
Reward / Objective Function	- Negative relocation count or transport time - Penalty for blocking or invalid moves - Bonus for FIFO / correct retrieval order - Weighted efficiency index combining cost, time, and utilization	RL optimization target combining CRP efficiency and throughput, inspired by all prior studies.
Data & Input Parameters	- Historical operation data (container IDs, retrieval order, timestamps) - Zone capacity and utilization - Pallet size, homogeneity assumptions	Forms the empirical and simulation input base; aligns with Duque-Jaramillo (2024) and Waubert de Puiseau (2022) setups.

V. Comparative Analysis of Reviewed Studies

1. Liu et al. (2025). "A Q-learning-based algorithm for the block relocation problem"
https://link.springer.com/article/10.1007/s10732-024-09545-y

Research Focus: This paper tackles the restricted block relocation problem with duplicate priorities (RBRP-dup), which models a block stacking system where multiple items share the same retrieval priority. The authors seek to minimize the number of unnecessary relocations during retrieval operations — a key issue in container yards and logistics terminals where stacking constraints impose heavy operational costs.

Methodology: The authors model the RBRP-dup as a Markov decision process and apply a Q-learning algorithm in two phases. In the first (learning) phase, they incorporate an optimal-rule-integrated behavior policy and heuristic dynamic initialization to shrink the state-action space and improve convergence. In the second (optimization) phase, they combine the learned Q-values with a heuristic search procedure to refine decision-making. They benchmark their method against state-of-the-art exact and heuristic solutions on standard datasets.

Key Findings: The results demonstrate that for large and complex instances, the Q-learning-based approach outperforms conventional exact algorithms and heuristics — in about 27% of such cases it found better solutions (improvement of 1.9%–6.8%), and in another ~21% it matched the best known solutions. Moreover, the introduction of the optimal-rule policy and initialization method significantly accelerated convergence and stabilized learning. However, for small-scale instances, the advantage over exact algorithms was negligible.

Relevance: This work is relevant because it marks one of the first successful applications of reinforcement learning (specifically Q-learning) to a classical combinatorial optimization problem in logistics. It demonstrates that RL can be a meaningful alternative to exact and heuristic approaches in retrieval and relocation systems, offering a new tool for warehouse and yard layout optimization, an area closely aligned with our project's focus.

Scope Comparison: Whereas many earlier studies on block or container relocation problems focus on static heuristics, integer programming, or metaheuristics for small-to-medium problem sizes, this paper extends the scope by targeting large, complex instances and incorporating learning-based decision frameworks. By contrast, our project moves beyond stack-level relocation to a full warehouse layout context spanning multiple item types and zones, while Liu et al. focus on "block stacking" retrieval optimization, we aim for a general relocation and slotting optimization framework across a warehouse.

2. Wu, Chiu & Wu (2024). "Deep Reinforcement Learning for Task Assignment and Shelf Reallocation in Smart Warehouses".
https://www.researchgate.net/publication/380049934_Deep_Reinforcement_Learning_for_Task_Assignment_and_Shelf_Reallocation_in_Smart_Warehouses

Research Focus: This paper investigates how to improve operational efficiency in robot-assisted smart warehouses by integrating task assignment (which shelf to pick next) with shelf reallocation (where to return it) under dynamic and uncertain conditions. The authors focus on enhancing coordination between multiple robots and storage zones by allowing the system to make data-driven decisions using real-time inventory and positional information.

Methodology: The authors model warehouse operations as a deep reinforcement learning (DRL) problem using a deep Q-network (DQN) framework. The state space includes robot positions, shelf locations, task queues, and congestion levels, while the action space involves both task assignment and shelf return placement. The DRL agent learns policies that balance throughput and travel cost, and the model is benchmarked against heuristic methods across simulated warehouse layouts of various sizes.

