The Pennsylvania State University

The J. Jeffrey and Ann Marie Fox Graduate School

Multi-Objective Warehouse Allocation Using Non-Preemptive Goal Programming

A paper in

Industrial Engineering and Operations Research

by

Balaji Adithya Dhandapani Shanmugasundaram

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2025

The paper of Balaji Adithya Dhandapani Shanmugasundaram was reviewed and approved by the following:

Paul Griffin
Professor
Thesis Advisor

Anirudh Subramanyam
Assistant Professor
Reader

1 Table of Contents

2 List of Figures

3 List of Tables

4 Acknowledgements

I would like to express my sincere gratitude to my advisor, Prof. Paul Griffin, for his unwavering support, guidance, and encouragement throughout this research. His expertise in warehouse allocation was pivotal in shaping both the direction of this work and my interest in the subject.

I am deeply thankful to Prof. Anirudh Subramanyam for his practical and engaging instruction in Linear Programming. His mentorship in translating real-world problems into structured mathematical models—particularly through Mixed Integer Linear Programming—provided a strong analytical foundation for this study.

I also wish to thank Andrea Jankovic, Director of Supply Chain at Alstom, for approving the project and facilitating access to essential data and organizational resources. Finally, I am grateful to Zoe Yarsky, Logistics Engineer at Alstom, for her detailed walkthrough of warehouse operations and the valuable practical insights she shared, which significantly enriched the applied aspects of this research.

5 Abstract

Warehouse slotting—the assignment of stock-keeping units (SKUs) to storage locations—has a pronounced impact on order-picking cost, service level, and space utilization. This paper presents a weighted-goal programming (GP) model that simultaneously minimizes (i) hand-picking travel distance, (ii) forklift travel distance, (iii) unsafe elevations for hand-pickable items, and (iv) elevation of heavy parts, while ensuring complete allocation of demand. The model incorporates physical bin–part compatibility, stackability, part demand, and warehouse geometry. A case study using proprietary data from the transportation industry demonstrates the model's ability to allocate parts for given inventory. The formulation is implemented in Pyomo and solved with CBC.

5.1 Keywords

Warehouse slotting; storage location assignment problem; goal programming; Pyomo; mixed-integer programming; supply-chain optimization

6 Introduction

Warehouses account for up to 55 % of total logistics costs, with order-picking often representing the single largest cost driver (Viveros *et al.*, 2021). Strategic placement of inventory—*slotting*—is therefore critical. Traditional approaches rely on heuristics such as ABC or Cube-per-Order-Index (COI), but these seldom capture the multi-objective nature of modern facilities that handle mixed hand-picking and forklift zones, varied product geometries, and safety constraints.

Goal programming (GP) offers a flexible mechanism to reconcile conflicting criteria by minimizing weighted deviations from pre-specified targets. Despite its long history in multi-criteria decision-making (Tamiz *et al.*, 1995), applications of GP to slotting remain sparse. This paper contributes a comprehensive GP formulation that integrates spatial, ergonomic, and operational factors within a single optimization model and validates it on an industrial data set.

7 Literature Review

7.1 Classical formulations of the Storage-Location Assignment Problem (SLAP)

Early SLAP research concentrated on single-objective formulations that minimize picker travel distance or maximize space utilization. Comprehensive surveys by De Koster et al. and later Staudt et al. show that most models before 2018 relied on ABC, COI or pure distance heuristics, and were usually solved by exact MILP or constructive heuristics. Recent contributions continue to refine exact methods: for example, Wang & Voudouris (2024) propose a two-step MIP/heuristic hybrid that solves instances with thousands of SKUs.

7.2 Layout-driven extensions: divisible slots, multilayer racks and heterogeneous zones

Modern facilities rarely have homogeneous rectangular slots. Viveros et al. (2021) introduced a slotting model that lets first-level pallet locations be divided into sub-slots so that small SKUs can share prime positions; their MILP reduced travel by 12% in a food DC. Other studies embed rack merging, column widening or rack-height decisions directly into the assignment model, but they still optimize a single cost metric, leaving ergonomic aspects aside.

