International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 1, 64-81, 2026
https://doi.org/10.33889/IJMEMS.2026.11.1.004

Mathematical Optimisation of 3D Container Loading Using Simulated Annealing and Ant Colony Algorithms

Penpark Mahanin
Department of Science and Mathematics,
Rajamangala University of Technology Isan, Surin Campus, Surin, Thailand.
E-mail: penpark.si@rmuti.ac.th

Ekrem Aljimi
Faculty of Applied Science,
Public University "Kadri Zeka", Gjilan, Republic of Kosova.
E-mail: ekrem.halimi@uni-gjilan.net

Thawatchai Boontan
Faculty of Science and Technology,
Rajabhat Maha Sarakham University, Maha Sarakham, Thailand.
Corresponding author: thawatchai.bo@rmu.ac.th

(Received on June 6, 2025; Revised on August 13, 2025; Accepted on September 4, 2025)

Abstract
The surge in online purchasing has intensified price competition, compelling businesses to reduce product costs and shipping fees to remain competitive in a rapidly expanding digital marketplace. For logistics service providers, an effective strategy for reducing shipping costs is to maximize the use of container storage capacity while minimizing wasted space, an approach referred to as the container loading problem. This classic optimisation challenge has wide applications in delivery companies, particularly due to the limited number of containers suitable for box packaging. As a result, manufacturers and postal delivery services have faced challenges in transporting and dispatching parcels efficiently. This highlights the need for an effective solution to the packing problem in rectangular containers. The proposed approach aims to reduce storage and shipping costs while minimizing processing and delivery times. To accomplish this, metaheuristic algorithms, particularly Simulated Annealing (SA) and Ant Colony Optimisation (ACO), were used in combination with the Axis Order Test (AOT) and Corner Point Placing (CPP). The performances of SA-AOT, SA-CPP, ACO-AOT, and ACO-CPP in terms of space utilisation and processing time were then compared. The results indicated that the ACO-CPP model was more effective than the others, achieving a maximum space utilisation of up to 98.19 per cent and having the fastest processing time (under 0.2 hours). The ACO-CPP model reduced packaging time and operational costs, offering a sustainable solution for logistics providers in the new era of e-commerce.

Keywords- Three-dimensional packing problem, Axis order test, Corner point placing, Simulated annealing, Ant colony optimisation.

1. Introduction

According to a Statista report, global retail e-commerce sales amounted to approximately 3.6 trillion U.S. dollars in 2024 and are forecast to reach around 5 trillion U.S. dollars by 2030. (Statista Research Department, 2025). E-commerce has grown rapidly in recent years, a trend that expanded following the COVID-19 pandemic. The crisis reshaped consumer behavior, making online shopping more prevalent as people increasingly valued convenience, speed, and simplicity. Thus, the demand for faster and more efficient logistics has increased, placing significant pressure on businesses, especially on transport and logistics providers. Therefore, there is a need to optimise operations and reduce costs while maintaining speed and quality of service. We can see that recent developments, such as the supply chain digital twin concept, have been shown to enhance supply chain resilience and decision-making through real-time simulation and optimization (Barykin et al., 2020). For example, practical applications of digital twins for modern logistics operations enabled the optimisation of the flow of goods and information across the end-to-end supply chain, and also supported the use of snapshots to forecast supply chain dynamics. However, inefficient pallet and container loading still persist, causing unnecessary costs due to underutilized space or degradation of products in storage, especially if environmental factors like humidity are not properly managed (Almasarwah et al., 2023). These issues become particularly important when considering the available shipping modes. Globally, commodities are primarily shipped via land, sea, and air transport. Among these modes, the sea transports more than 50% of the value and 80% of the volume of international trade due to its comparatively lower cost than air transport (Ferrari et al., 2023).

