iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

A MULTI-OBJECTIVE OPTIMIZATION OF ROUTING AND SLOTTING POLICIES USING NSGA-II TO IMPROVE SPACE UTILIZATION AND ORDER FULFILLMENT TIME IN A UNIT LOAD WAREHOUSE

Yaman Ayman M. Daboul
Master's Degree Student, Department of Industrial Engineering,
King Abdulaziz University, Jeddah, Saudi Arabia

Ahmed Atef S. Bakhsh
Professor, Department of Industrial Engineering,
King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT
This paper presents a multi-objective optimization framework that jointly optimizes storage slotting, picker routing, and rack beam-height configuration in a forklift-operated selective-rack warehouse containing 2,304 pallet slots across four vertical levels. A graph-based evaluation model calibrated with real operational data (32,630 orders; 208 SKUs; 228,228 picks) quantifies order fulfilment time and vertical space utilization for candidate solutions. Two slotting policies (ABC class-based pooling and Dedicated) and three routing heuristics (S-shape, Return, Largest-gap) are evaluated within an NSGA-II search that generates Pareto-optimal trade-offs between minimizing fulfilment time and minimizing space waste. A MILP sub-model determines the optimal rack beam-height profile across the four levels. Results show that ABC class-based slotting combined with Return routing and an optimized rack configuration of (2.4, 2.0, 1.6, 0.8 m) reduces travel distance to 477,590 km and fulfilment time to 90,787 hours—a 30% improvement over the Random/S-shape baseline (680,450 km; 129,500 h). Height utilization reaches 99.02%, compared with 76.98% for Dedicated storage. The ABC pooling policy also reduces the maximum storage requirement by 34.5% relative to Dedicated storage (2,007 vs. 3,063 slots). Robustness testing under demand multipliers of 1.5× to 3.5× confirms near-linear scaling in fulfilment time and stable space utilization (~99%). TOPSIS multi-criteria ranking yields a closeness coefficient of 0.958 for the recommended ABC + Return solution, decisively outperforming both the baseline (0.510) and Dedicated (0.268).

Keywords:
forklift order picking; ABC class-based storage; Return routing; vertical space utilization; NSGA-II; TOPSIS.

1. INTRODUCTION

Warehouses are critical nodes in modern supply chains, linking supply, storage, and production. Operational efficiency is typically measured through order fulfilment time and space utilization—two objectives that often conflict. Maximizing storage density can increase travel and handling time, while layouts designed for retrieval speed may waste vertical capacity through honeycombing. In selective-rack warehouses operated by forklifts, order picking is the dominant cost driver, accounting for 55–65% of total operating cost [1]. Two decisions largely determine picking effort: where pallets are stored (slotting) and how forklifts travel to collect them (routing). Because these decisions interact—slotting shapes pick density while routing responds to it—optimizing one in isolation often produces limited gains [2]. Despite extensive research on the Storage Location Assignment Problem (SLAP) and routing heuristics independently [3,4], joint optimization for forklift-only, multi-level selective-rack warehouses remains underexplored. Most NSGA-II studies target automated systems (AS/RS) or manual picker-to-parts settings [5], and few embed congestion penalties or vertical handling costs within the optimization fitness function. Furthermore, the pipeline from Pareto optimization to actionable implementation through multi-criteria decision making (MCDM) is rarely documented end-to-end. This paper addresses these gaps through an integrated framework applied to a real industrial case—the TBS raw-materials warehouse in Saudi Arabia. The contributions are: (i) a MILP-based rack beam-height optimizer that eliminates vertical dead space; (ii) a joint NSGA-II optimization of slotting policy, routing heuristic, and rack configuration on a

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

navigation-graph model with congestion approximation; (iii) TOPSIS-based selection of the best-compromise solution; and (iv) robustness validation under demand growth scenarios.

2. METHODOLOGY

2.1 Problem Setting and Data

The case study warehouse contains 2,304 pallet slots distributed across four rack levels with a single I/O point. Three pallet-height categories exist: 0.8 m (49.2% of stock), 1.0 m (30.4%), and 1.2 m (20.4%). The evaluation sample comprises 32,630 historical orders involving 208 active SKUs and 228,228 individual picks. Forklift parameters are: travel speed 1.5 m/s, pick time 15 s/pallet, level-change penalty 12 s/level above ground, and capacity of 1 pallet per trip.