7.3 Multi-objective and ergonomic perspectives

Growing attention is being paid to worker safety and sustainability. Sun et al. (2021) quantify trunk flexion and shoulder EMG at seven shelf heights and derive placement zones that lower cumulative spinal load by 35%. Likewise, Diefenbach et al. (2024) propose a bi-objective model that trades off labor cost against joint-load indices; their weighted constraint approach shows that a 5 % cost increase can halve ergonomic risk. These studies confirm that distance-only objectives overlook substantial human factors.

7.4 Goal programming (GP) for multi-criteria warehouse decisions

GP provides a transparent way to reconcile conflicting criteria by minimizing weighted deviations from targets. Foundational reviews by Tamiz, Jones & Romero (1998) document its versatility across logistics and manufacturing. In warehousing, however, GP has mostly been applied to pallet-stacking or space-balancing tasks—for example, Perera et al. (2022) and Nguyen et al. (2021) use weighted GP to align pallet counts with dock capacities and shift schedules. Dedicated GP formulations for slotting remain scarce; none simultaneously capture travel, reachability and weight-elevation trade-offs.

7.5 Research gap

The literature survey reveals the following blind spots:

1. GP-based slotting models. Despite GP's success in other warehousing problems, no published work embeds more than two operational goals in a single slotting formulation like Viveros et al., 2021 and Rahman, 2019 which only considered distance metrics. Also, Joint consideration of ergonomic reach limits and heavy-item elevation was not done. Existing ergonomic models focus on either of the above factors but ignore the combined effect (Zangaro et al., 2019).
2. Unified treatment of manual and mechanized zones. Most studies split hand-picking and forklift areas, which masks cross-zone trade-offs.

The weighted GP model developed in this study addresses these gaps by:

1. A weighted-GP formulation that captures these four goals while enforcing one-SKU-per-bin and capacity rules.
2. An algorithmic procedure to generate bin coordinates, interpolated travel distances, and per-pair capacity limits, enabling realistic physics-based feasibility checks.
3. An open Python/Pyomo implementation accompanied by a case-study evaluation.

8 Problem Description

The warehouse consists of multiple runs, bays, levels, and bins. Except for run all the mentioned terminologies refer to their connotations in traditional warehouse language. Run refers to a linear section of storage.

Each bin is characterized by length *L*, width *W*, height *H*, and a pair of travel distances from each end of the run to hand-picking (HP) and forklift (FL) entry points.
Parts are described by dimensions, weight, quantity, stacking flag (indicates whether the part can be stacked), and hand-pickability flag (indicates whether the part can be handpicked by workers). This modeling framework specifically targets the allocation of large, bulky parts—such as door frames, wheels, and panels—that are typically stored directly on warehouse floors or racks without the use of containers, unlike smaller items such as screws or bolts that require binning. In this study, the term bulky refers to parts that are physically large in at least one dimension (length, width, or height) and not suitable for storage in standard containers or bins. The task is to assign the required number of units of every part to feasible bins, respecting:

- **Geometric feasibility:** part footprint and height must fit the bin, accounting for rotation.
- **Stackability:** non-stackable items may occupy only a single layer.
- **One-SKU-per-bin:** each bin stores at most one-part type.
- **Complete allocation:** all units must be placed in a bin.

Four *soft* goals guide the assignment:

Goal	Description	Deviation Variable
G1	Minimize HP travel distance, weighted by pick frequency	d_1^+
G2	Minimize FL travel distance for non-HP items, weighted by pick frequency	d_2^+
G3	Minimize elevation above 1.6 m for HP items	d_3^+
G4	Minimize elevation of heavy items (weight × elevation)	d_4^+

9 Mathematical Model

The solution consists of two stages (i) Feasible part-bin-quantity set generation (ii) Optimal part-bin allocation. The first step was carried out with conditional logic in Python i.e. by considering the bin and part dimensions, the maximum number of parts that fit in each bin is generated. Let P be the set of parts, B the set of bins, and $(\mathrm{p}, \mathrm{b}) \in \phi$ the set of feasible part–bin pairs.
The second stage consists of selecting the most optimal subset of the part-bin pairs from the feasible set, which was performed with Mixed Integer Linear Programming (MILP).