However, shipping by sea has several constraints, such as limited routes, operating times, container and port capacity, fuel costs, seasonal changes shifts in demand, and loading capacity. Since shipping can result in costly delays or dispatch of underutilized containers leading to wasted resources, one of the most critical constraints is loading capacity, in that it is significantly affected by the inaccurate estimation of the number of containers. The container loading problem, a subset of the broader packing problem, is an optimisation challenge that involves arranging items efficiently within a container, truck, or pallet (Zhao et al., 2016; Pietri et al., 2021). As such, the relationship between logistics and container loading is critical to the overall efficiency of the supply chain. Efficient container loading has a direct impact on the logistics system since it helps to optimise transportation efficiency, lowers shipping costs, and reduces fuel consumption (Doukas et al., 2021). In addition, organized packing methods lead to improved handling, reduced shipping time, and better environmental performance. These are all areas of primary focus in green logistics, which is widely accepted around the world. Thus, the container loading problem is considered one of the most important factors in improving overall logistics service performance (Jugović, 2020; De Souza et al., 2022; Chien et al., 2024). To achieve both effectiveness and sustainability in meeting customer demands, organizations are increasingly embracing new technologies such as metaheuristic algorithms to address difficult problems of optimal packing. Given these challenges in logistics optimisation, container loading remains one of the most critical issues. To address these issues, many researchers have become increasingly interested in using computational methods to improve operations management. Among these, the metaheuristic approach is widely employed to solve optimisation problems, particularly packing problems.

Metaheuristic techniques are algorithms for optimisation that are used to find the optimal solution to complex problems. Exact solutions are costly to compute and may not guarantee the discovery of an optimal solution will be found (Tomar et al., 2023; Houssein et al., 2025). Consequently, researchers increasingly prioritize metaheuristic approaches to derive high-quality approximations for complex optimisation challenges. This has led to an increasing number of studies focusing on how metaheuristic algorithms can be used to address container loading scenarios. For instance, in 2021, Penpark et al. (2021) employed a heuristic method combining a genetic algorithm (GA) and simulated annealing (SA) to manage food box packing problems in rectangular containers. The findings indicated that the proposed method significantly reduced packaging time and decreased the total cost of food container distribution (Penpark et al., 2021). Romero et al. (2023) pioneered a hybrid approach to address real-world bin packing problems using quantum annealing techniques. The study introduced a hybrid quantum-classical approach to solve real-world 3D Bin Packing Problems (Q4RealBPP) with realistic constraints. Their findings revealed that Q4RealBPP effectively addressed all generated instances, including various real-world scenarios. The final two occurrences, 3dBPP_11 and 3dBPP_12, were particularly important since they activated all defined limitations (Romero et al., 2023). Later in the same year, Yang et al. (2024) proposed a two-layer heuristic approach for solving container packaging problems. Their research introduced an innovative technique for designing container sizes to better accommodate packaged items. Moreover, 3D-MBSBPP and 3D-ODP techniques have been combined and used to solve a multidimensional bin design and packing problem, developing a two-layer heuristic method. In this framework, the inner layer utilized a deterministic constructive heuristic to generate effective solutions, while the outer layer applied genetic algorithms (GA) to design appropriate container types. Simulation results demonstrated the effectiveness of this combined approach, providing better solution quality and reducing the cost of bins. These findings represent significant progress in both applied research and relevant industries (Yang et al., 2024).

The research discussed above has potential applications in the land transportation industry, particularly in addressing the container loading problem. As indicated in previous studies, the global export sector plays a critical role in international economic development and the expansion of global trade. The continued growth in export activities is largely driven by advancements in manufacturing infrastructure and technological innovation including IoT devices (e.g., sensors, cameras, smart locks) for cargo tracking and smart sensors. In addition, smart sensors in the supply chain can improve operating efficiency through automation, decrease repair costs and maintenance downtime through better monitoring, and perform real-time inventory tracking with improved demand planning. Moreover, the increasing demand for high-quality products and services has broadened access to global markets. The rapid increase in global e-commerce further contributes to this growth, presenting both new opportunities and challenges to the export and logistics industry. Among these challenges, efficient container packing management is a critical concern. Poorly managed packing operations can lead to financial losses, delayed delivery, and compromised performance. In addition, unhappy customers may not place future orders, and may warn family, friends and social media followers not to utilize the company. Such issues may negatively affect the credibility of exporting countries and diminish consumer confidence in products delivered through international trade. Improving logistical efficiency has become a necessity for companies aiming to meet rising consumer expectations. In response, e-commerce companies and other businesses around the world have been developing new features and services. These strategies aim to make the online shopping experience as smooth and convenient as in-store purchasing. However, the rapid development of online transactions has created logistical issues, including delivery delays and rising shipping transit costs. Projections indicate that transportation costs will continue to rise in the coming decades (Raj et al., 2024; Tjandra et al., 2024). This is predicted to drive companies around the world into adopting advanced cargo planning solutions for improved efficiency, security, and cost-effectiveness in transportation management. The implementation of initiatives such as container packing optimisation, route optimisation, and sustainable logistics practices will be vital in addressing the challenges within the global export industry. Businesses engaged in international trade must enhance these areas to meet the growing demand for fast, reliable shipping services.

