Case Study: Intelligent Systems in Production

RL-Driven Warehouse Item Relocation Optimization

Prof. Hamidreza Heidari

Binil Sajeev, Abdelraheem Zekry, Orlando Manrique, Arthur Reis

1 Topic & Overview

- Project Title: **RL-Driven Warehouse Item Relocation Optimization**
- Team:
  - Binil Sajeev, Abdelraheem Zekry Reinforcement Learning & Optimization
  - Orlando Manrique Geometric Modeling & Simulation
  - Arthur Reis Data Engineering, Analytics & Validation
- Industrial Problem: **Limited space and dynamic inventory make manual warehouse allocation impossibly complex, leading to congestion and inefficiency.**
- Goal: **Develop an RL-based system to optimize item relocation strategies, maximizing storage density while minimizing movement costs.**

2 Input Parameters / Data

Data Source:

Inventory: Synthetic generation based on confidential automotive supply chain statistics (Privacy-Preserving).

Topology: Procedurally generated hybrid layout (Block + Container logic).

Parameters - Warehouse Layout:

Coordinate System: Cartesian (*x, y, z*) in mm.

Rack Logic:

Max Rack Height: 2,200 mm (Constraint).
Aisle Width: 1,200 mm; Rack Spacing: 50 mm.
Organization: Segregated by Location Type (bin dimensions)

Storage Configuration:

Back-to-Back, multi-level shelving.
Address Label: Row-Bay-Level (e.g., R01-B001-L01).
Storage Policy: Single-SKU per bin (No mixed storage).

Size:

Prototype: 50 unique items.
Target(Industrial level): ~70,000 unique items.

Inputs and Units:

1. Inventory & SKU Specifications (Parts Data)

Physical Geometry:

Dimensions: Length, Width, Depth (mm).
Volume: Unit & Aggregate requirement (mm^3).
Weight: Mass per unit (kg).

Supply Chain State:

Stock Quantity: Number of containers/boxes on hand (Integer).
Throughput Profile: Demand (Normalized Intensity Score).

2. Warehouse Topology & Bin Profile (Location Data)

Bin Capacity:

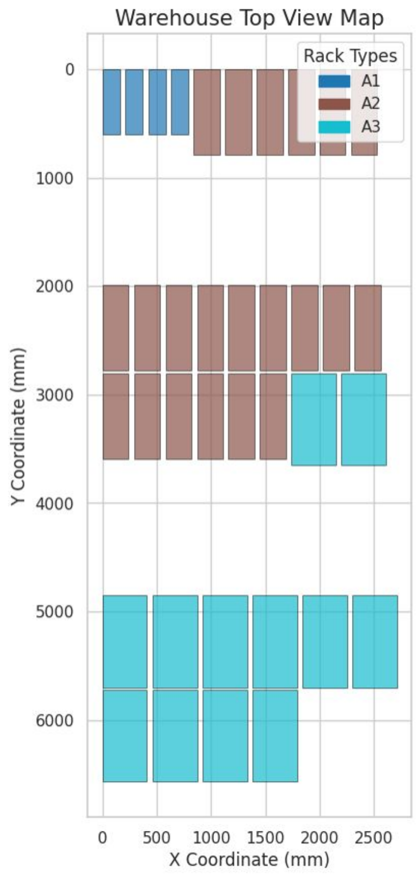
Internal Dimensions: Width, Depth, Height (mm).
Volumetric Limit: Maximum usable space (mm^3).

Spatial Definition:

Global Position: Cartesian coordinates in the facility (mm).
Logical Address: Row, Bay, and Level indices.

Heuristic Metrics:

Accessibility Rating: Calculated score representing ease of picking (e.g., ground level vs. top shelf).



2 Input Parameters / Data

Synthetic Dataset Generation Methodology:

Original Dataset (Confidential) → Summary Statistics → Synthetic Data Generation → Validation

Parts
- Dimensions
- Weight
- Initial inventory
- Demand/year

Locations
- 3d Coordinates
- Bin Dimensions

Allocations
- Allocated Part ID
- Allocated Qty
- Location

Descriptive Statistics

Mean, Median, Std Dev, Min/Max for all variables.

