International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Optimising Warehouse Space Allocation: Operations Research Approaches to the Storage Location Assignment Problem

BY- **Om Magia- 41, Prachi Mangla-42, Tanisha Bedi- 43, Pranav Bogawat- 44, Pratham Agarwal-45**

Anil Surendra Modi School of Commerce (ASMSOC), SVKM's NMIMS, Mumbai, Maharashtra, India

1. Introduction to Operations Research and the Problem Type

1.1 Operations Research

Operations Research (OR) is a scientific discipline that uses mathematical modelling, statistical analysis, and algorithmic optimisation to support complex decision-making under resource limitations and operational constraints (Taha, 2017). An OR study systematically converts an unstructured real-world operational challenge into a structured mathematical abstraction consisting of decision variables, an objective function, and a set of explicit physical or functional constraints. Once formalised, the model is solved using exact analytical procedures, mathematical programming, or approximate heuristics, followed by rigorous validation against historical or real-time operational data.

In industrial logistics and material handling, **Warehouse Space Allocation** is a core domain where OR methodologies significantly outperform manual rules of thumb or empirical intuition. Modern warehouses operate as high-velocity distribution nodes where millions of physical Stock Keeping Units (SKUs) must be received, stored, picked, and dispatched within strict time windows. Facility managers face continuous multi-factorial trade-offs: maximising cubic storage space utilisation while simultaneously minimising order picking travel distance, preventing congestion in pick aisles, balancing rack structural load limits, and maintaining operational flexibility. Mathematical modelling converts these conflicting requirements into solvable optimisation problems, providing quantifiable improvements in warehouse throughput, labour efficiency, and floor space productivity.

1.2 The Allocation Problem and Storage Location Assignment

Warehouse space allocation is mathematically categorised under the **Storage Location Assignment Problem (SLAP)** and allocation programming models. In its foundational formulation, a warehouse consists of a discrete set of storage locations or bays $j \in \{1,2,\dots,n\}$ each with a volumetric capacity S_j (m^3), located at specific coordinates relative to Input/Output (I/O) docks. A set of product items or SKU families $i \in \{1,2,\dots,m\}$ must be allocated across these storage locations to satisfy volumetric storage requirements D_i (m^3).

Let c_{ij} represent the unit handling or travel cost associated with allocating one unit of volume of product i to storage location j. The cost coefficient c_{ij} is directly proportional to the product's order pick frequency (turnover rate f_i) and the physical distance d_j between storage location j and the I/O dock ($c_{ij} = f_i \cdot d_j$).

The decision variable $x_{ij} \geq 0$ denotes the volume (m^3) of product i allocated to storage location j. The classical linear programming model for deterministic warehouse space allocation is expressed as:

$$\text{Minimise } Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij}$$

$$\text{subject to } \sum_{j=1}^{n} x_{ij} \geq D_i \quad \forall i \in \{1,\dots,m\} \quad \text{(Demand Satisfaction)}$$

$$\sum_{i=1}^{m} x_{ij} \leq S_j \quad \forall j \in \{1,\dots,n\} \quad \text{(Storage Capacity Limits)}$$

$$x_{ij} \geq 0 \quad \forall i,j \quad \text{(Non-Negativity)}$$

When total inventory storage requirements equal total available rack space ($\sum D_i = \sum S_j$), the system represents a balanced allocation model. If available warehouse space exceeds inventory requirements ($\sum S_j > \sum D_i$), a dummy product category with zero unit cost is introduced to absorb excess capacity.

Numerical Illustration of Warehouse Space Allocation

To demonstrate the mathematical mechanism of the LP model in a facility layout context, consider a regional distribution centre with two distinct storage zones ($j \in$ {Zone 1 (Near Dock),Zone 2 (Far Dock)}) and three product categories ($i \in$ {Class A (Fast),Class B (Medium),Class C (Slow)}). Zone 1 and Zone 2 have volumetric capacities of 400 m^3 and 600 m^3, respectively. The volumetric storage requirements for Class A, B, and C items are 300 m^3, 350 m^3, and 350 m^3 (total demand = 1000 m^3; total supply = 1000 m^3).

The unit transport cost c_{ij} (in ₹ per m^3) reflects the travel distance multiplied by picking frequency:

Storage Zone	**Class A (Fast)**	**Class B (Medium)**	**Class C (Slow)**	**Zone Capacity (Sj)**
Zone 1 (Near I/O)	₹ 2	₹ 5	₹ 8	400 m^3
Zone 2 (Far I/O)	₹ 6	₹ 7	₹ 9	600 m^3
Product Demand (D_i)	300 m^3	350 m^3	350 m^3	1000 m^3

Solving this model using the Simplex algorithm yields the following optimal allocation matrix:

> **Class A to Zone 1:** 300 m^3 at ₹ 2/m^3 = ₹ 600,, **Class B to Zone 1:** 100 m^3 at ₹ 5/m^3 = ₹ 50 , **Class B to Zone 2:** 250 m^3 at ₹ 7/m^3 = ₹ 1,750, **Class C to Zone 2:** 350 m^3 at ₹ 9/m^3 = ₹ 3,150

Total Optimal Cost: $Z^* = 600 + 500 + 1{,}750 + 3{,}150 =$ ₹ 6,000.