2.2 Navigation Graph and Distance Model

The warehouse layout is modeled as an undirected graph G = (V, E) with four node types: the I/O node, aisle support nodes, storage slot nodes, and virtual aisle-foot projection nodes. Movement is restricted to orthogonal aisle and cross-aisle edges. Each slot is connected to the walkable network through a virtual projection node on the nearest aisle centerline. Travel cost from the I/O point to any slot is computed via Dijkstra's algorithm. Level penalties are added as stub-edge weights: zero for ground level and $(\ell-1) \times 12$ s for upper levels.

2.3 Routing Heuristics

Three standard heuristics are evaluated. S-shape traverses each visited aisle fully. Return enters from one cross-aisle to the deepest required pick and returns to the same side. Largest-gap identifies the largest unvisited segment within an aisle and decides whether to traverse or return based on gap size [6]. Each heuristic generates a waypoint sequence evaluated on the graph.

2.4 Slotting Policies

ABC class-based pooling classifies SKUs by cumulative demand (A ≤ 80%, B ≤ 95%, C remainder) and restricts each class to distance-ranked zones on each level. A single pallet-height category is assigned per level to minimize within-level height heterogeneity. Dedicated slotting enforces one-SKU-per-slot exclusivity without level–height restrictions. Both policies satisfy slot-height capacity, full stock coverage, and assignment–quantity linking constraints.

2.5 Rack Beam-Height Optimization

A MILP model treats level heights H_1–H_4 as decision variables subject to: $\sum H_l = 6.8$ m, $H_1 > H_2 > H_3 > H_4$, and demand fulfillment for each pallet-height category. Binary variables x_{ijk} select a product-height/level/stacking-factor combination. The objective minimizes total waste: $Z = 6.8 - \sum(i \times k) \times x_{ijk}$. Feasible configurations are pre-enumerated and fed as a discrete gene into NSGA-II.

2.6 Bi-Objective Formulation and NSGA-II

The problem minimizes two objectives: (f_1) mean order fulfilment time, computed as the sum of travel time, pick time, level penalties, and a GI/G/1 queueing-based congestion proxy [7]; and (f_2) space waste, defined as 1 − (utilized height / occupied-slot capacity). The NSGA-II chromosome encodes the slotting mode, routing gene, SKU priority keys, per-level zoning fractions, a rack-configuration gene, and per-level pallet-height assignments. Population size is 500 with 50 generations for ABC and 150/25 for Dedicated runs, implemented in PyMOO [8].

2.7 Mathematical Model

This section formalizes the bi-objective optimization problem used to jointly determine (i) slotting and routing decisions and (ii) a feasible rack-height profile for the four rack levels. The model is evaluated over an order-history sample and uses precomputed route-dependent round-trip distances.

2.7.1 Sets and indices

- $\mathcal{K}$: set of SKUs, indexed by k.
- $\mathcal{S}$: set of available storage slots, indexed by s, with $|\mathcal{S}| = 2304$.
- $\mathcal{L} = \{1,2,3,4\}$: set of rack levels, indexed by ℓ.
- $\mathcal{S}_\ell \subseteq \mathcal{S}$: slots located on level ℓ; $\ell(s) \in \mathcal{L}$denotes the level of slot s.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

- $\mathcal{C} = \{A, B, C\}$: ABC classes, indexed by c; $\kappa(k) \in \mathcal{C}$is the class of SKU k. Let $\mathcal{K}^c = \{k \in \mathcal{K}: \kappa(k) = c\}$.
- $\mathcal{R}$: set of routing heuristics, indexed by r(e.g., Return, Largest-gap).
- $\mathcal{M} = \{0,1,2\}$: slotting policies, indexed by m, where $m = 0$Random, $m = 1$ABC, $m = 2$Dedicated.
- $\mathcal{H} = \{0.8,1.0,1.2\}$: admissible pallet-height categories (m), indexed by h.
- $\mathcal{K}^{rack}$: set of feasible rack-height configurations, indexed by κ. Each configuration defines level clear heights $\{H_\ell(\kappa)\}_{\ell \in \mathcal{L}}$ satisfying $H_1(\kappa) > H_2(\kappa) > H_3(\kappa) > H_4(\kappa)$ and $\sum_{\ell \in \mathcal{L}} H_\ell(\kappa) = 6.8$.