Key Findings: The DQN-based framework significantly improved task completion time and overall throughput compared to heuristic baselines, especially in larger and denser environments. The approach also demonstrated stable convergence and good scalability, showing potential for real-time use. The results suggest that reinforcement learning can effectively handle the high-dimensional, dynamic nature of warehouse coordination problems.

Relevance: This study is relevant to our project because it showcases how reinforcement learning can be used to coordinate warehouse operations that involve both movement and reallocation decisions. It provides useful insights for modeling environment states and designing reward functions that align efficiency and adaptability, both essential for our relocation optimization framework.

Scope Comparison: While Wu et al. focus on dynamic, real-time decision-making for robot task scheduling and shelf return operations, our project emphasizes static or semi-static warehouse configuration planning — determining optimal item relocation strategies to improve density and accessibility before peak operations. However, we do plan to partially simulate warehouse operations alongside the optimization process to ensure that our relocation strategies remain practical when applied in dynamic or semi-real environments.

3. Chen, Langevin & Riopel (2011). "A Tabu Search Algorithm for the Relocation Problem in a Warehousing System".
https://www.sciencedirect.com/science/article/abs/pii/S0925527310003506

Research Focus: This study examines the item relocation problem in a warehousing system operating under dynamic conditions. It focuses on determining which items to relocate, and to which destinations, in order to minimize travel and handling costs when congestion or inefficiencies occur. The authors treat relocation as a discrete optimization problem involving capacity, accessibility, and operational constraints.

Methodology: The authors formulate the relocation problem as an integer-linear program and develop a two-stage heuristic to generate initial feasible solutions. A Tabu Search metaheuristic is then applied to iteratively improve these solutions. The search explores a neighborhood of possible relocation moves while using a short-term memory (the tabu list) to prevent cycling back to recently visited configurations, thus promoting exploration of new, potentially better layouts.

Key Findings: The Tabu Search algorithm significantly improved upon the initial heuristic solutions, consistently reducing relocation and travel costs across multiple test scenarios. It proved particularly effective in moderately sized warehouse problems, where it achieved near-optimal solutions with relatively low computational effort. However, its efficiency declined slightly with increasing problem complexity, highlighting scalability limits common to metaheuristic methods.

Relevance: This work is important because it demonstrates how heuristic search techniques can address warehouse relocation efficiently without requiring real-time learning. It serves as a strong baseline against which reinforcement learning approaches, like those in our project, can be compared — particularly in terms of solution quality and computational cost.

Scope Comparison: While Chen et al. focus on dynamic, operational-level relocation within a warehouse, our project expands the scope to full-layout optimization across zones and item categories. Their approach relies on handcrafted search strategies, whereas ours aims to use reinforcement learning to automatically learn relocation policies that balance density, accessibility, and movement cost, potentially achieving more general and adaptive outcomes.

4. Leon, J. F., Li, Y., Martín, X. A., Calvet, L., Panadero, J., & Juan, A. A. (2023). "A Hybrid Simulation and Reinforcement Learning Algorithm for Enhancing Efficiency in Warehouse Operations"
https://www.mdpi.com/1999-4893/16/9/408

Research Focus: This paper investigates how to combine simulation and reinforcement learning to improve storage location assignment and overall handling efficiency in warehouse operations. It addresses the challenge of variability in demand, storage and retrieval operations, and the interaction between different warehouse actors (e.g., workers, automated systems) by using a simulation-driven environment to train an RL agent.

Methodology: The authors build a simulation of warehouse operations that can model stochastic aspects (arrival rates, retrievals, re-slotting) and then embed a reinforcement learning agent to learn assignment policies for storage locations (i.e., where to place items) in that simulated environment. They compare learned policies with traditional heuristics under several scenarios, reporting performance improvements in throughput and space utilization.

Key Findings: Their experiments show that hybrid simulation + RL significantly outperforms heuristic methods for storage location assignment in terms of both operational efficiency (reduced movement and retrieval time) and space utilization. The study also reveals that incorporating simulation of stochastic demand and system interactions is important for RL to learn useful policies rather than overfitting to a static layout.