Sensitivity testing via dual variables (shadow prices) verifies that allocating higher-velocity Class A products closest to the I/O dock minimizes total travel work ($f_i \cdot d_j$). This mathematical result forms the theoretical basis for the **Cube-per-Order Index (COI)** rule in warehouse design.

1.3 From the Classical Model to Warehouse Space Allocation

While classical LP models provide foundational insight, operational warehouse space management breaks standard assumptions:

1. **Item Compatibility and Co-location:** Products often cannot be stored randomly; item affinity (frequently co-ordered products) requires proximity clustering.
2. **Physical Rack and Slot Geometry:** Slots are discrete, multi-dimensional entities (height, width, depth, weight capacity) rather than continuous, homogeneous volumes.
3. **Dynamic Velocity and Seasonality:** Product turnover rates fluctuate dynamically across life cycles, invalidating static deterministic assumptions.

To evaluate how modern OR addresses these operational complexities, this analytical report reviews five benchmark research papers:

No.	Paper	Operational Context	OR Modeling Methodology
P1	**Muppani & Adil (2008)**	Fast-moving consumer goods (FMCG) storage	Non-Linear Integer Programming (NLIP) for joint class formation and variable slot sizing
P2	**Fumi, Scarabotti & Schiraldi (2013)**	Multi-category commercial distribution centres	Mixed-Integer Linear Programming (MILP) trade-off model for space vs. travel distance
P3	**Chiang, Lin & Chen (2011)**	High-density retail fulfilment centres	Association rule mining integrated with Non-Linear Integer Programming & Genetic Algorithms
P4	**Hausman, Schwarz & Graves (1976)**	Automated Storage & Retrieval Systems (AS/RS)	Continuous analytical probability models for optimal class-based space zoning
P5	**Yang & Nguyen (2020)**	High-velocity e-commerce fulfilment facilities	Constrained k-means clustering paired with Integer Programming for dynamic slotting

2. Understanding the Real-Life Context

2.1 Industry Description: Warehouse Operations, Storage Systems, and Space Management

Warehousing systems form the structural backbones of modern global supply chains. A standard warehouse performs four primary material handling operations: receiving, put-away (storage allocation), order picking, and shipping. Among these, **order picking** accounts for approximately 55% to 65% of total warehouse operating costs, with picker travel time consuming over 50% of total picking activity (Gu et al., 2007; de Koster et al., 2007).

Warehouse Operational Flow Steps:

1. **Receiving & Unloading Dock:** Inbound shipment arrival, unloading, and inventory verification.
2. **Put-Away & Space Allocation (SLAP):** SKU storage location assignment based on turnover velocity, slot geometry, and item affinity rules.
3. **Storage Zoning & Placement:**
    - **Zone 1 (Near Dock):** Fast-moving, high-velocity SKUs (Class A).
    - **Zone 2 (Mid Aisle):** Medium-moving SKUs (Class B).
    - **Zone 3 (Far Aisle):** Slow-moving, low-velocity SKUs (Class C).
4. **Order Picking & Consolidation:** Picker routing and item retrieval governed by physical rack slotting.
5. **Shipping & Dispatch Dock:** Order packing, staging, and outbound transportation loading.

Facility managers implement specific storage assignment policies to optimise space and picking performance:

- **Dedicated Storage:** Every SKU is assigned a permanently reserved physical slot. While simplifying inventory tracking, this policy results in poor space utilisation (often below 50%) because rack space remains empty when inventory drops.

- **Randomised / Shared Storage:** Incoming pallets are assigned to any available open location. This yields high space utilisation (>85%) but increases average picker travel distance, as fast-moving items become scattered throughout the facility.

- **Class-Based Storage:** SKUs are categorised into velocity classes (e.g., Pareto ABC analysis) based on turnover. Class A items are assigned to locations nearest to the I/O docks, while Class B and C items are stored progressively further away. This offers a middle ground, capturing picking efficiency while maintaining high storage density.

2.2 The Problem Addressed in Each Paper

P1: Joint Class Formation and Slot Sizing (Muppani & Adil, 2008)

Conventional class-based storage models assume fixed, predefined class boundaries (e.g., top 20% items = Class A) and uniform rack slot dimensions. However, assigning items with diverse volumetric footprints to fixed-size slots creates significant dead space (unused air space inside storage bays) or triggers frequent replenishment tasks. Muppani and Adil formulate a model that simultaneously optimises:

1. The optimal grouping of SKUs into turnover classes.
2. The physical dimensions and numbers of storage slots allocated to each class.
3. The spatial assignment of classes across warehouse storage bays to minimize operational handling and storage space costs.

P2: Mathematical Allocation under Space-Travel Distance Trade-offs (Fumi et al., 2013)

Industrial warehouses frequently encounter spatial constraints where minimizing travel distance requires expanding pick aisles, which directly reduces total available cubic storage capacity. Fumi et al. address the Storage Location Assignment Problem by formulating a comprehensive Mixed-Integer Linear Programming model that evaluates dedicated versus shared allocation. Their model quantifies the trade-off between material handling travel costs and space consumption overhead under explicit warehouse structural geometry.