9.1 Decision Variables

- x_{pb} number of units of part p in bin b (integer)
- y_{pb} binary indicator (1 if part p occupies bin)

9.2 Parameters

Deviation variables $d_1^+, \ldots, d_4^+$ represent over-achievement of each goal. There are no underachievement deviation variables as all the terms in the expression are positive. Parameters include per-pair capacities C_{pb}, pick frequencies f_p , weights w_p , elevations h_b , travel distances $\delta_b^{HP}, \delta_b^{FL}$, and priority weights $w_1, \ldots, w_4$. h_{max}^{HP} is the maximum height at which handpickable items can be placed. $\chi_p^{HP}$is the indicator parameter that determines whether a part is handpickable. β parameter indicates the penalty for assigning new bins.

9.3 Constraints

1. **One SKU per bin** $\sum_{p:(p,b)\in\Phi} y_{pb} \leq 1 \qquad (\forall b)$
   Ensures each bin holds only a single SKU to avoid mixing items and simplify picking.

2. **Capacity linking** $x_{pb} \leq C_{pb} . y_{pb} \qquad (\forall (p,b) \in \Phi)$

Prevents exceeding bin-specific capacity and activates only if the bin is assigned.

3. **Demand satisfaction** $\sum_{b\in B} x_{pb} = D_p \qquad (\forall p)$

Guarantees that all required units of each part are allocated somewhere.

9.4 Goal Equations

1. **Hand-picking distance**

$$\sum_{(p,b)\in\Phi} f_p \cdot \delta_b^{HP} \cdot x_{pb} - d_1^+ = 0$$

Minimizes total travel distance for hand-pickable item, giving priority to frequency of picks

2. **Forklift distance**

$$\sum_{(p,b)\in\Phi} f_p \cdot \delta_b^{FL} \cdot x_{pb} - d_2^+ = 0$$

Minimizes total travel distance for items that require forklift handling, giving priority to frequently.

3. **Reachability**

$$\sum_{(p,b)\in\Phi} x_{pb} \cdot \max(0, h_b - h_{max}^{HP}) \cdot \chi_p^{HP} - d_3^+ = 0$$

Penalizes placing hand-pickable items above ergonomically reachable height.

4. **Heavy-weight elevation**

$$\sum_{(p,b)\in\Phi} x_{pb} \cdot h_b \cdot w_p - d_4^+ = 0$$

Discourages placing heavy items on high levels, promoting safety and ease of access.

9.5 Objective

$$\min\left(w_1 d_1^+ + w_2 d_2^+ + w_3 d_3^+ + w_4 d_4^+ + \beta \cdot \sum_{(p,b)\in\Phi} \chi_{pb}\right)$$

Minimizes the weighted deviations from all goals while penalizing the use of additional bins.

10 Model Implementation

The workflow below summarizes the working of the program:

- Read part and layout spreadsheets
- Interpolate column-wise travel distances
- Generate Bins with given Run dimensions and compute coordinates/elevations
- Alter bin width as per the user input to allow for protruding parts
- Merge bins as per user input to allow for storage of longer parts
- Enumerate feasible (p,b) pairs and capacities by finding the number of parts that can be allocated by choosing the maximum from lengthwise and widthwise placement for every part bin combination.
- Build the Pyomo model by constraints and goal equations using deviation variables
- Solve with CBC (or Gurobi for larger instances)
- Obtain the most optimal set of bin-part-quantity combinations

11 Case Study

To evaluate the proposed goal programming model, it was applied to a real-world industrial warehouse operated by Alstom in Pittsburgh. The input data comprised detailed part specifications and the corresponding warehouse layout. The layout was designed based on the operational constraints imposed by the existing handling equipment, specifically the Hyster N30ZDRS2 stand-up forklift.