ABC zoning (level-specific). For each level ℓ, the slots are ranked from nearest to farthest based on the adopted distance ranking rule, and then partitioned into zones A, B, and $C$by level-wise fractions.

2.7.2 Parameters

- $h_k > 0$: pallet height (m) of SKU k.
- $P_k \in \mathbb{Z}_{\geq 0}$: planned pallets of SKU $k$that must be stored.
- $f_k \in \mathbb{Z}_{\geq 0}$: pick frequency (total pallet picks) of SKU $k$in the evaluation sample.
- $d_s^{(r)} \in \mathbb{R}_{\geq 0}$: precomputed round-trip distance (m) from I/O to slot $s$under route r.
- $v^{pick} > 0$: average forklift speed (m/s).
- $\theta_{pick} \geq 0$: fixed pick/handling time per pallet (s).
- $\theta_{level}(\ell) \geq 0$: level-dependent handling penalty for servicing one pallet at level ℓ.
- $C \in \mathbb{Z}_{>0}$: forklift capacity per trip (pallets).
- Zone fraction bounds: $\underline{\alpha} \leq \alpha_\ell \leq \bar{\alpha}$, $\underline{\beta} \leq \beta_\ell \leq \bar{\beta}$ for each ℓ.

2.7.3 Decision variables

Core assignment variables

- $n_{ks} \in \mathbb{Z}_{\geq 0}$: number of pallets of SKU $k$assigned to slot s.
- $z_{ks} \in \{0,1\}$: equals 1 if SKU $k$occupies slot s(i.e., $n_{ks} > 0$).
- $y_s \in \{0,1\}$: equals 1 if slot $s$is occupied by any pallet.

Policy and configuration variables

- $m \in \mathcal{M}$: slotting policy (Random/ABC/Dedicated).
- $r \in \mathcal{R}$: routing heuristic.
- $y_\kappa \in \{0,1\}$: rack configuration selection variable.
- α_ℓ, β_ℓ: ABC zone fractions for each level ℓ(active only if $m = 1$).
- $a_{sc} \in \{0,1\}$: equals 1 if slot $s$belongs to class zone c(ABC only).
- $q_{\ell h} \in \{0,1\}$: equals 1 if level $\ell$is assigned pallet-height category h(ABC only).
- $\rho \in [0, \bar{\rho}]$: reserve fraction (Dedicated only).
- $\mathbf{u} = (u_k)_{k \in \mathcal{K}}, u_k \in [0,1]$: priority keys used by the decoder to induce a stable SKU allocation order (algorithmic decision vector component).

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Induced level/slot clear heights

$$H_\ell(\mathbf{x}) = \sum_{\kappa \in \mathcal{K}^{rack}} y_\kappa \, H_\ell(\kappa), H_s(\mathbf{x}) = H_{\ell(s)}(\mathbf{x}).$$

2.7.4 Feasibility constraints (shared)

Rack selection

$$\sum_{\kappa \in \mathcal{K}^{rack}} y_\kappa = 1. \qquad \text{(A1)}$$

Slot height capacity

$$\sum_{k \in \mathcal{K}} n_{ks} \, h_k \le H_s(\mathbf{x}) \forall s \in \mathcal{S}. \qquad \text{(A2)}$$

Linking and integrality

Define the height-based slot capacity for SKU $k$in slot $s$as

$$M_{ks}(\mathbf{x}) = \left\lfloor \frac{H_s(\mathbf{x})}{h_k} \right\rfloor. \qquad \text{(A3)}$$

Then enforce

$$n_{ks} \le M_{ks}(\mathbf{x}) \, z_{ks} \forall k \in \mathcal{K}, \forall s \in \mathcal{S}. \qquad \text{(A4)}$$

Full stock coverage

$$\sum_{s \in \mathcal{S}} n_{ks} = P_k \forall k \in \mathcal{K}. \qquad \text{(A5)}$$

lot occupancy definition

$$\sum_{k \in \mathcal{K}} n_{ks} \le M_s \, y_s \forall s \in \mathcal{S}, \qquad \text{(A6)}$$

where $M_s$is a sufficiently large slot-specific upper bound.