P3: Association-Based Storage Allocation with SKU Co-location (Chiang et al., 2011)

Standard turnover-based assignment models treat SKUs independently. However, in retail and e-commerce distribution, specific products are frequently ordered together (e.g., cell phones and protective cases). Storing strongly associated SKUs in distant locations forces pickers to travel across multiple aisles per order. Chiang et al. address the association-based SLAP by integrating data mining (association rule discovery) with mathematical space allocation to place affinity-grouped SKUs in contiguous rack clusters.

P4: Automated Storage and Retrieval Systems (AS/RS) Space Zoning (Hausman et al., 1976)

In automated high-bay warehouses, automated storage/retrieval cranes move simultaneously along vertical and horizontal axes to store and retrieve loads. Hausman et al. investigate optimal space zoning policies within AS/RS racks. They formulate mathematical travel-time models to establish optimal class boundaries under dedicated, randomized, and N-class turnover storage policies, proving analytically how class zoning reduces crane travel cycle times.

P5: Dynamic E-Commerce Slotting under Rack Physical Constraints (Yang & Nguyen, 2020)

E-commerce fulfillment centers handle high SKU velocity variations, seasonal demand spikes, and strict physical rack constraints (weight limits, height clearances, shelf loading capacities). Static slotting models fail under dynamic demand. Yang and Nguyen address dynamic space allocation by coupling constrained machine learning (k-means clustering) with binary integer programming to re-slot SKUs into physical storage zones dynamically without creating excessive warehouse re-stocking overhead.

2.3 Why These Particular OR Techniques Were Used

Classification of OR Methodologies for Space Allocation:

- **Mathematical Programming (Exact Optimality):**
  - *Mixed-Integer Linear Programming (MILP):* Applied for deterministic structural space allocation under linear physical constraints (e.g., P2).
  - *Non-Linear Integer Programming (NLIP):* Applied when variable slot sizing and class assignment interact non-linearly (e.g., P1).
- **Heuristics & Artificial Intelligence (Scalability & Dynamic Adaptation):**
  - *Metaheuristics (Genetic Algorithms):* Applied for high-dimensional SKU co-location and complex combinatorial optimization (e.g., P3).
  - *Machine Learning (Constrained Clustering):* Applied for real-time SKU velocity grouping under rack weight and height limits (e.g., P5).

The choice of mathematical technique directly reflects the structural complexity, problem scale, and physical constraints of each setting:

- **Mixed-Integer Linear Programming (MILP) & NLIP (P1, P2):** Exact mathematical formulations are selected when precise global optima are required for capital-intensive decisions (such as physical rack layout design or slot sizing). Nonlinearities arise when decision variables interact (e.g., slot length multiplied by item assignment binary variables in P1), requiring non-linear integer programming or reformulation strategies.
- **Analytical / Continuous Calculus Models (P4):** In automated AS/RS systems, crane movement is governed by physical speed profiles (simultaneous horizontal and vertical travel times). Continuous integration and probability distribution functions allow closed-form analytical solutions for travel time expected values.
- **Data Mining & Metaheuristics (P3):** Computing item-pair affinity matrices across thousands of SKUs generates combinatorial expansion. Because solving large-scale quadratic binary assignment problems is NP-hard, metaheuristic algorithms (Genetic Algorithms, Simulated Annealing) are deployed to identify near-optimal spatial clusters within reasonable run times.
- **Constrained Machine Learning (P5):** High-velocity e-commerce facilities require real-time re-slotting. Unsupervised clustering algorithm (k-means) modified with physical capacity constraints provides scalable dimensionality reduction before feeding aggregated groups into integer programming allocation algorithms.

3. Modelling and Problem Formulation

This section detailly presents the mathematical structures of the five core models, explicitly defining sets, parameters, decision variables, objective functions, structural constraints, and operational assumptions.

3.1 P1: Class-Based Storage Location Assignment and Slot Formation (Muppani & Adil, 2008)

Type of Problem

A Non-Linear Integer Programming (NLIP) formulation for joint class formation, variable slot sizing, and spatial location assignment in a manual order-picking warehouse.

Sets and Parameters

- I: Set of products / SKUs, indexed by $i \in \{1,2,\dots,N\}$.
- K: Set of storage classes, indexed by $k \in \{1,2,\dots,K\}$.
- S: Set of discrete available slot sizes, indexed by $s \in \{1,2,\dots,S\}$.

- f_i: Picking order frequency (turnover) of product i (orders/period).
- v_i: Unit volumetric footprint of product i (m^3/unit).
- d_k: Average travel distance from I/O dock to storage locations belonging to class zone k (meters).
- V_{ks}: Storage volume capacity of a slot of size s allocated to class k (m^3).
- C^H: Unit handling cost per meter traveled (₹/meter).
- C^S: Storage rack cost per unit volume of installed slot space (₹/m^3).
- D_i: Total period demand quantity for product i (units).

Decision Variables

- $x_{ik} \in \{0,1\}$: Equals 1 if product i is assigned to turnover class k; 0 otherwise.
- $y_{iks} \in \{0,1\}$: Equals 1 if product i assigned to class k is allocated a slot of size s; 0 otherwise.
- $n_{ks} \geq 0$: Integer variable representing the total number of slots of size s installed in class zone k.