Relevance: This work is highly relevant for your project because your goal is precisely to optimize item relocation across warehouse layouts (balancing density vs movement cost) via RL. The fact that this study uses a simulation + RL framework for storage location assignment makes it a strong precedent. It indicates that simulation environments are viable and that RL can deliver meaningful gains in warehouse layout contexts.

Scope Comparison: Leon, J et al. (2023) focus on operational-level improvements in storage assignment and handling efficiency using hybrid simulation and learning. The approach is primarily concerned with optimizing storage and retrieval during active operations. In contrast, the current research direction emphasizes pre-operational, static warehouse configuration optimization, particularly the relocation and densification of items across multiple zones and categories. While both share the goal of improving spatial and operational efficiency, Leon, J et al. emphasize dynamic adaptability during execution, whereas the broader scope here involves structural optimization and planning prior to operational deployment.

5. Waubert de Puiseau, C., Nanfack, D. T., Tercan, H., Löbbert-Plattfaut, J., & Meisen, T. (2022).. "Dynamic Storage Location Assignment in Warehouses Using Deep Reinforcement Learning"

https://www.mdpi.com/2227-7080/10/6/129

Research Focus: This paper addresses the problem of dynamically assigning storage locations to items in warehouses using *Deep Reinforcement Learning (DRL)*. Traditional static slotting or rule-based assignment strategies fail to adapt to variations in product demand and order frequency. The study aims to minimize total handling cost by optimizing storage assignments that adaptively change over time as warehouse conditions evolve.

Methodology: The authors model the warehouse environment as a Markov Decision Process (MDP) and apply a *Double Deep Q-Network (DDQN)* algorithm. The state space includes current storage utilization, item categories, and demand frequencies. Actions represent the assignment of incoming items to specific storage zones, while rewards are designed to penalize travel distance and reward improved space efficiency. The DRL agent is trained in a simulated environment and benchmarked against heuristic rules like ABC classification and frequency-based allocation.

Key Findings: The DRL-based approach achieved significantly lower handling and travel costs compared to heuristic and conventional optimization methods. It demonstrated the ability to generalize across different warehouse demand patterns and remained robust under stochastic input conditions. Additionally, the adaptive learning capability allowed the system to respond efficiently to fluctuations in incoming item profiles.

Relevance: This work is directly relevant to our project as it uses deep reinforcement learning for optimizing storage decisions in a warehouse. While it focuses on dynamic storage assignment for incoming goods, the underlying learning principles—defining states, actions, and rewards—align closely with our goal of developing an RL-driven relocation optimization system.

Scope Comparison: Whereas Waubert et al. address *dynamic assignment* of new items to storage zones in an operational warehouse, our project extends the idea to *static relocation* of existing items across zones to improve spatial density and accessibility. Both approaches share the same optimization goal—minimizing travel and maximizing utilization—but differ in timing (real-time vs pre-optimization) and objective scope (assignment vs relocation).

6. Wei, L., Wei, F., Schmitz, S., & Kunal, K. (2021). "Optimization of Container Relocation Problem via Reinforcement Learning"

https://proc.logistics-journal.de/article/view/1027

Research Focus: This study explores the application of reinforcement learning to the *Container Relocation Problem (CRP)*—a logistical challenge analogous to item relocation in warehouses. The goal is to minimize the number of unnecessary container moves during retrieval operations in container terminals, which closely resembles minimizing redundant item relocations in dense warehouse environments.

Methodology: The authors design an RL-based decision model using a Q-learning algorithm where each state represents a container stack configuration, and actions correspond to moving a container to a different stack or slot. The reward function penalizes unnecessary relocations and incentivizes moves that reduce future obstruction. Simulation experiments were conducted to train the agent under various container yard configurations, and performance was compared against rule-based strategies such as Lowest Position First (LPF) and MinMax policies.