To solve such logistical problems, global logistics companies are employing modern technologies in packing processes to reduce operational costs, cargo frequency and environmental impact. Current studies have also focused on improved heuristic approaches based on dynamic corner fitness and spatial knowledge reuse to enhance the efficiency of container loading operations. (Fang et al., 2024; Liu & Jiang, 2024).

This study proposes a fast and effective method for solving packing problems in rectangular containers by combining metaheuristic algorithms, particularly Simulated Annealing (SA), and Ant Colony Optimisation (ACO). The novelty of these hybrid approaches lies in the integration of metaheuristic algorithms with distinct placement strategies —Axis Order Test (AOT) and Corner Point Placing (CPP). This model enables detailed cross-comparison of hybrid strategies. Our approaches operate without the need for prior training and can be used for real-time logistics applications. Additionally, combining pheromone-based learning with simulated annealing improves the convergence rate and optimises packing efficiency without adding to the computational cost.

This article is organized as follows: Section 2 describes the formulation for the three-dimensional packing problem. Section 3 outlines the main structure of the proposed metaheuristic algorithm. The empirical results and discussion of both solutions are discussed in Section 4. Finally, Section 5 summarises the key findings of the study.

2. Mathematical Formulation for the Three-Dimensional Packing Problem

This section presents the mathematical formulation for the three-dimensional container loading problem, which involves packing a set of items (i.e., boxes) within a container such that there is no overlap and all items remain within the container's boundaries. The proposed method builds upon concepts originally developed for the analogous two-dimensional packing problem, and then applies them to the three-dimensional packing problem (3D-BPP) through substantial changes and incremental improvements.

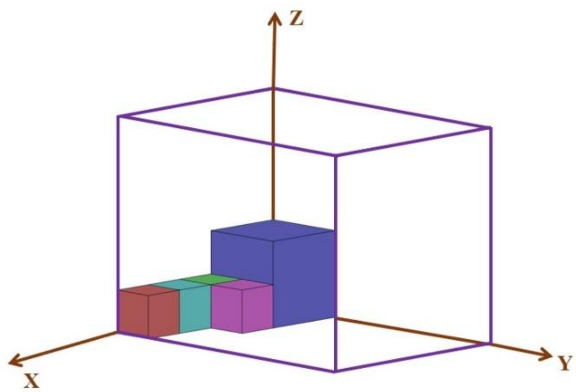


Figure 1. The axis order test (AOT).

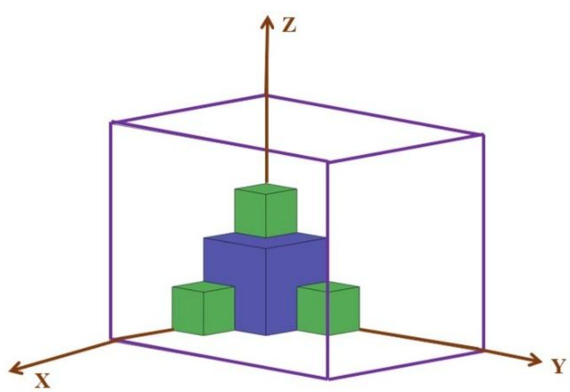


Figure 2. Corner point placing (CPP).

Therefore, in this study, we consider the three-dimensional packing problem (3D-BPP) using the Axis Order Test (AOT), as shown in **Figure 1**, and Corner Point Placing (CPP), as shown in **Figure 2**, with the following conditional constraints: (i) the items have to be placed without overlapping, (ii) the items should be packed with their edges parallel to the container's borders, and (iii) the items are cuboid and cannot be rotated, because in practical logistics applications, many items must be kept upright to avoid damage or prevent leakage such as food boxes, liquid containers, and medical equipment. Therefore, condition number 3 should maintain a fixed orientation to contribute to increased safety and optimise the cargo unloading process. The description of the two methods is as follows:

2.1 Axis Order Test (AOT)

The Axis Order Test (AOT) algorithm is a strategic approach for optimizing the placement sequence of three-dimensional rectangular items within a container. Algorithm 1 contains a pseudo-code representation of the AOT algorithm, which is used to solve three-dimensional packing problems. This concept is inspired by the bottom-left-fill metaheuristic algorithm used in two-dimensional packing scenarios. Each item (i.e., box) is initially positioned at the top right-hand corner of the sheet and then moved successively downwards and leftwards as far as possible until the item (i.e., box) is placed in a stable position without overlapping, as shown in **Figure 1**. In this research, each item $i = \{1, 2, 3, \dots, N\}$ is defined by its dimensions:
(Rx_i, Ry_i, Rz_i).

where, x_i, y_i, z_i are the coordinates of the bottom-left-front corner of item i,
Rx_i is the width of item (i.e., box) i
Ry_i is the length of item (i.e., box) i
Rz_i is the height of item (i.e., box) i
The dimensions of the container are denoted as:
(X, Y, Z).

where,
X is the width of the container i,
Y is the length of the container i,
Z is the height of the container i,
and N represents the total number of items (i.e., boxes).
The purpose of the AOT is to design an optimal packing configuration so that:

$$\sum_{i=1}^{N} V_i \leq V_c \quad (1)$$

Here, V_c is the volume of the container $X \times Y \times Z$.
V_i is the volume of item (i.e., box) i, which is $Rx_i \times Ry_i \times Rz_i$.
The details of AOT are presented in Algorithm 1 and we can consider the optimisation formulation as follows:

***Objective Function*:** Maximize the total volume of packed items

$$V = max \sum_{i=1}^{N} \delta_i \cdot V_i \quad (2)$$

where, $\delta_i \in \{0,1\}$ indicates whether item (i.e., box) i is placed (1) or not (0), V is the total volume of packed items.

***Constraints*:**
(i) *Container boundaries* (*for all* i):

$$X \geq x_i + Rx_i \cdot \delta_i,\ Y \geq y_i + Ry_i \cdot \delta_i, Z \geq z_i + Rz_i \cdot \delta_i \quad (3)$$

(ii) *Non-overlapping items* (*for all* $i \neq j$):

$$(x_j \geq x_i + Rx_i) \vee (x_i \geq x_j + Rx_j) \vee (y_j \geq y_i + Ry_i) \vee (y_i \geq y_j + Ry_j) \vee (z_j \geq z_i + Rz_i) \vee (z_i \geq z_j + Rz_j) \quad (4)$$

This ensures that item (i.e., box) i and j do not overlap in three-dimensional space.

(iii) *Binary decision variables* (*for all* i): $\delta_i \in \{0,1\}$.

Algorithm 1 Axis Order Test (AOT) Packing Algorithm

Input: A container dimensions (X, Y, Z), and a set of N boxes with dimensions (Rx_i, Ry_i, Rz_i)
Output: Coordinates $c_i = (x_i, y_i, z_i)$ for each packed item
for $k = 1$ **to** N **do**
    Initialize candidate position $c_i = (0,0,0)$of container
    while item k not placed **do**
        if item k fits at c without overlapping **then**
            Place item k at $c_k = c$
            Mark item k as placed
            break
        end if
        Move c along Z-axis
        if Z-direction fully placed **then**
            Reset Z, move along Y-axis
        end if
        if Y-direction fully placed **then**
            Reset Y and Z, move along X-axis
        end if
        if All feasible directions tested **then**
            Mark item k as unplaced; **break**
        end if
    end while
end for
Return All feasible positions $\{c_1, \ldots, c_n\}$

2.2 Corner Point Placing (CPP)

The Corner Point Placing (CPP) method is based on the intersection of three-dimensional planes and identifies potential placement positions by considering the plane sections formed by the top, bottom, left, and right sides of items already placed in the container, as illustrated in **Figure 2**. All previously packed items are fixed in position and cannot be moved further down, top, up, or backward. Besides, whenever a new item (i.e., box) is added to the container, its position should not overlap with an already packed item . As a consequence, the boundary between the empty and occupied spaces is defined by the set of all corner points within the container. Let N denote the number of items in CPP method. The first item (i.e., box) is positioned at the origin coordinates $X_1 \times Y_1 \times Z_1$. Following this placement (i.e., box), three new points are generated at the edges of the item: $((0 + X_1, 0,0), (0,0 + Y_1, 0), (0,0,0 + Z_1))$, excluding the original corner $(0,0,0)$. Subsequent items are placed at these new positions. If an item cannot be placed at a given point, alternative items are considered until a suitable one is found. This technique reduces unused space and increases the overall compactness of the packing arrangement. The pseudo code for determining the placement order is presented in Algorithm 2:

Algorithm 2 Corner Point Placing (CPP) Algorithm

Input: A container dimensions (X, Y, Z), and a set of N boxes with dimensions (Rx_i, Ry_i, Rz_i)
Output: Placement coordinates $c_i = (x_i, y_i, z_i)$ for each box
Initialize corner point set $C_1 = (0,0,0)$
for $k = 1$ **to** N **do**
    placed = false
    for all c in C_k do
        if box k fits at c without overlap or overflow **then**
            Place box k at $C_k = c$
            C_{k+1} update the new corner points generated by placing
            Remove the used corner point from C_{k+1}
            placed = true; **break**
        end if
    end for
    if not placed **then**
        k = k + 1
    end if
end for
Return $\{c_1, c_2, \ldots, c_n\}$

Let $i \in \{1, 2, \ldots, N\}$ be the set of items (i.e., boxes) to be packed, (Rx_i, Ry_i, Rz_i) are the dimensions of item (i.e., box), x_i, y_i, z_i are the coordinates of the bottom-left-front corner of item, and $C_k \subset \mathbb{R}^3$ is the set of available corner points at step k. The algorithm selects a corner point $c_k = (x, y, z) \in C_k$ at each packing step k such that:

(i) Feasibility constraint- Each item must lie entirely within the container's boundaries:

$$X \geq x_i + Rx_i, \quad Y \geq y_i + Ry_i, \quad Z \geq z_i + Rz_i \tag{5}$$

(ii) Non-overlapping constraint- The item should not overlap with any previously placed item (i.e., box) j, for all $j < i$:

$$(x_j \geq x_i + Rx_i) \vee (x_i \geq x_j + Rx_j) \vee (y_j \geq y_i + Ry_i) \vee (y_i \geq y_j + Ry_j) \vee (z_j \geq z_i + Rz_i) \vee (z_i \geq z_j + Rz_j) \tag{6}$$

(iii) Corner preference rule – Among all feasible corner points, choose the one minimizing:

$$\min_{(x,y,z) \in C_k} z + \beta y + \gamma x \tag{7}$$

Here, β, γ are the small positive constants used to break ties and prefer front-left-lower positions .When an item is placed at $(x_i, y_i, z_i) = c_k$, the corner point list C_{k+1} is updated by adding new corner points generated from the exposed corners of the newly placed item.

3. Metaheuristics Method

This study proposes a model utilising Simulated Annealing (SA) and Ant Colony Optimisation (ACO) to derive efficient and optimal solutions to packaging problems in containers or similar scenarios. A novel methodological contribution of this study is the structured integration of each metaheuristic algorithm —Simulated Annealing (SA) and Ant Colony Optimisation (ACO)—with two placement strategies: Axis Order Test (AOT) and Corner Point Placing (CPP). The performance of SA-AOT, SA-CPP, ACO-AOT, and ACO-CPP was then compared. The aim of this work is to minimise resource usage or costs associated with packaging and transportation processes, which may involve reducing storage space in containers, decreasing delivery times, or lowering transportation expenses. The details of SA and ACO are illustrated in the following sections.

3.1 Simulated Annealing (SA)

Simulated Annealing (SA), introduced by Kirkpatrick et al. (1983), is a probabilistic metaheuristic inspired by the annealing process in metallurgy, where materials are heated and slowly cooled to minimise their structural energy. This method approximates the global optimum of a given function by processing many local minimums. In combinatorial optimisation problems, such as three-dimensional packing, SA explores large and non-convex solution spaces by allowing, with probability, inferior solutions to avoid local optima and search for a global optimum. Generally, this algorithm employs an iterative movement based on a changeable thermal parameter, approximating the metal annealing procedure (Henderson et al., 2003; Sahab et al., 2013). The concept of simulated annealing in combinations is illustrated in the flowchart in **Figure 3**.