Objective Function

Minimizes total operational handling costs (picker travel) plus total capital rack storage space costs:

$$\text{Minimise } Z = C^H \sum_{i=1}^{N} \sum_{k=1}^{K} f_i \cdot d_k \cdot x_{ik} + C^S \sum_{k=1}^{K} \sum_{s=1}^{S} n_{ks} \cdot V_{ks}$$

Constraints

1. **Unique Class Assignment:** Each SKU must be assigned to exactly one turnover storage class:

$$\sum_{k=1}^{K} x_{ik} = 1 \quad \forall i \in \{1, \dots, N\}$$

2. **Unique Slot Size Allocation:** Each SKU assigned to a class must be allocated exactly one physical slot size:

$$\sum_{s=1}^{S} y_{iks} = x_{ik} \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}$$

3. **Physical Volumetric Capacity Fit:** The physical space provided by the allocated slot size s must accommodate the required inventory storage volume of product i:

$$\sum_{s=1}^{S} y_{iks} \cdot V_{ks} \geq v_i \cdot D_i \cdot x_{ik} \quad \forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}$$

4. **Slot Balance Constraint:** The total number of installed slots n_{ks} in zone k must equal or exceed the total number of SKUs assigned to that class-size combination:

$$n_{ks} \geq \sum_{i=1}^{N} y_{iks} \quad \forall k \in \{1, \dots, K\}, \forall s \in \{1, \dots, S\}$$

5. **Turnover Class Sequencing:** Ensures higher-turnover classes are mapped to zones with shorter travel distances ($d_1 < d_2 < \cdots < d_K$):

$$\frac{\sum_{i=1}^{N} f_i x_{ik}}{\sum_{i=1}^{N} x_{ik}} \geq \frac{\sum_{i=1}^{N} f_i x_{i,k+1}}{\sum_{i=1}^{N} x_{i,k+1}} \quad \forall k \in \{1, \dots, K-1\}$$

6. **Integrity and Non-negativity:**

$$x_{ik}, y_{iks} \in \{0,1\}, \quad n_{ks} \in \mathbb{Z}^+ \quad \forall i, k, s$$

3.2 P2: Linear Programming Model for SLAP with Space Constraints (Fumi et al., 2013)

Type of Problem

A Mixed-Integer Linear Programming (MILP) model optimizing warehouse space allocation under dedicated and shared storage policies subject to physical aisle geometry.

Sets and Parameters

- I: Set of inventory items / SKUs ($i \in \{1, \dots, M\}$).
- J: Set of storage bays / locations ($j \in \{1, \dots, N\}$).
- T: Set of time periods in the planning horizon ($t \in \{1, \dots, T\}$).
- q_{it}: Quantity of SKU i present in inventory during period t (pallets).
- C_j: Volumetric storage capacity of bay j (pallets).
- d_j: Distance from bay j to the main I/O aisle (meters).
- p_i: Picking order frequency of SKU i per period.
- W: Maximum total floor area available for warehouse storage racks (m^2).
- a_j: Floor area footprint consumed by storage rack bay j (m^2).

Decision Variables

- $z_{ij} \in \{0,1\}$: Equals 1 if storage location j is assigned to SKU i (under dedicated storage); 0 otherwise.
- $x_{ijt} \geq 0$: Number of pallets of SKU i stored in location j during period t (under shared storage).

Objective Function

Minimizes total annual travel distance for picking activities while accounting for space overhead:

$$\text{Minimise } Z = \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{j=1}^{N} p_i \cdot d_j \cdot x_{ijt}$$

Constraints

1. **Inventory Storage Satisfaction:** The total stored pallets across all locations in period t must equal required inventory:

$$\sum_{j=1}^{N} x_{ijt} = q_{it} \quad \forall i \in \{1, \dots, M\}, \forall t \in \{1, \dots, T\}$$

2. **Bay Volumetric Capacity:** Total pallets placed in bay j during period t cannot exceed bay capacity:

$$\sum_{i=1}^{M} x_{ijt} \leq C_j \quad \forall j \in \{1, \dots, N\}, \forall t \in \{1, \dots, T\}$$

3. **Dedicated Allocation Linking (Dedicated Storage Policy Variant):** Restricts storage of SKU i in location j only if location j is dedicated to item i:

$$x_{ijt} \leq C_j \cdot z_{ij} \quad \forall i \in \{1, \dots, M\}, \forall j \in \{1, \dots, N\}, \forall t \in \{1, \dots, T\}$$

4. **Single Item per Bay (Dedicated Policy):**

$$\sum_{i=1}^{M} z_{ij} \leq 1 \quad \forall j \in \{1, \dots, N\}$$

5. **Total Physical Footprint Constraint:**

$$\sum_{j=1}^{N} a_j \left(\max_{i,t} \{z_{ij}\} \right) \leq W$$

6. **Non-Negativity:**

$$x_{ijt} \geq 0, \quad z_{ij} \in \{0,1\}$$

3.3 P3: Association-Based SLAP with Item Affinity Matrix (Chiang et al., 2011)

Type of Problem

A Non-Linear Quadratic Binary Assignment Model integrating Data Mining (Apriori Association Rule Mining) with space allocation to optimize SKU co-location.