Discretization

Length segmented into Bins (captures non-linear shapes).

Conditional Profiling

Width/Depth/Weight distributions per Length Bin.

Ratio-Based Constraints

W/L, D/W, Weight/Vol ratios

Probabilistic Bin Selection

Select Length Bin (weighted by frequency).

Sample dimensions uniformly within bin limits.

Sample Inventory/Demand from Log-Normal distributions.

Constraint Application

Enforce Ratio Logic (e.g., Width must be 40–60% of Length).

-Geometric Check: Vpart<= Vbin

Procedural ID Generation

Assign new unique IDs

Aggregate Comparison

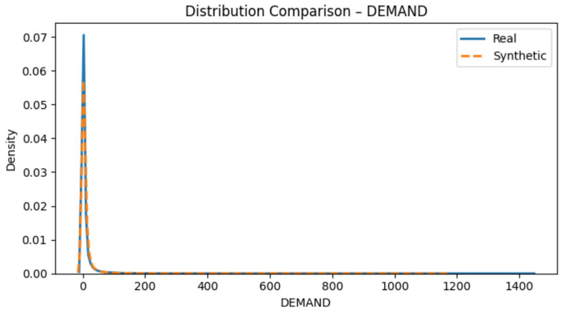
Synthetic ≈ Source Histograms.

Correlation Verification

Pearson matrix check (e.g., Volume vs. Weight)

Geometric Sanity

0% of items generated are physically impossible



3 Algorithm + Simulation Flowchart

Core Logic: Simulate a warehouse allocation problem using **three progressively smarter strategies.**

Baseline (Chaotic Allocation) – simulates poor manual placement

- Randomized SKU and bin selection (simulates manual, unplanned storage)
- Geometry-aware placement (checks all 6 box orientations)
- Single-SKU per bin, but no demand or accessibility logic
- Boxes placed until bin capacity is reached, then move to next bin
- Results in high bin fragmentation and poor space utilization
- Serves as a worst-case reference baseline

Heuristic Allocation – rule-based optimization

- SKUs sorted by descending demand
- Geometry-aware bin filtering (valid physical fit only)
- Bin selection based on combined score:

➔ Accessibility (PICK_SCORE)

➔ Space utilization (fill ratio)

- Greedy placement: fills best bins first until SKU stock is exhausted
- Improves utilization and reduces used bins vs baseline
- Fixed rules → no learning or adaptation

Reinforcement Learning Allocation – adaptive learning-based optimization

- Formulated as a reinforcement learning problem (Q-learning)
- Agent learns optimal bin choice per SKU through reward feedback
- Reward balances:

➔ Physical feasibility

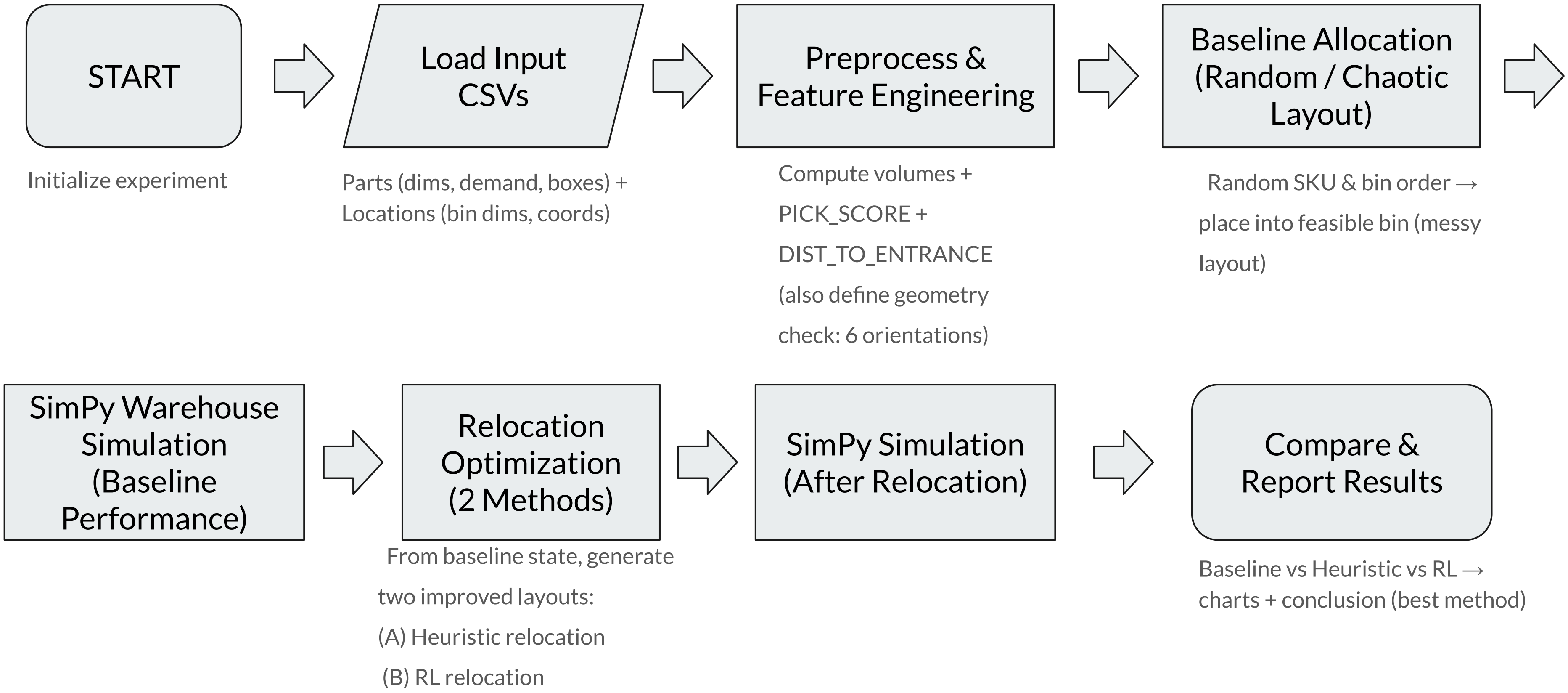
➔ Space utilization efficiency

➔ Demand-weighted accessibility

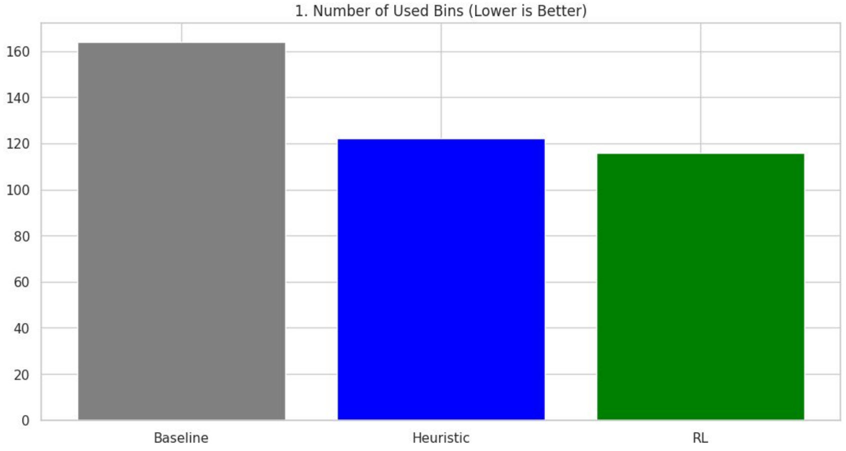
- Learns non-linear tradeoffs ignored by heuristics
- Achieves highest average utilization and fewer bins
- Slightly higher travel distance due to utilization-first strategy