Key Findings: The reinforcement learning approach successfully learned optimal relocation policies that minimize unnecessary moves and retrieval delays. The RL model achieved a reduction in relocations by 15–25% compared to baseline heuristics. Additionally, it demonstrated good scalability across different yard sizes and stacking densities, validating its use for larger, real-world environments.

Relevance: This paper directly relates to the *relocation optimization* aspect of our project. Though the setting is a container yard rather than a warehouse, both domains share the same operational objective—optimizing movement efficiency and space utilization through learned decision-making policies.

Scope Comparison: While Wei et al. (2021) focus on optimizing relocations in container stacking systems, our work extends this to a general warehouse layout that includes multiple item types, zones, and storage constraints. Their study emphasizes reducing retrieval delay; ours focuses on balancing relocation cost and storage density through a unified RL optimization framework.

7. Cestero et al. (2022). "Storehouse: A Reinforcement Learning Environment for Optimizing Warehouse Management".

https://www.researchgate.net/publication/361900783_Storehouse_a_Reinforcement_Learning_Environment_for_Optimizing_Warehouse_Management

Research Focus: This paper introduces *Storehouse*, an open-source reinforcement learning environment designed to facilitate research in warehouse management problems. The environment provides a flexible simulation platform where RL agents can be trained for diverse warehouse tasks such as order picking, layout optimization, and task scheduling.

Methodology: The authors develop a modular simulation framework that represents warehouse operations through grid-based layouts, agent interactions, and task generation. They implement reinforcement learning agents using standard libraries like PyTorch and OpenAI Gym, enabling comparisons across multiple algorithms (e.g., DQN, PPO, and A3C). The study demonstrates how RL agents can learn efficient picking and placement strategies through continuous simulation feedback.

Key Findings: Storehouse successfully serves as a testbed for training and benchmarking RL algorithms in warehouse-related applications. Experiments revealed that agents trained within this environment could learn to minimize travel distance and improve order-picking efficiency. The framework's adaptability allows integration with different warehouse configurations and operational objectives.

Relevance: This study is crucial for our project because it validates the concept of using a *simulated warehouse environment* as a foundation for training RL agents. Our proposed system similarly relies on a synthetic grid-based warehouse model for training an agent to optimize relocation strategies.

Scope Comparison: While Cestero et al. aim to create a general-purpose RL simulation environment for warehouse research, our focus lies in developing a specialized RL agent within such an environment for item relocation optimization. Storehouse provides methodological groundwork—environment design, state encoding, and RL interfacing—on which our project's customized relocation model can be built.

8. Troch et al. (2023). "Solving the Storage Location Assignment Problem Using Reinforcement Learning"

https://www.researchgate.net/publication/373907319_Solving_the_Storage_Location_Assignment_Problem_Using_Reinforcement_Learning

Research Focus: This paper presents a reinforcement learning-based solution to the *Storage Location Assignment Problem (SLAP)*, a key challenge in e-commerce fulfillment centers. The objective is to determine optimal product placements within a warehouse to enhance retrieval speed and operational throughput.

Methodology: A simulated small-scale warehouse layout is constructed to train an RL agent responsible for deciding product placement. The environment models parameters such as SKU types, order frequency, and storage constraints. The RL agent learns placement policies using a Q-learning approach, where rewards correspond to reduced retrieval time and improved layout compactness. Performance is evaluated against rule-based baselines such as frequency-based and random allocation methods.

Key Findings: The RL-based approach improved warehouse efficiency by approximately 20% compared to traditional heuristic allocation. The model demonstrated the ability to generalize across product categories and warehouse configurations, outperforming static rule-based systems in adaptive scenarios. The results confirm that reinforcement learning can produce scalable and flexible storage policies.

Relevance: This paper directly supports our project's hypothesis that RL can optimize storage and relocation decisions. While the authors focus on initial item assignment, their findings provide strong evidence that RL frameworks can learn efficient spatial layouts—insight directly applicable to our relocation problem.