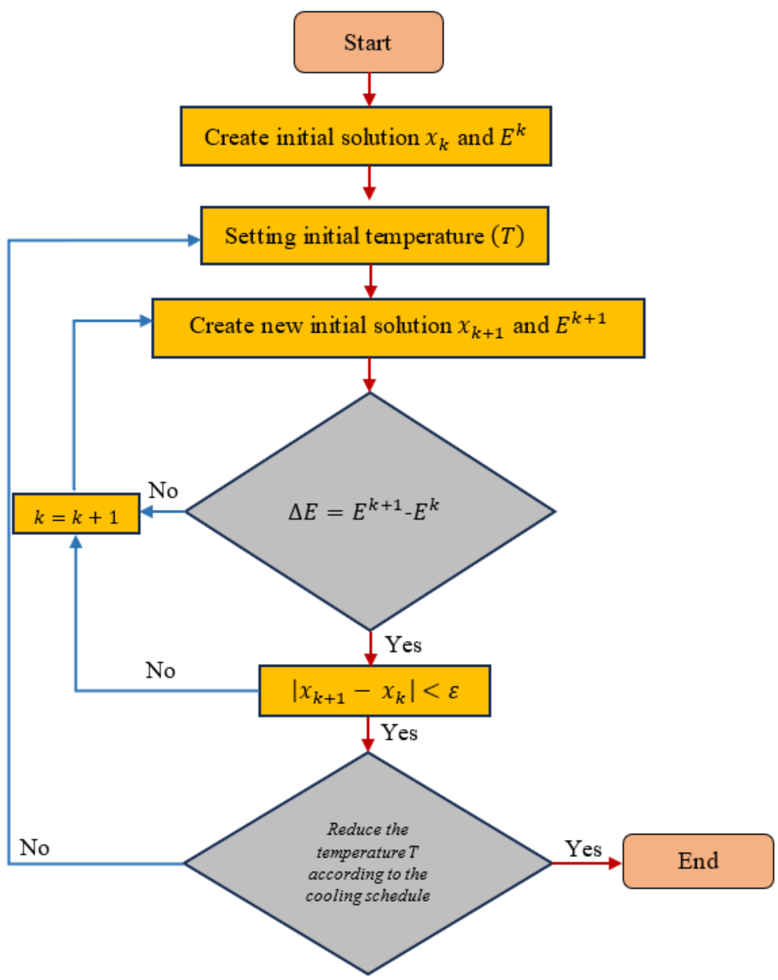


Figure 3. Illustration of the flowchart of the simulated-annealing algorithm (SA).

Considering the flowchart in **Figure 3**, let E_k and E_{k+1} denote the objective function values (or energy levels) at iterations k and $k+1$, respectively. The action to accept the next point or the action at iteration $k+1$ depends on the change in the objective function values at both points. As a consequence, the system's energy changes from E^k to E^{k+1}, which is defined as $\Delta E = E^{k+1} - E^k$. The value of ΔE is employed in the computation of the probability distribution $P = min\left(1, exp(\Delta E/T)\right)$. Subject to the condition that ΔE is non-positive, indicating a decrease in the objective function value, the probability P is set to 1. The acceptance probability in the system for moving from state x_k to x_{k+1} is given by :

$$P = \begin{cases} 1 & if, \Delta E > 0 \\ exp\left(\frac{\Delta E}{T}\right) & if, \Delta E \leq 0 \end{cases} \quad (8)$$

where, $T > 0$ is the current temperature. Therefore, the solution is accepted if $exp(\Delta E/T)$ is accepted, where ΔE is positive (Delahaye et al., 2019). The value of the objective function at any point within the considered domain is denoted as follows:

$$E^k = \sum_{i=1}^{N} \delta_i \cdot V_i \quad (9)$$

where, $\delta_i \in \{0,1\}, \delta_i = 1$ if the item i is placed, $\delta_i = 0$ if the item i is not placed, V_i is the volume of item and i, E_k is the unused space. In addition, the cooling schedule can be described as:

$$T_{k+1} = \alpha T_k \text{ , } \alpha \in (0,1) \quad (10)$$

From the above equation, the typical value of α was chosen between 0.90-0.99 (McKendall & Dhungel, 2024). In this study, we selected $\alpha = 0.95$ based on preliminary experiments that balanced convergence speed and solution quality. The process continues until $T_k \leq T_{min}$, where T_{min} was determined empirically based on problem size.