2.7.5 Objective 2: space inefficiency (shared)

Let utilized vertical height be

$$U(\mathbf{x}) = \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} n_{ks} \, (\mathbf{x}) \, h_k. \qquad \text{(A7)}$$

To penalize fragmentation, utilization is computed over occupied slots only:

$$H_{tot}(\mathbf{x}) = \sum_{s \in \mathcal{S}} H_s \, (\mathbf{x}) \, y_s(\mathbf{x}), Util(\mathbf{x}) = \frac{U(\mathbf{x})}{H_{tot}(\mathbf{x})}. \qquad \text{(A8)}$$

The second objective is

$$f_2(\mathbf{x}) = 1 - Util(\mathbf{x}). \qquad \text{(A9)}$$

2.7.6 Objective 1: fulfilment-time proxy (shared form)

Fulfilment time is evaluated through a decomposed proxy:

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

$$f_1(\mathbf{x}) = T_{travel}(\mathbf{x}) + T_{level}(\mathbf{x}) + \theta_{pick} \sum_{k \in \mathcal{K}} f_k + Delay(\mathbf{x}), \qquad \text{(A10)}$$

$$\text{where} \qquad T_{travel}(\mathbf{x}) = \frac{D_r(\mathbf{x})}{v^{pick}}. \qquad \text{(A11)}$$

The delay term $Delay(\mathbf{x})$represents congestion via an analytical approximation based on induced flows and a queueing-based penalty on a predefined congestion edge set; its exact construction is provided in the evaluation model section. The remaining policy-dependent elements are $D_r(\mathbf{x})$and $T_{level}(\mathbf{x})$, defined separately for ABC and Dedicated slotting.

2.7.7 Model 1: ABC class-based slotting

ABC zoning and strict assignment

Each slot is assigned to exactly one class zone:

$$\sum_{c \in \mathcal{C}} a_{sc} = 1 \forall s \in \mathcal{S}. \qquad \text{(B1)}$$

Level-wise zone sizes are induced by $\alpha_\ell, \beta_\ell$and the distance-ranked ordering of $\mathcal{S}_\ell$. The first $\lfloor \alpha_\ell \mid \mathcal{S}_\ell \mid \rfloor$slots become zone A, the next $\lfloor \beta_\ell \mid \mathcal{S}_\ell \mid \rfloor$become zone B, and the remainder become zone C(implemented by the decoder).

Strict zoning (no overflow) is enforced by:

$$z_{ks} \leq a_{s,\kappa(k)} \forall k \in \mathcal{K}, \forall s \in \mathcal{S}. \qquad \text{(B2)}$$

One pallet-height category per level (ABC)

$$\sum_{h \in \mathcal{H}} q_{\ell h} = 1 \forall \ell \in \mathcal{L}, \qquad \text{(B3)}$$

and height-category compatibility is enforced by:

$$z_{ks} \leq q_{\ell(s),\, h_k} \forall k \in \mathcal{K}, \forall s \in \mathcal{S}. \qquad \text{(B4)}$$

ABC distance proxy and level exposure

ABC aggregates slot distances at the class–level layer and maps them back to SKUs via their level shares. Define the set of level-ℓ slots used by class c:

$$\mathcal{S}_{c\ell}(\mathbf{x}) = \left\{ s \in \mathcal{S}_\ell : \sum_{k:\kappa(k)=c} n_{ks}(\mathbf{x}) > 0 \right\}. \qquad \text{(B5)}$$