Scope Comparison: Troch et al. address the initial storage assignment process in warehouses, while our project emphasizes subsequent *item relocation* to enhance layout efficiency and accessibility. Both share the same core principle—using RL to optimize item positioning based on learned policies—but differ in operational stage (initial assignment vs post-assignment optimization).

9. Arslan, E. (2025). "Optimizing Human-Centric Warehouse Operations: A Digital Twin Approach Using Dynamic Algorithms and AI/ML".

https://dergipark.org.tr/en/pub/verimlilik/article/1524701

Research Focus: Integrate digital twin with AI/ML to optimize manual warehouse operations (picking, routing, slotting), focusing on reinforcement learning algorithms.

Methodology: 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.

Key Findings: Improved KPIs in warehouse management. Approx. 28.6% reduction in picking time; inventory turnover improved 20%; forecast accuracy rose from 85% to 92%; labor cost reduction in 15%.

Relevance: 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.

Scope Comparison: While this paper focuses on real-time operational optimization using Digital Twins and algorithm switching in human-centric warehouses, our project targets the structural optimization of warehouse layouts through reinforcement learning. In addition, our simulation incorporates dynamic and fluctuating demand profiles across different SKUs, introducing temporal complexity similar to real environments. The author emphasizes continuous execution-time adaptation, our work explores how a learned relocation policy can proactively densify and reorganize a warehouse layout under evolving demand patterns.

10. Matloff (2008). "Introduction to Discrete-Event Simulation and the SimPy Language".

https://heather.cs.ucdavis.edu/matloff/public_html/156/PLN/DESimIntro.pdf

Research Focus: This document introduces the core principles of discrete-event simulation and explains how system behavior can be represented through discrete changes in state. It highlights common use cases in logistics, manufacturing, and queueing systems. Special emphasis is placed on how events, processes, and resources interact over time to model realistic operational scenarios.

Methodology: The author adopts a process-oriented approach using Python's SimPy library, showing how generator functions represent concurrent activities. Examples illustrate scheduling events, handling waiting times and managing shared resources such as machines or workers. Various queueing patterns and delay mechanisms are demonstrated to model complex workflows.

Key Findings: SimPy enables modular and scalable simulation design, allowing researchers to emulate resource contention and time-dependent behavior. It supports priority queues, state monitoring, and controlled randomization for experimentation. This makes it a suitable framework for testing performance impacts of alternative process strategies.

Relevance:
Our project relies on SimPy to model warehouse operations such as retrieval, storage, and congestion under fluctuating demand. The process-based paradigm aligns well with real-world warehouse workflows, where events are triggered by item requests. This foundation supports RL training by providing a controllable, realistic environment.

Scope Comparison: Unlike our work, this document focuses on teaching simulation mechanics rather than optimizing warehouse layouts or decision policies. It does not incorporate intelligent control, dynamic demand adaptation, or machine learning-driven relocation strategies. However, it serves as reference to model diverse processes as discrete event simulations.

11. Torlaschi (2020). "Development of a Discrete-Event Simulation Tool with Genetic Algorithms for Automated Warehouses"

https://webthesis.biblio.polito.it/18180/

Research Focus: It explains the core principles of discrete-event simulation and how system behavior can be represented through discrete changes in state. It highlights common use cases in logistics, manufacturing, and queueing systems. Special emphasis is placed on how events, processes, and resources interact over time to model realistic operational scenarios.

Methodology: The author builds a SimPy-based simulator capable of modeling storage technologies such as AS/RS, shuttles, and lifts. A genetic algorithm is then applied to explore large parameter spaces, evaluating candidate configurations based on user-defined fitness metrics. The optimization iteratively evolves solutions to approximate globally optimal warehouse designs.

Key Findings: The simulation reveals trade-offs between concurrency, energy consumption, and system cost depending on the selected storage technology and configuration. Results show that multi-objective optimization can improve efficiency beyond traditional heuristic design methods. The tool demonstrates flexibility in evaluating diverse operational scenarios.