Key forklift specifications include:

Fork Length	= 1.06m
Maximum Horizontal Extension	= 1.06m
Maximum Run Width	= 2.12m
Maximum Elevation of highest level	= 7.89m

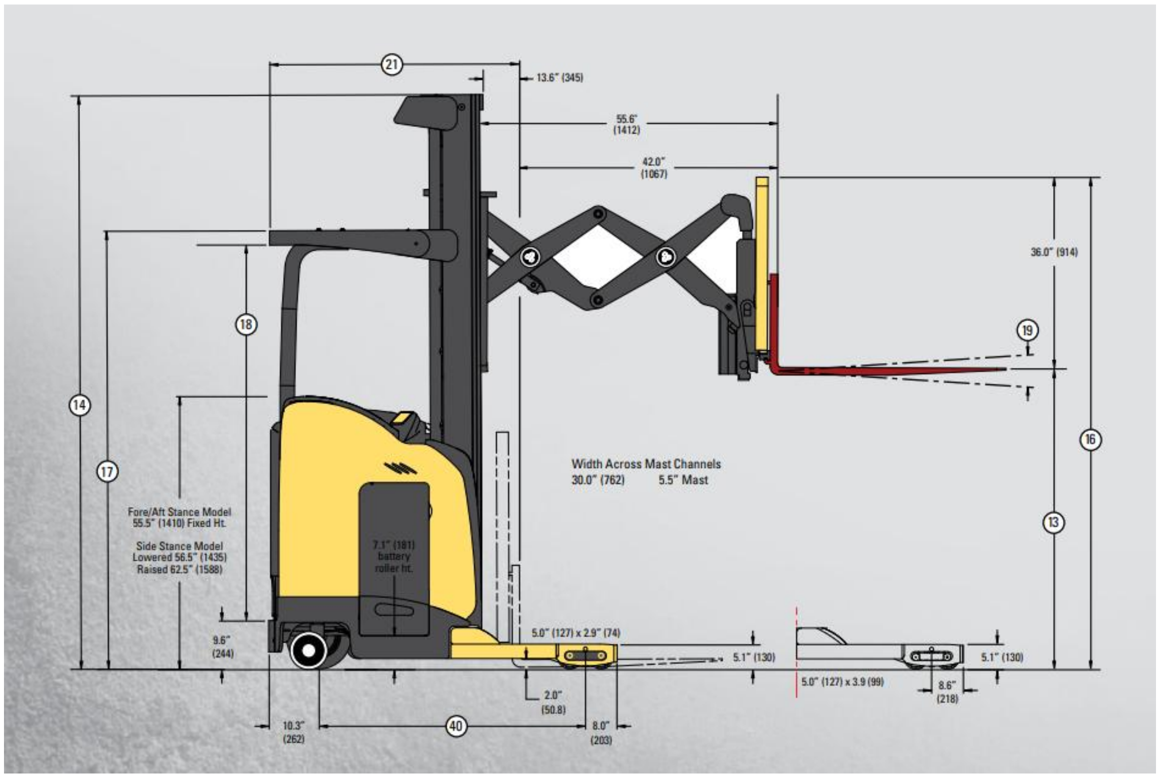

Figure 1. Side view of Stand-up Forklift

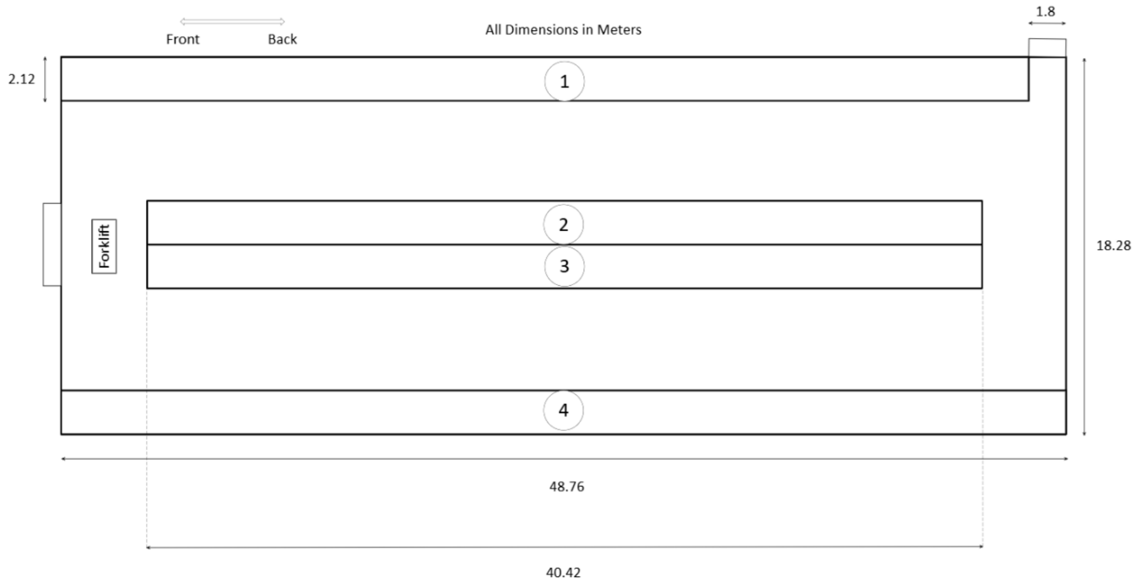

Figure 2. Warehouse Layout

The table below displays the specifications of the Warehouse layout

Table 1: Warehouse Layout Dimensions

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	7	54	50	3	47	2.1	4	2	3	12	3	3.9	5
2	6.3	49	49	8.1	41	2.1	3	2	3	14	3	4.5	5
3	6.3	49	51	10	41	2.1	3	2	3	14	3	4.5	5
4	7	56	66	17	49	2.1	3	3	3	16	3	5.4	8

Table 2: Warehouse Layout Table Column Descriptions

Column Name	Description
A	Run Number
B	Run Distance from Front to Entrance: FL door
C	Run Distance from Back to Entrance: FL door
D	Run Distance from Front to Entrance: HP door
E	Run Distance from Back to Entrance: HP door
F	Run Length
G	Run Width
H	Number of Bays in Run
I	Number of Levels in Run
J	Height of each Level
K	Length of each level in Bay
L	Number of Bins in each Bay
M	Length of each bin
N	Elevation of last level from ground

The run width in the model was determined by assuming the forklift must access the entire run's width. A warehouse configuration with four uniformly wide runs was selected for the case study. For spatial optimization, Manhattan distance metrics were employed to approximate traversal distances within the warehouse layout.

Table 3: Case Study Model Weights

Symbol in Pyomo Model	Description	Weight
w1	Minimizing Handpicking Distance	1000
w2	Minimizing Forklift Travel Distance	100
w3	Keep Handpickable items below safe reach	200
w4	Minimize the elevation of heavy weight items	1000
BIN_PENALTY	The Penalty for assigning parts to new bins	0.01
HP_MAX_HEIGHT	Maximum Height of Handpickable parts (m)	1.6

12 Results

Upon execution, the model successfully allocated all parts to available bins while satisfying the defined constraints related to various goals. The Below outputs are displayed as the side profile of the runs. The left side of the run in the image represents the front section of the run in the layout. Bins (cells) are numbered by levels (rows) and columns (columns) for each run. Each cell contains the output in the following format:
<Bin number> → <Part Number (12 digits)> × <Quantity>