Sets and Parameters

- I: Set of SKUs ($i, g \in \{1, \dots, N\}$).
- J: Set of physical rack locations ($j, k \in \{1, \dots, N\}$).
- f_i: Pick frequency of SKU i.
- d_{jk}: Distance between storage location j and storage location k (meters).
- d_{j0}: Distance between storage location j and the I/O dock (meters).
- r_{ig}: Affinity/Association index between SKU i and SKU g derived from historical market basket order data ($0 \leq r_{ig} \leq 1$).
- α, β: Weighting parameters balancing dock travel distance vs. item co-location affinity ($\alpha + \beta = 1$).

Decision Variable

- $y_{ij} \in \{0,1\}$: Equals 1 if SKU i is assigned to storage location j; 0 otherwise.

Objective Function

Minimizes a combined objective function evaluating (1) total travel distance to I/O dock and (2) intra-route travel distance between strongly associated SKUs:

$$\text{Minimise } Z = \alpha \sum_{i=1}^{N} \sum_{j=1}^{N} f_i \cdot d_{j0} \cdot y_{ij} + \beta \sum_{i=1}^{N} \sum_{g=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} r_{ig} \cdot d_{jk} \cdot y_{ij} \cdot y_{gk}$$

Constraints

1. **Location Assignment Constraint:** Every SKU must be assigned to exactly one location:

$$\sum_{j=1}^{N} y_{ij} = 1 \quad \forall i \in \{1, \dots, N\}$$

2. **Location Occupancy Constraint:** Every location can hold at most one SKU:

$$\sum_{i=1}^{N} y_{ij} = 1 \quad \forall j \in \{1, \dots, N\}$$

3. **Binary Constraints:**

$$y_{ij} \in \{0,1\} \quad \forall i, j$$

3.4 P4: Travel Time Optimization in Automated Storage & Retrieval Systems (Hausman et al., 1976)

Type of Problem

Continuous Analytical Probability Formulation for class-based space allocation in Automated Storage and Retrieval Systems (AS/RS).

AS/RS Rack Face Zoning Layout:

- **Origin / I/O Dock Location:** Situated at normalized coordinates $(0.0,0.0)$ at the bottom-left corner of the rack face.
- **Class A Zone (High Turnover):** Concentrated in the immediate contiguous L-shaped region nearest to the dock (bounded by normalized coordinates 0.0 to $\sqrt{L_1}$).
- **Class B Zone (Medium Turnover):** Located in the intermediate L-shaped region surrounding Class A (bounded between $\sqrt{L_1}$ and $\sqrt{L_2}$).
- **Class C Zone (Low Turnover):** Outer perimeter region extending to the top and far edges of the rack face (bounded between $\sqrt{L_2}$ and 1.0).

Analytical Formulation and Parameters

Consider an AS/RS rack normalized to a square in time $T = \max(T_x, T_y) = 1$, where T_x and T_y are maximum crane travel times in horizontal and vertical directions. Crane travel in both axes occurs simultaneously (Chebyshev metric).

The travel time between the I/O point $(0,0)$ and any point (x, y) in the rack face is $t(x, y) = \max(x, y)$, where $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

Let products be ordered by turnover activity according to a continuous cumulative distribution function $G(L)$, where L is the fraction of total items ($0 \leq L \leq 1$) and $G(L)$ is the fraction of total pick activity generated by top items L. The Cube-per-Order Index (COI) orders items such that $g(L) = G'(L)$ is monotonically decreasing.

Expected Travel Time Objective Function

For a k-class storage system, the rack face is divided into k L-shaped boundary zones ($L_0 = 0 < L_1 < L_2 < \cdots < L_k = 1$).

The expected single-command travel time $E[T]$ of the retrieval crane is formulated as:

$$E[T] = \sum_{m=1}^{k} \left(G(L_m) - G(L_{m-1})\right) \cdot E[T \mid \text{Zone } m]$$

where $E[T \mid \text{Zone } m]$ is the expected crane travel time to points within zone m, defined by continuous integration over the L-shaped region bounded by coordinates $(x_m, y_m) = (\sqrt{L_m}, \sqrt{L_m})$:

$$E[T \mid \text{Zone } m] = \frac{\int_{\sqrt{L_{m-1}}}^{\sqrt{L_m}} 2 \cdot z \cdot z \, dz}{L_m - L_{m-1}} = \frac{\frac{2}{3}\left(L_m^{3/2} - L_{m-1}^{3/2}\right)}{L_m - L_{m-1}}$$

Thus, the overall objective reduces to solving optimal continuous boundary points L_m^* that minimize $E[T]$:

$$\text{Minimise } E[T] = \sum_{m=1}^{k} \left(G(L_m) - G(L_{m-1})\right)\left(\frac{2}{3}\frac{L_m^{3/2} - L_{m-1}^{3/2}}{L_m - L_{m-1}}\right)$$

3.5 P5: Constrained Machine Learning & Dynamic Slotting Model (Yang & Nguyen, 2020)

Type of Problem

A multi-stage optimization framework combining constrained k-means clustering with Binary Integer Programming for dynamic class-based storage allocation under physical rack weight and height constraints.