Relevance: This work reinforces the value of simulation-driven optimization when analytical approaches fail to account for concurrency and system stochasticity. It also demonstrates how simulation outputs can guide strategic infrastructure decisions. The general approach aligns with our intention to evaluate layout performance numerically before operational deployment.

Scope Comparison: While Torlaschi optimizes static warehouse design parameters, our project focuses on dynamic relocation strategies responding to shifting SKU demand. Instead of genetic search, we employ reinforcement learning to learn policies from sequential experience. This addresses operational adaptability rather than infrastructure selection.

12. Duque-Jaramillo et al. (2024). "Warehouse Management Optimization Using a Sorting-Based Slotting Approach"

https://www.jiem.org/index.php/jiem/article/view/5661

Research Focus: This paper studies how slotting decisions influence warehouse efficiency by integrating SKU physical characteristics, layout constraints, and heterogeneous demand patterns. It highlights the cost contribution of slotting and order picking and emphasizes the need for methods that jointly consider activities to reduce travel time and improve throughput.

Methodology: The authors develop a mixed-integer linear programming (MILP) model combined with a two-phase algorithm. In phase one, SKUs are prioritized using ABC and Cube per Order Index (COI) classification. In phase two, total operation times are calculated under multiple slot assignment sequences and compared through simulation using demand data from a distribution center. The Traveling Salesman Problem (TSP) heuristic is used as a benchmark.

Key Findings: Simulation results show that combining priority-based sorting with specific slot assignment sequences significantly reduces total travel time. The best performance is obtained when assigning rows and columns early in the sequence, which place high-priority SKUs closer to entry/exit points. Differences between ABC and COI strategies become more pronounced when demand is heterogeneous, especially in A and B categories, implying a need to periodically reevaluate allocations

Relevance: This work reinforces the importance of demand-aware slotting policies and shows that even small changes to allocation strategies can accumulate into significant time savings. Its demand-driven dynamics, priority-based allocation, and simulation validation directly relate to our project's focus warehouse layouts under shifting SKU popularity using RL.

Scope Comparison: While this study analyzes priority-based slotting through linear optimization and TSP-inspired heuristics, it does not explore adaptive relocation policies or sequential learning. Our reinforcement-learning approach builds on this foundation by dynamically adjusting item placement under evolving demand, learning policies iteratively instead of optimizing a single static assignment.

13. Mojumder & Nuruzzaman (2025). "AI-Driven Optimization of Warehouse Layout and Material Handling: A Quantitative Study on Efficiency and Space Utilization".

https://rast-journal.org/index.php/RAST/article/view/21/21

Research Focus: This study investigates how AI-based layout and material-handling optimizations can improve warehouse space utilization and operational throughput, particularly under constraints of storage capacity, picking flow, and handling equipment.

Methodology: The authors conduct a systematic review following the PRISMA framework, synthesizing findings across multiple warehouse functions. They comparatively analyze AI techniques—including supervised learning, reinforcement learning, deep learning, and hybrid systems—based on measured operational outcomes such as cycle time, storage density, congestion, and picking efficiency. A quasi-experimental design is used to compare AI-enabled configurations against traditional baseline strategies

Key Findings: AI-based optimization consistently reduces cycle times by 15–45%, increases volumetric space utilization by 20–35%, and improves order accuracy to above 98%. Techniques such as Deep Q-Learning and Actor-Critic models are highlighted for dynamic layout reconfiguration, while clustering methods enhance SKU affinity zoning. The study identifies gaps including lack of real-time benchmarking, limited cross-industry generalization, and data availability constraints.

Relevance: This work directly supports our project by demonstrating that AI-driven slotting and adaptive layout decisions yield significant improvements in throughput, space utilization, and congestion reduction—especially under dynamic demand, heterogeneous SKUs, and high item velocity. It strengthens the case for using reinforcement learning to optimize relocation policies rather than relying on static heuristics.