4 Algorithm + Simulation Flowchart

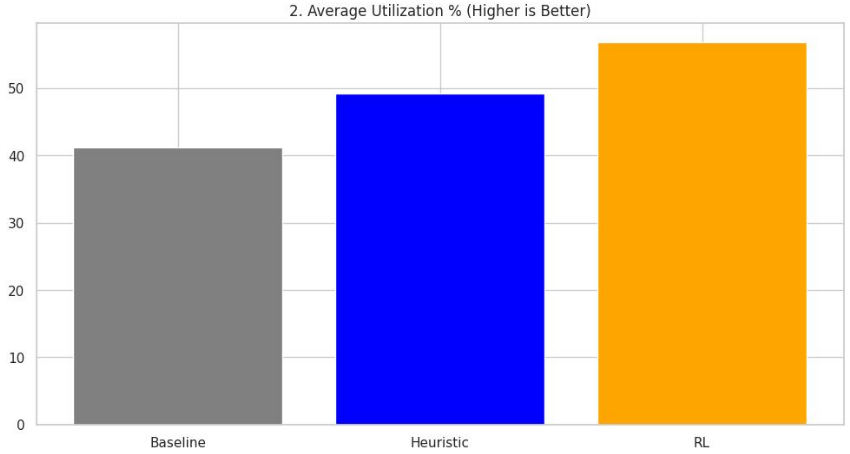
Overall Simulation & Optimization Flowchart:



5 Output & Results

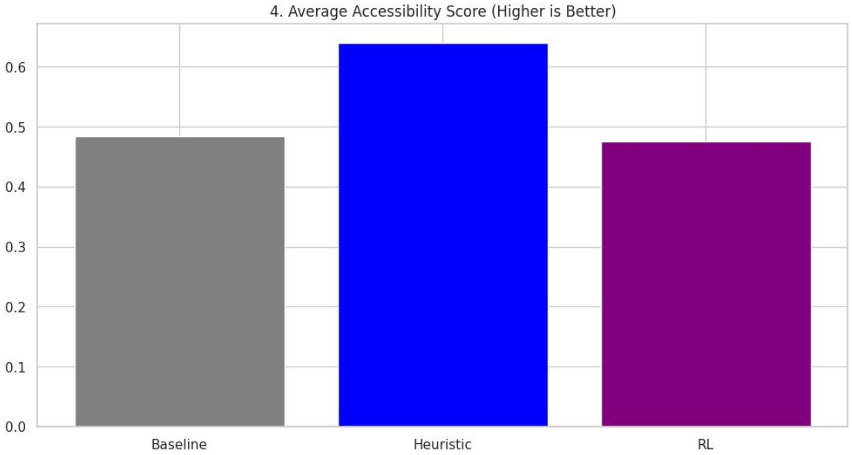


- **Baseline** uses the **highest number of bins** due to random, unstructured placement.
- **Heuristic method** significantly reduces bin usage by demand-aware, geometry-feasible packing.
- **RL approach** achieves the **lowest bin count**, learning compact placement strategies.
- Fewer bins directly imply **better space consolidation** and lower operational footprint.

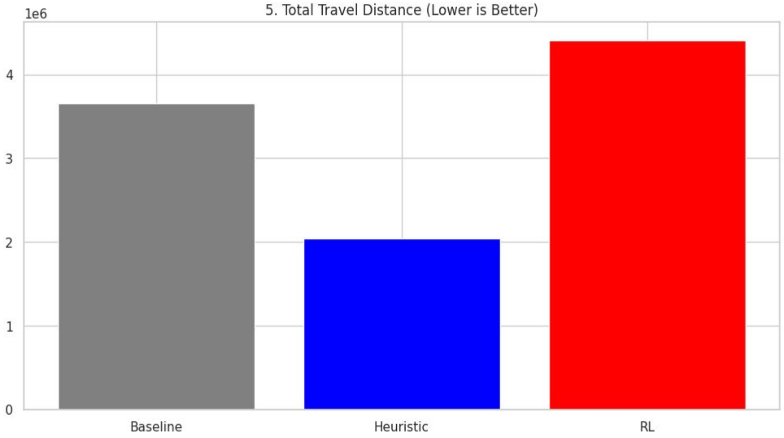


- **Baseline utilization is lowest**, indicating inefficient space usage.
- **Heuristic allocation** improves utilization through greedy filling of suitable bins.
- **RL achieves the highest utilization**, favoring near-perfect bin filling.
- Higher utilization reflects **denser storage and reduced wasted capacity**.

5 Output & Results



- **Baseline accessibility is moderate** due to random placement.
- **Heuristic method achieves the highest accessibility**, prioritizing high-demand items near entrance.
- **RL slightly sacrifices accessibility** to optimize overall space utilization.
- Demonstrates **trade-off between accessibility and compactness**.