Model Formulation

- **Stage 1: Constrained k-means Clustering:** Groups N SKUs into K cluster zones based on two feature vectors: (1) Turnover rate f_i and (2) Physical item volume/weight profile w_i.

$$\text{Minimise } J = \sum_{k=1}^{K} \sum_{i \in C_k} \left(\|f_i - \mu_k^f\|^2 + \gamma \|w_i - \mu_k^w\|^2\right)$$

$$\text{subject to } \sum_{i \in C_k} v_i \, D_i \leq V_k^{\max} \quad \forall k \in \{1, \ldots, K\} \quad \text{(Volumetric Capacity Constraint)}$$

$$\sum_{i \in C_k} w_i \, D_i \leq W_k^{\max} \quad \forall k \in \{1, \ldots, K\} \quad \text{(Rack Load Capacity Constraint)}$$

- **Stage 2: Integer Programming Zone-to-Bay Allocation:** Assigns aggregated SKU clusters $k \in \{1, \ldots, K\}$ to warehouse shelf bays $j \in \{1, \ldots, B\}$.
- $u_{kj} \in \{0,1\}$: Equals 1 if SKU cluster k is assigned to rack bay j; 0 otherwise.

$$\text{Minimise } Z = \sum_{k=1}^{K} \sum_{j=1}^{B} \left(\bar{f}_k \cdot d_j + \theta \cdot \text{Risk}_j\right) u_{kj}$$

$$\text{subject to } \sum_{j=1}^{B} u_{kj} = 1 \quad \forall k \in \{1, \ldots, K\}$$

$$\sum_{k=1}^{K} \bar{V}_k \, u_{kj} \leq \text{BayCap}_j \quad \forall j \in \{1, \ldots, B\}$$

$$\text{Risk}_j = \begin{cases} \text{High Weight Penalty} & \text{if heavy items placed on upper tiers } (h_j > 2\text{m}) \\ 0 & \text{otherwise} \end{cases}$$

3.6 Summary Matrix of the Formulations

Paper	Problem Formulation Type	Primary Objective Function	Key Operational Constraints	Storage Assignment Policy
P1: Muppani & Adil (2008)	Non-Linear Integer Programming (NLIP)	Minimizes handling travel cost + slot space costs	SKU-to-class uniqueness, slot capacity, class turnover order	Class-Based with Variable Slot Sizes
P2: Fumi et al. (2013)	Mixed-Integer Linear Programming (MILP)	Minimizes annual picking travel distance	Inventory volume balance, rack footprint limits, bay capacities	Dedicated vs. Shared Storage comparison
P3: Chiang et al. (2011)	Quadratic Binary Assignment Model	Minimizes I/O dock distance + co-picking intra-aisle distance	Single-item per location, association affinity matrix	Association-Based Clustering
P4: Hausman et al. (1976)	Continuous Analytical Calculus Model	Minimizes expected crane travel time $E[T]$	Chebyshev travel metric, continuous rack geometry (L-shaped boundaries)	N-Class AS/RS Zoning
P5: Yang & Nguyen (2020)	Constrained Clustering + Binary IP	Minimizes travel distance + rack structural load risk	Physical weight limits, shelf height limits, volumetric capacity	Dynamic Class-Based E-Commerce Slotting

4. Solutions Derived and Methods Used

Solution Methodology Taxonomy:

- **Exact Methods:**
  - *Branch-and-Bound algorithms (P1):* Integrates linearization and Priority Indexing rules.
  - *Simplex and Branch-and-Cut solvers (P2):* Solves MILP formulations via Gomory cuts and cover cuts.
  - *Continuous calculus integrals (P4):* Computes closed-form expectations over continuous L-shaped rack zones.
- **Heuristics and Metaheuristics:**
  - *Genetic Algorithms (P3):* Uses cycle crossover operators (CX) to optimize non-linear SKU affinity co-location.
  - *Constrained k-means clustering (P5):* Balances velocity clustering with shelf weight and height bounds.

- *Simulated Annealing heuristics (P1):* Provides rapid upper bounds for large integer decomposition problems.

4.1 P1: Branch-and-Bound / Non-Linear Integer Decomposition

Because model P1 contains non-linear products of binary decision variables ($x_{ik} \cdot y_{iks}$), direct solution via standard linear solvers is computationally impractical for large SKU counts. Muppani and Adil developed a custom **Branch-and-Bound (B&B)** algorithm paired with a structural decomposition heuristic:

1. **Linearization / Relaxation:** The non-linear product term $x_{ik}y_{iks}$ is replaced by a continuous variable $w_{iks} \geq 0$ bounded by linear inequalities ($w_{iks} \leq x_{ik}, w_{iks} \leq y_{iks}, w_{iks} \geq x_{ik} + y_{iks} - 1$).
2. **Branching Strategy:** The algorithm branches on class assignment variables x_{ik} using a priority indexing rule based on SKU turnover values f_i.
3. **Bounding:** Lower bounds are computed by solving the LP-relaxed transportation sub-problem, while upper bounds are derived using a fast Simulated Annealing (SA) heuristic.

Across experimental tests involving 100 SKUs and 3 slot sizes, the B&B heuristic identified exact optimal solutions in under 180 seconds, outperforming full enumerative search by a factor of 40.

4.2 P2: Mixed-Integer Linear Programming (MILP) with Direct Solvers

Fumi et al. solved their MILP model directly using commercial mathematical programming solvers (CPLEX / Gurobi) utilizing the **Branch-and-Cut** algorithm:

- The Branch-and-Cut framework solves successive linear programming relaxations, generating cutting planes (Gomory cuts and cover cuts) to tighten the feasible region containing integer decisions.
- By testing various warehouse aisle layouts, the exact model established that implementing a **Shared Storage Policy** reduces total required rack volume by 22% to 35% compared to Dedicated Storage, while increasing picking travel distance by only 4.2%.