Scope Comparison: Compared to this broad, cross-domain synthesis, our project narrows the focus to simulation-based reinforcement learning for item relocation within a defined warehouse grid. Instead of industry-wide analysis, we produce an applied prototype environment where an RL agent learns sequential re-slotting under fluctuating demand, addressing one of the empirical gaps identified: the lack of granular, scenario-based validation.

14. Oliveira et al. (2022). “Efficient Multi-Constrained Task Allocation in Smart Warehouses”.

https://assets-eu.researchsquare.com/files/rs-1720697/v1_covered.pdf

Research Focus: This study tackles the complex problem of assigning tasks to a diverse fleet of warehouse robots. The goal is to intelligently decide which robot should pick up which item and deliver it where, considering real-world constraints like their different speeds, carrying capacities, and starting positions.

Methodology: The authors developed a smart algorithm called DoNe-CPTA. It works by dividing the warehouse into optimized "zones" for each robot based on its capabilities, using a technique similar to creating territories on a map. The algorithm is designed to avoid sending multiple robots to the same delivery point and dynamically adjusts its calculations based on each robot's current load and speed.

Key Findings: When tested in simulations, their method significantly outperformed existing approaches. It achieved a 33% reduction in total travel distance, computed the task assignments 96% faster, and in some scenarios, required 18% fewer robots to complete the same work.

Relevance: This research is a great example of solving a complex, constrained optimization problem in warehouses. While it focuses on robot coordination, its core ideas—like intelligent spatial partitioning and dynamic cost modeling—are directly useful for building a system that learns to optimize item placement and movement.

Scope Comparison: Oliveira's team concentrates on **task allocation** (who does what, where). Our research focuses on **item relocation** (what goes where, when). Both aim to boost warehouse performance, but they address different parts of the puzzle. Their framework can inform how we model decisions and costs in our system.

15. Viveros P. et al. (2021). “Slotting Optimization with Divisible Storage Locations”.

https://www.mdpi.com/2076-3417/11/3/936

Research Focus: This paper addresses the storage location assignment problem with a novel twist: making the bottom-level storage slots "divisible," meaning they can be split to hold multiple different SKUs. The goal is to simultaneously maximize space utilization and minimize picking travel time.

Methodology: The authors developed a Mixed-Integer Linear Programming (MILP) model. To manage the complexity, they broke the problem down into four sequential sub-problems: deciding which slots to divide, assigning product families to aisles, choosing specific divisible slots, and handling floor storage.

Key Findings: Their optimized model using divisible slots increased storage capacity by over 18%. By strategically placing high-demand items in full-sized, easy-to-reach slots, they also significantly reduced the travel distance and cost for warehouse operations.

Relevance: This work is directly relevant as it dives deep into the trade-offs between storage density and retrieval efficiency. The concept of "divisible slots" is a powerful idea we can incorporate into our own models to define the state and action space for relocation.

Scope Comparison: The main difference is the optimization approach. This paper uses a **static, one-time mathematical model** for slotting, while our research uses **RL for dynamic, ongoing relocation**. We share the same core goals (density and efficiency), but our method is designed to adapt to changing conditions over time.

VI. Conclusion

The literature indicates that warehouse optimization is shaped by both structural factors, such as layout and storage allocation, and operational dynamics, including demand variability and relocation sequences. While heuristics and metaheuristics provide baseline strategies, reinforcement learning emerges as a promising approach for developing adaptive, data-driven policies that can respond to dynamic conditions, improve throughput, and minimize unnecessary movements. The studies reviewed underscore the importance of carefully defining state representations, action spaces, and reward functions to capture both efficiency and practical constraints.

Looking forward, the simulation and modeling phase will need to address challenges such as scalability, stochastic demand, and realistic representation of warehouse operations. Insights from previous studies suggest that reward design, environment fidelity, and computational constraints will be key factors influencing performance. By anticipating these challenges, the next stage can focus on ensuring that RL agents are capable of learning robust relocation strategies that balance operational efficiency, spatial utilization, and adaptability under evolving conditions.