← Front------------------Back→
Run 1 Bin-Part-Quantity Assignments

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12
Level 2	13 -> TC3154 448G04 × 2	14 -> TC3051 575H06 × 2	15 -> TC3154 873G01 × 1	16 -> TC3154 873G01 × 1	17 -> TC3157 716G02 × 2	18 -> TC3157 716G02 × 2	19 -> TC3156 786G04 × 2	20 -> TC3156 786G04 × 3	21 -> TC3156 786G04 × 3	22 -> TC3156 782G10 × 3	23 -> TC3156 782G10 × 3	24 -> TC3158 301H01 × 8
Level 1	01 -> TC3051 372G01 × 2	02 -> TC3154 150G01 × 2	03 -> TC3051 372G01 × 2	04 -> TC3051 372G01 × 2	05 -> TC3154 150G01 × 2	06 -> TC3154 146G02 × 1	07 -> TC3154 146G02 × 1	08 -> TC3154 146G02 × 1	09 -> TC3161 942G01 × 7	10 -> TC3159 602G01 × 8	11 -> TC3159 603G01 × 8	12 -> TC3159 601G01 × 8

Run 2 Bin-Part-Quantity Assignments

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
Level 2	34 -> TC3154 146G02 × 1	35 -> TC3154 146G02 × 1	36 -> TC3154 873G01 × 1	37 -> TC3157 716G02 × 2	38 -> TC3153 165G01 × 2	39 -> TC3153 165G01 × 2	40 -> TC3153 165G01 × 2	41 -> TC3153 165G01 × 2	42 -> TC3153 165G01 × 2
Level 1	25 -> TC2267 D80G01 × 6	26 -> TC3164 567G01 × 4	27 -> TC3051 372G01 × 2	28 -> TC3154 150G01 × 2	29 -> TC3154 146G02 × 1	30 -> TC3154 146G02 × 1	31 -> TC3154 146G02 × 1	32 -> TC3153 165G01 × 4	33 -> TC3156 782G10 × 9

Run 3 Bin-Part-Quantity Assignments

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
Level 2	52 -> TC3154 448G04 × 2	53 -> TC3154 448G04 × 2	54 -> TC3154 873G01 × 1	55 -> TC3157 716G02 × 2	56 -> TC3153 165G01 × 2	57 -> TC3153 165G01 × 2	58 -> TC3153 165G01 × 2	59 -> TC3153 165G01 × 2	60 -> TC3153 165G01 × 2
Level 1	43 -> TC3151 528G01 × 8	44 -> TC3158 514H01 × 8	45 -> TC3156 091G01 × 4	46 -> TC3051 909G01 × 4	47 -> TC3151 609G01 × 4	48 -> TC3792 C11H03 × 4	49 -> TC3163 995G01 × 1	50 -> TC3154 150G01 × 1	51 -> TC3153 165G01 × 2

Run 4 Bin-Part-Quantity Assignments

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9
Level 3		80 -> TC3154873G01 × 1	81 -> TC3154873G01 × 1	82 -> TC3154873G01 × 1	83 -> TC3154873G01 × 1	84 -> TC3154869G01 × 2	85 -> TC3161942G01 × 1	86 -> TC3156782G10 × 1	87 -> TC3153165G01 × 2
Level 2	70 -> TC2267D80G01 × 2	71 -> TC3154448G04 × 2	72 -> TC3161969G01 × 4	73 -> TC3154869G01 × 2	74 -> TC3154869G01 × 2	75 -> TC3154869G01 × 2	76 -> TC3161970G01 × 4	77 -> TC3153165G01 × 2	78 -> TC3153165G01 × 2
Level 1	61 -> TC3153163G01 × 8	62 -> TC3153163G01 × 8	63 -> TC3051575H06 × 6	64 -> TC3051576H03 × 4	65 -> TC3163995G01 × 3	66 -> TC3154150G01 × 1	67 -> TC3051576H03 × 4	68 -> TC3051367G01 × 8	69 -> TC3051366G01 × 8

13 Discussion

The GP framework unifies multiple operational goals without resorting to ad-hoc priority rules. Because deviations are expressed in natural engineering terms (distance-meters, kilogram-meters), management can interpret trade-offs directly. Integration with the existing Warehouse Management System is straightforward: the optimizer exports an Excel layout identical to the WMS template.