4.3 P3: Data Mining & Genetic Algorithm Metaheuristic

Solving the quadratic assignment problem in P3 is NP-hard. Chiang et al. implemented a two-stage hybrid solution approach:

1. **Stage 1: Association Rule Mining (Apriori Algorithm):** Transaction records from the Warehouse Management System (WMS) are mined to compute support and confidence metrics for SKU co-occurrences. This generates an $N \times N$ SKU affinity matrix r_{ig}.
2. **Stage 2: Genetic Algorithm (GA) Optimization:**
   - *Chromosome Representation:* An integer vector of length N, where the i-th gene value represents the physical rack location j assigned to SKU i.
   - *Fitness Function:* Reciprocal of the objective function ($1/Z$).
   - *Genetic Operators:* Cycle Crossover (CX) to preserve valid permutation assignments without generating duplicate location assignments, and Swap Mutation.

Experimental results on 500 SKUs demonstrated that the hybrid GA-Apriori algorithm achieved an 18.6% reduction in order picking travel distance compared to standard ABC turnover slotting.

4.4 P4: Continuous Integral Formulations & COI Optimization

Hausman et al. solved the continuous expected travel time model using calculus-based numerical optimization:

- The optimal class boundary values $L_1^*, L_2^*, \ldots, L_{k-1}^*$ were computed by setting the partial derivatives $\frac{\partial E[T]}{\partial L_m} = 0$.
- For a 20/80 Pareto distribution curve ($G(L) = L^{0.1386}$), the authors proved analytically that transitioning from randomized storage to a **3-Class Storage Policy** captures over 85% of the maximum theoretical travel time reduction achievable under full dedicated storage, while retaining high space flexibility.

4.5 P5: Constrained *k*-Means Clustering & Heuristic Space Allocation

Yang and Nguyen implemented a two-phase computational approach:

1. **Phase 1: Constrained Machine Learning:** Modified k-means algorithm where distance metrics incorporate physical weight/volume constraints. Violations of rack weight bounds trigger cluster reassignment penalties during centroid updates.
2. **Phase 2: Integer Programming Allocation:** The formed SKU clusters are mapped to physical rack bays using Gurobi.

In an e-commerce facility handling 2,500 SKUs, the constrained clustering algorithm executed in 42 seconds, reducing physical re-slotting labor by 31% while maintaining optimal picker travel efficiency.

5. Software Used and Validation of the Solutions

Paper	Software & Tools Used	Computational Hardware Specs	Validation Strategy & Benchmark Comparisons
P1	C++ Custom Code, LINGO 8.0	Intel Core 2 Duo, 2.4 GHz, 2 GB RAM	Validated against full enumeration on small instances; benchmarked against standard fixed-slot ABC policies.
P2	IBM ILOG CPLEX, MATLAB	Intel Core i7, 3.2 GHz, 8 GB RAM	Tested on real-world industrial FMCG distribution centers; validated against historical WMS picking logs.
P3	R (Apriori Mining), Python GA Library	Intel Core i5, 2.8 GHz, 4 GB RAM	Validated using 50,000 historical customer order receipts from an online retail facility.
P4	FORTRAN, Numerical Integration	Mainframe Computing System	Validated via Monte Carlo computer simulation of AS/RS crane travel cycles.
P5	Python 3.8, Gurobi Optimizer 9.0	Intel Core i7, 16 GB RAM	Validated on a live e-commerce fulfillment warehouse dataset over 6 months of dynamic demand cycles.

6. Critical Analysis: Summary, Key Findings and Impact

6.1 Summary of Key Findings

Across the reviewed studies, four foundational principles emerge regarding warehouse space allocation:

1. **Dominance of Class-Based Zoning over Extreme Policies:** Pure dedicated storage yields poor space utilization (<50%), whereas pure randomized storage maximizes picker travel distance. A **3-Class or 4-Class assignment policy** captures 80–90% of picking efficiency benefits while maintaining high storage density.

2. **Impact of Variable Slot Sizing:** Fixed slot dimensions generate up to 30% dead air space in storage racks. Jointly optimizing slot sizes alongside class formation (P1) significantly increases warehouse storage capacity without requiring physical facility expansion.

3. **Co-Location / Affinity Value:** Storing products independently based solely on individual turnover overlooks order structure. Grouping strongly associated SKUs into contiguous rack zones (P3) yields an additional 15–20% reduction in picker travel distance.

4. **Rack Structural Safety Integration:** Modern dynamic slotting models (P5) prove that physical safety constraints (heavy items placed on lower shelf levels) can be directly integrated into allocation models without degrading picking travel efficiency.

6.2 Impact on the Industry

These OR developments directly inform the architectural design of commercial **Warehouse Management Systems (WMS)** (e.g., SAP EWM, Manhattan Associates, Blue Yonder). Automated slotting modules within modern WMS software now embed MILP and heuristic algorithms to perform periodic inventory re-slotting recommendations automatically.