Let the class–level mean distance be:

$$\bar{d}_{rc\ell}(\mathbf{x}) = \frac{1}{\mid \mathcal{S}_{c\ell}(\mathbf{x}) \mid} \sum_{s \in \mathcal{S}_{c\ell}(\mathbf{x})} d_s^{(r)}, \mid \mathcal{S}_{c\ell}(\mathbf{x}) \mid > 0. \qquad \text{(B6)}$$

SKU level shares are:

$$N_{k\ell}(\mathbf{x}) = \sum_{s \in \mathcal{S}_\ell} n_{ks}(\mathbf{x}), \pi_{k\ell}(\mathbf{x}) = \frac{N_{k\ell}(\mathbf{x})}{P_k}. \qquad \text{(B7)}$$

Expected distance per SKU:

$$\bar{d}_{rk}^{ABC}(\mathbf{x}) = \sum_{\ell \in \mathcal{L}} \pi_{k\ell}(\mathbf{x})\, \bar{d}_{r,\kappa(k),\ell}(\mathbf{x}). \qquad \text{(B8)}$$

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Total distance proxy:

$$D_r^{ABC}(\mathbf{x}) = \sum_{k\in\mathcal{K}} f_k \; \tilde{d}_{rk}^{ABC}(\mathbf{x}), T_{travel}^{ABC}(\mathbf{x}) = \frac{D_r^{ABC}(\mathbf{x})}{v^{pick}}. \qquad (B9)$$

Level-handling term:

$$T_{level}^{ABC}(\mathbf{x}) = \sum_{k\in\mathcal{K}} f_k \sum_{\ell\in\mathcal{L}} \pi_{k\ell}(\mathbf{x})\, \theta_{level}(\ell). \qquad (B10)$$

ABC bi-objective model

$$\min_{\mathbf{x}} \; (f_1^{ABC}(\mathbf{x}), f_2(\mathbf{x})) \text{s.t. (A1)–(A6) and (B1)–(B4).} \qquad (B11)$$

2.7.8 Model 2: Dedicated slotting

No mixing within a slot

Each slot may contain pallets from at most one SKU:

$$\sum_{k\in\mathcal{K}} z_{ks} \le 1 \forall s \in \mathcal{S}. \qquad (C1)$$

Dedicated distance proxy and level exposure

Define slot shares for SKU k:

$$w_{ks}(\mathbf{x}) = \frac{n_{ks}(\mathbf{x})}{P_k}, \sum_{s\in\mathcal{S}} w_{ks}(\mathbf{x}) = 1. \qquad (C2)$$

Expected distance per SKU:

$$\tilde{d}_{rk}^{DED}(\mathbf{x}) = \sum_{s\in\mathcal{S}} w_{ks}(\mathbf{x})\, d_s^{(r)}. \qquad (C3)$$

Total distance proxy:

$$D_r^{DED}(\mathbf{x}) = \sum_{k\in\mathcal{K}} f_k \; \tilde{d}_{rk}^{DED}(\mathbf{x}), T_{travel}^{DED}(\mathbf{x}) = \frac{D_r^{DED}(\mathbf{x})}{v^{pick}}. \qquad (C4)$$

Level-handling term:

$$T_{level}^{DED}(\mathbf{x}) = \sum_{k\in\mathcal{K}} f_k \sum_{s\in\mathcal{S}} w_{ks}(\mathbf{x})\, \theta_{level}(\ell(s)). \qquad (C5)$$

2.8 TOPSIS Ranking

Pareto solutions are ranked using four criteria with expert weights: fulfilment time f_1 (cost, w = 0.30), height utilization f_2 (benefit, w = 0.30), maximum storage requirement f_3 (cost, w = 0.20), and stability under demand growth f_4 (cost, w = 0.20). Vector normalization, ideal/anti-ideal computation, and closeness-coefficient ranking follow the standard TOPSIS procedure [9].

3. RESULTS

3.1 Storage Requirement

Before optimization, maximum storage requirements were computed under both policies. As shown in Table 1, ABC class-based pooling requires 2,007 slots compared with 3,063 for Dedicated storage—a 34.5% reduction—because SKUs within each class share capacity and do not all peak simultaneously.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Table 1. Maximum storage requirement comparison.