14 Limitations

- The model only considers a static demand snapshot, though practical for automotive and rail industry, stochastic demand should be accounted for fast paced supply chains
- The model does not account for the current or previous warehouse layout, meaning it generates an optimal placement from scratch each time without considering the cost or feasibility of rearranging existing part allocations.
- The bin assignment logic only works for non-containerized items i.e. items directly stored on rack
- A grouping constraint-useful for placing SKUs with practical associations closer together-was not included, as it would have introduced non-linearity into the model formulation.

15 Conclusion and Future Work

This study demonstrates that weighted goal programming provides a practical, interpretable, and extensible framework for multi-criteria warehouse slotting. In the study, the model delivered reductions in travel distance and ergonomic risk while fully utilizing available storage. However, the model is limited in scope. It assumes static demand, which is reasonable for industries like automotive and rail but may not suit fast-moving supply chains. Additionally, it does not consider the current warehouse layout, meaning it optimizes from scratch each time without accounting for the cost of rearranging existing allocations. Furthermore, the current formulation only applies to direct-placement parts and lacks grouping logic to collocate related SKUs, which could enhance picking efficiency in assembly-driven environments. With the help of commercial solvers grouping constraints for enabling snake-picking could be added. Future research can extend the approach to stochastic demand profiles, grouping constraints, and real-time re-slotting triggered by demand shocks.

16 References

- Duque-Jaramillo, J. C., Cogollo-Flórez, J. M., Gómez-Marín, C. G., & Correa-Espinal, A. A. (2024). Warehouse management optimization using a sorting-based slotting approach. Journal of Industrial Engineering and Management, 17(1), 133–150.
- R. de Koster, T. Le-Duc, & K. J. Roodbergen. (2007). Design and control of warehouse order picking: A literature review. European Journal of Operational Research, 182(2), 481–501. https://doi.org/10.1016/j.ejor.2006.07.009
- Perera, D., Mirando, U., & Fernando, A. (2022). Warehouse space optimization using linear programming model and goal programming model. Sri Lanka Journal of Economics, Statistics, and Information Management, 1(1), 103–124.
- Tamiz, M., Jones, D. F., & El-Darzi, E. (1995). A review of goal programming and its applications. Annals of Operations Research, 58, 39–53.
- Tamiz, M., Jones, D. F., & Romero, C. (1998). Goal programming, compromise programming and reference point method formulations: Linkages and utility interpretations. Journal of the Operational Research Society, 49(9), 986–991.
- Viveros, P., González, K., Mena, R., Kristjanpoller, F., & Robledo, J. (2021). Slotting optimization model for a warehouse with divisible first-level accommodation locations. Applied Sciences, 11(3), 936.
- Solano-Charris, E. L., Rojas-Reyes, J. J., & Montoya-Torres, J. R. (2019). The storage location assignment problem: A literature review. International Journal of Industrial Engineering Computations, 10(2), 167–194.
- Silva, A., Coelho, L. C., Darvish, M., & Renaud, J. (2020). Integrating storage location and order picking problems in warehouse planning. Transportation Research Part E: Logistics and Transportation Review, 140, 102003.
- Lavender, S. A., Sun, C., Xu, Y., & Sommerich, C. M. (2021). Ergonomic considerations when slotting piece-pick operations in distribution centers. Applied Ergonomics, 97, 103554.
- Diefenbach, H., Grosse, E. H., & Glock, C. H. (2024). Human- and cost-centric storage assignment optimization in picker-to-parts warehouses. European Journal of Operational Research, 315(3), 1049–1068.