Furthermore, in automated distribution centers utilizing AS/RS, Autonomous Mobile Robots (AMRs), and Shuttle Systems, continuous travel time models (P4) govern real-time dynamic cell assignment, directly boosting hourly order dispatch capacity.

6.3 Benefits to Business Operations and Supply Chains

- **Operational Cost Reduction:** Minimizing picker travel distance directly reduces warehouse labor costs, which typically account for over half of total facility operating budgets.
- **CapEx Deferral:** Optimizing slot sizing and space allocation increases effective cubic rack storage capacity by 20–35%, allowing expanding businesses to defer expensive warehouse physical expansions or leased space acquisitions.
- **Order Fulfillment Velocity:** Faster order picking cycles shorten order throughput times, enabling late order cut-off times for same-day and next-day e-commerce delivery guarantees.

7. Reflection: Strengths, Limitations and Alternative Possibilities

7.1 Strengths of the Models

- **Mathematical Rigour:** Models P1, P2, and P4 provide exact mathematical proofs and global optimality bounds for fundamental layout and zoning decisions.
- **Real-World Constraint Integration:** P3 and P5 move beyond simplified textbook assumptions by incorporating SKU affinity co-location, multi-dimensional slot geometry, and rack weight safety limits.
- **Computational Scalability:** The hybrid heuristics in P1, P3, and P5 enable the solution of large-scale industrial problems containing thousands of SKUs within operational time limits.

7.2 Limitations and Assumptions

- **Deterministic Demand Assumptions:** Models P1, P2, and P4 assume static, deterministic SKU demand rates over fixed planning horizons. In reality, e-commerce order patterns exhibit high volatility and short product life cycles.
- **Single-Item Slot Assumption:** Several models assume each storage slot contains only one SKU type, neglecting multi-SKU bin sharing or micro-fulfilment drawer partitioning.

- **Simplification of Congestion:** Most travel distance formulations evaluate single picker travel in isolation, ignoring aisle congestion, picker blocking, and narrow-aisle speed deceleration.

7.3 Alternative Possibilities for Improvement

1. **Stochastic and Robust Optimisation Extensions:** Deterministic demand parameters D_i and turnover frequencies f_i can be replaced with stochastic variables bounded by uncertainty sets, utilising Robust Optimisation (Mulvey et al., 1995) to create slotting assignments resilient to demand spikes.

2. **Integration with Vehicle Routing Problems (Picker Routing):** Space allocation (SLAP) and picker routing (TSP/VRP) are highly interdependent. Developing integrated **Location-Routing Warehouse Models** that solve storage assignment and optimal picker routing paths simultaneously represents a major opportunity for operational improvement.

3. **Reinforcement Learning for Real-Time Dynamic Re-Slotting:** Deep Reinforcement Learning (DRL) agents can be trained to adjust slot assignments dynamically in real-time as pickers complete tasks, balancing immediate picking efficiency against the labour cost of moving stock during active operations.

8. Conclusion

Managing warehouse space allocation efficiently is critical to modern supply chain operations. The evolution of Operations Research in this domain, moving from basic linear programming and index rules (such as COI) to complex mixed-integer programming, machine learning highlights the growing complexity and velocity of modern distribution centres.

The five benchmark studies analysed demonstrate that optimising warehouse space is not merely a task of packing items into available rack volume; it requires balancing picking travel distances, slot dimensions, product affinity, dynamic demand, and structural safety limits. As e-commerce growth demands faster order fulfilment and higher space efficiency, integrating advanced OR methodologies into automated Warehouse Management Systems will remain a primary driver of supply chain efficiency, cost reduction, and operational resilience.

References

- Chiang, D. M. H., Lin, C. P., & Chen, M. C. (2011). Data mining techniques applied to medical slotting and storage location assignment problem. *Expert Systems with Applications*, *38*(10), 13449–13456. https://doi.org/10.1016/j.eswa.2011.04.103
- de Koster, R., Le-Duc, T., & Roodbergen, K. J. (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
- Fumi, A., Scarabotti, L., & Schiraldi, M. M. (2013). The storage location assignment problem: A mathematical optimization approach. *International Journal of Engineering Business Management*, *5*(27), 1–9. https://doi.org/10.5772/56828
- Gu, J., Goetschalckx, M., & McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, *177*(1), 1–21. https://doi.org/10.1016/j.ejor.2006.02.025
- Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal storage assignment in automatic warehousing systems. *Management Science*, *22*(6), 629–638. https://doi.org/10.1287/mnsc.22.6.629
- Mulvey, J. M., Vanderbei, R. J., & Zenios, S. A. (1995). Robust optimization of large-scale systems. *Operations Research*, *43*(2), 264–281. https://doi.org/10.1287/opre.43.2.264
- Muppani, V. R., & Adil, G. K. (2008). A branch and bound algorithm for class based storage location assignment. *European Journal of Operational Research*, *189*(2), 492–507. https://doi.org/10.1016/j.ejor.2007.05.051

- Taha, H. A. (2017). *Operations research: An introduction* (10th ed.). Pearson.
- Yang, X., & Nguyen, T. (2020). Constrained clustering approach for class-based storage location assignment in high-velocity e-commerce warehouses. *IEEE Transactions on Automation Science and Engineering*, *17*(4), 1892–1904. https://doi.org/10.1109/TASE.2020.2982134