- **Baseline travel distance is high**, caused by chaotic item placement.
- **Heuristic method minimizes travel distance**, strongly favoring demand-weighted accessibility.
- **RL shows higher travel distance** due to aggressive space compaction.
- Highlights differing optimization priorities: **movement efficiency vs storage density**.

The results show a trade-off where the heuristic method excels in accessibility and travel distance, while the RL approach achieves superior space utilization and bin consolidation.

6 Comparison

Reference Paper

Kim, J., Méndez, F. A., & Jimenez, J. A. (2020). Storage location assignment heuristics based on slot selection and frequent itemset grouping for large distribution centers. IEEE Access

https://scispace.com/pdf/storage-location-assignment-heuristics-based-on-slot-b1u5mic59s.pdf

- Greedy heuristics define only one priority rule, e.g. Manhattan distance, first available slot, product turnover, size etc. RL can adapt the number of parameters according to the complexity of the problem, e.g. dimensions, orientations, distances, etc. All of them combined.
- Allocations are done based on that rule or at random. It just takes the best possible decision at the given time, no consideration for past or future events.
- On this paper, itemset frequency is considered while our work treats each each SKU individually.
- The greedy algorithm is suitable for initial allocations but not feasible for dynamic relocation processes.

7 — Validation / Proof

Validation Strategy

Comparative Benchmarking: The RL model was validated against two control scenarios using identical datasets:

Baseline (Chaotic): Random allocation to simulate unorganized operations.

Heuristic (Greedy): Standard "Best Fit" logic to simulate traditional WMS algorithms.

Physical Constraint Enforcement: Results were validated through a hard-coded physics engine (calculate_max_stack_count) ensuring every allocation respects geometric dimensions, and bin volume limits preventing "impossible" storage.

Convergence Monitoring: Model reliability was proven by tracking reward stability over 800 training episodes (Chart 3), confirming the agent transitioned from exploration (randomness) to exploitation (optimized strategy).

Why this Validation is Reliable

Deterministic Feasibility: Unlike theoretical models, this simulation rejects any placement that violates physical bin boundaries, guaranteeing 100% executable allocation plans.

Metric Alignment: The model was scored against conflicting real-world KPIs—simultaneously optimizing for Space (Utilization %) and Speed (Distance/Pick Score)—proving robustness across multiple operational goals.

Data Examples

Geometric Constraint Validation ("Physics Engine")

We enforced strict collision detection to ensure items physically fit.

Example: **Part 75003920 (Length: 526mm)** was correctly rejected from **Type A1 bins (Width: 160mm)** and successfully validated only in **Bin A2-00002 (Type A2)**, proving the constraint logic works.

Demand Sensitivity Checks ("Logic Validation")

We Cross-referenced demand data against distance metrics to ensure the model penalizes inefficiency.

Example: **High-demand Part 14891290 (Demand: 97 units)** was flagged during validation for being placed in **Bin A2-00137 (5,340mm from entrance)**, confirming the model correctly identifies negative reward scenarios for fast-movers.

Utilization Accuracy ("Math Verification")

We manually verified volume calculations to ensure density rewards were accurate.

Example: **Part 40459274** in **Bin A3-00042** was calculated at **74.31% utilization**. We validated this against the raw **dimensions (2 boxes × 33.6L volume ÷ 90.6L bin capacity),** ensuring the agent optimized for real-world density.

8 — Discussion, tangible outputs and gaps

- Heuristics and traditional methods can be effective when the warehouse considered static conditions, taking into account dimensions and item classification.
- RL Agents enhance productivity of warehouses in dynamic and complex scenarios.
- In comparison with heuristics, the RL agent has slightly better performance than the Greedy Heuristic. The model should be trained with a large scale dataset and tested on a more complex environment to justify its use in comparison to traditional heuristics.
- The learning process is still too simple, more parameter should be included to be able to generalize to a larger variety of warehouse layouts and conditions.
- Long term simulation is still required to validate results.

Thank you

Binil Sajeev, Abdelraheem Zekry, Orlando Manrique, Arthur Reis