Metric	Definition	Result (Slots)
M_DED	Sum of each SKU's peak	3,063
M_ABC	Sum of class-zone peaks	2,007

3.2 NSGA-II Optimization

The NSGA-II converged to fully feasible populations by generation 9 (from 1% feasibility at generation 1). Table 2 compares the best solutions at base demand.

Table 2. Best NSGA-II solutions at base demand (1.0×).

Policy	Routing	Distance (km)	Time (h)	Height Util (%)	Rack Config (m)
ABC	Return	477,590	90,787	99.02	2.4/2.0/1.6/0.8
Dedicated	Return	556,689	105,507	76.98	2.4/2.0/1.2/1.2
Baseline (Random)	S-shape	680,450	129,500	85.00	Uniform 1.2

The ABC + Return strategy reduces fulfilment time by 30% relative to the baseline and by 14% relative to Dedicated + Return. The optimized rack configuration (2.4, 2.0, 1.6, 0.8 m) achieves 99.02% height utilization by matching level heights to the pallet-height distribution—allocating the top level exclusively to 0.8 m pallets, which constitute 49.2% of stock. Figure 1 shows the Pareto front for Dedicated and ABC models.

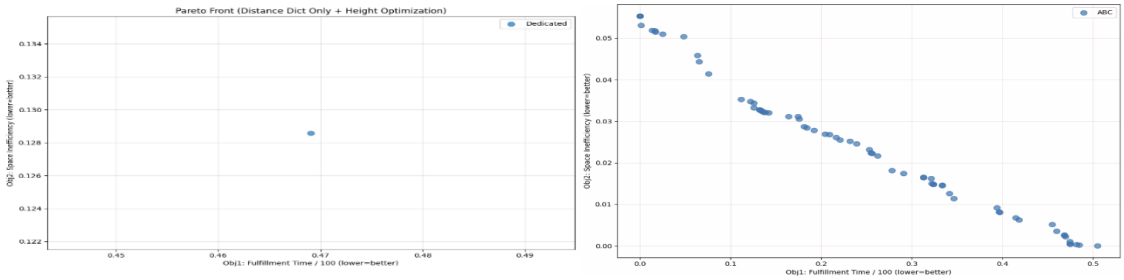

Figure 1 Pareto fronts of Dedicated and ABC models

3.3 Robustness Under Demand Growth

Table 3 shows the recommended ABC + Return solution under demand multipliers of 1.0× to 3.5×. Fulfilment time scales near-linearly while height utilization remains stable at approximately 99%, confirming operational robustness within the existing layout.

Table 3. Robustness results for ABC + Return solution.

Demand Multiplier	Distance (km)	Fulfilment Time (h)	Height Util (%)
×1.0	477,590	90,787	99.02
×1.5	615,090	116,930	98.82
×2.5	1,093,107	207,797	98.93
×3.5	1,571,179	298,675	98.72

3.4 TOPSIS Multi-Criteria Ranking

TOPSIS ranking using four weighted criteria (fulfilment time 30%, height utilization 30%, storage requirement 20%, demand-growth stability 20%) produces the results in Table 4. ABC + Return achieves the highest closeness coefficient (0.958), decisively outperforming both alternatives.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

Table 4. TOPSIS final ranking.

Rank	Alternative	Closeness Coefficient
1	ABC + Return	0.958
2	Baseline (Random + S-shape)	0.510
3	Dedicated + Return	0.268

4. DISCUSSION

The results validate four key findings. First, joint optimization of slotting and routing produces multiplicative efficiency gains that neither decision achieves alone. The ABC policy clusters 79.9% of picks (Class A) near the I/O point, and Return routing exploits this concentration by avoiding unnecessary deep-aisle traversals. This synergy aligns with Silva et al. [2] and Bashatah and Elnaggar [10], who demonstrated that slotting and routing are reciprocally coupled decisions.

Second, the heterogeneous rack configuration (2.4/2.0/1.6/0.8 m) virtually eliminates vertical dead space. Under uniform 1.2 m beam heights, storing a 0.8 m pallet wastes 0.4 m per slot. Since half the inventory is 0.8 m, this waste is substantial. The MILP-optimized configuration assigns the top level exclusively to small pallets, raising height utilization from 85% (baseline) to 99.02%. This confirms the beam-height adjustment strategy advocated by Paveenchana and Phumchusri [11] and addresses the vertical-dimension gap identified by Liang et al. [12] and Gu et al. [3].

Third, ABC pooling reduces the storage footprint by 34.5% compared with Dedicated storage because slots are shared within class zones rather than permanently reserved per SKU. This pooling effect, combined with optimized beam heights, means the warehouse can accommodate growth without physical expansion.
Fourth, the TOPSIS analysis demonstrates that multi-criteria decision making is essential when objectives conflict. Dedicated storage scored lowest (0.268) despite acceptable fulfilment time because its spatial inefficiency (requiring 50% more slots) heavily penalized its overall ranking. This underscores the value of combining evolutionary optimization with structured MCDM for translating Pareto fronts into actionable decisions [9,13].

5. CONCLUSION

This paper developed and validated an integrated optimization framework for a forklift-operated selective-rack warehouse. By jointly optimizing ABC class-based slotting, return routing, and rack beam-height configuration through NSGA-II, the framework achieves a 30% reduction in order fulfilment time and 99% vertical space utilization compared with the existing Random/S-shape baseline—all without facility expansion. ABC pooling further reduces the storage footprint by 34.5% relative to Dedicated storage. TOPSIS multi-criteria ranking confirms the recommended solution (closeness coefficient 0.958) as the best compromise across time, space, capacity, and robustness criteria. Future work should incorporate discrete-event simulation for dynamic congestion validation, batching policy co-optimization, and extension to multi-warehouse networks.

REFERENCES

1) R. De Koster, T. Le-Duc, K.J. Roodbergen, “Design and control of warehouse order picking: A literature review”. Eur. J. Oper. Res., Vol. 182, No. 2, 2007, 481-501.
2) A. Silva et al., “Integrated storage location assignment and order picking optimization”. Comput. Ind. Eng., 2020.
3) J. Gu, M. Goetschalckx, L.F. McGinnis, “Research on warehouse design and performance evaluation: A comprehensive review”. Eur. J. Oper. Res., Vol. 203, No. 3, 2010, 539-549.
4) K.J. Roodbergen, R. De Koster, “Routing order pickers in a warehouse with a middle aisle”. Eur. J. Oper. Res., Vol. 133, No. 1, 2001, 32-43.
5) V. Lesch et al., “Optimizing storage assignment, order picking, and their interaction in mezzanine warehouses”. Appl. Intell., Vol. 53, 2023, 1570-1595.
6) C.G. Petersen, G.R. Aase, “A comparison of picking, storage, and routing policies in manual order picking”. Int. J. Prod. Econ., Vol. 92, No. 1, 2004, 11-19.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

7) J.C.-H. Pan, T.-H. Wu, “Throughput analysis for order picking system with multiple pickers and aisle congestion considerations”. Comput. Oper. Res., Vol. 39, No. 7, 2012, 1661-1672.
8) J. Blank, K. Deb, “pymoo: Multi-objective optimization in Python”. IEEE Access, Vol. 8, 2020, 89497-89509.
9) K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II”. IEEE Trans. Evol. Comput., Vol. 6, No. 2, 2002, 182-197.
10) J.A. Bashatah, G.R. Elnaggar, “Enhancing warehouse picking efficiency through integrated allocation and routing policies”. Appl. Sci., Vol. 15, No. 20, 2025, 11186.
11) K. Paveenchana, N. Phumchusri, “Warehouse capacity optimization by obsolete item removal and beam height adjustment”. M.S. thesis, Chulalongkorn Univ., Thailand, 2019.
12) C. Liang et al., “Storage location assignment in emergency reserve warehouses: A multi-objective optimization algorithm”. Mathematics, Vol. 13, No. 10, 2025, 1636.
13) F.M. Miguel, S. Muñoz, P. Vázquez, “Comparison of MOEAs in an optimization-decision methodology for a joint order batching and picking system”. Mathematics, Vol. 12, No. 8, 2024, 1246.