3.2 Ant Colony Optimisation (ACO)

In this metaheuristic approach, the Ant Colony Optimisation (ACO) algorithm is also considered for solving the optimal packing problem. This method was introduced in 2006 by Dorigo et al. (2006). The basic concept behind ACO originated from the ability of ants to find the most optimal path from their nest (N) to food sources (F) (Levine & Ducatelle, 2004). As ants travel, they deposit a chemical trail (pheromone) along their path. This pheromone trail serves as a guide for other ants to the target point. The path traveled by one ant is determined by the amount of pheromone deposited by prior ants. Ants employ pheromone deposit to find the most followed path, which is frequently the best or near-best option. Furthermore, as time passes, this chemical substance becomes less effective, and the amount left by one ant is dependent on the amount of food discovered and the number of ants using this pathway (Peng et al., 2005).

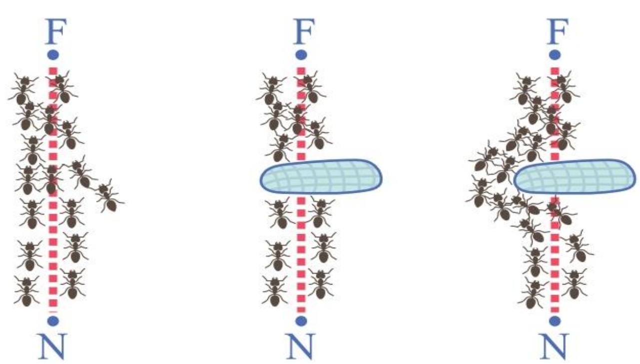


Figure 4. Illustration of ants encountering an obstacle.

Figure 4 shows an example of ants encountering an obstacle: (a) Ants are moving along a trail between their nest (N) and the food source (F). When an obstacle blocks the original path, two new options for travelling, left or right of the obstacle, emerge; (b) The ants' choice is influenced by the intensity of the pheromone trails left by previous ants. A higher pheromone concentration on the left path provides a stronger stimulus, increasing the probability that subsequent ants will choose it; (c) Over time, more pheromones accumulate along the shortest (or nearly optimal) path. The fundamental principles of ACO are detailed in the following steps.

(i) Initialization

The initialization phase of the ACO consists of two parts :representing the problem as a graph, and randomly placing ants on the graph's nodes. The problem is modeled as a graph $G = \langle N, E \rangle$, where each node (N) indicates a possible position (e.g., a corner point) for placing a box within the container, and the connections between these positions are shown by edges (E). The second step is to place a number of ants on randomly chosen nodes upon which each ant constructs a solution by moving from node to node based on a probabilistic node transition rule.

(ii) Node transition rule

In the context of the packing problem, each ant builds a feasible solution by moving from node to node in the packing graph, where each node represents an item placement configuration (position and orientation). Rather than making purely random decisions, ants rely on both the pheromone intensity and the desirability of the placement based on some heuristic factor (e.g., volume fit or spatial efficiency). In particular, for an ant currently located at node i, the probability of moving to a feasible next node $J \in N_i^{feasible}$ is defined as:

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \epsilon N_i^{feasible}} [\tau_{ik}]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \tag{11}$$

Here, τ_{ij} is the amount of pheromone deposited on edge $(i, j), \eta_{ij}$ is the heuristic desirability of choosing node j, typically based on volume fit or spatial efficiency, and the parameters controlling the influence of pheromone and heuristic are α and β respectively. In this study, we set $\alpha = 1$ and $\beta = 2$ based on widely accepted values in combinatorial optimization literature (Stützle et al., 2012). Infeasible nodes such as placements resulting in overlap or boundary violations are excluded from the candidate set $N_i^{feasible}$. This ensures that each ant generates only valid packing sequences. For other potential state transitions, the trail level and attractiveness are represented by $\tau(i, k)$ and $\eta(i, k)$.

(iii) Pheromone updating rule

Using the node transition rule iteratively, each ant moves along the edges of the graph from node to node, constructing a complete solution to the packing problem. Once all ants have generated their respective solutions, one full cycle of the ant colony algorithm is considered complete. At the end of each cycle, after all ants have constructed their respective packing solutions, the pheromone updating rule is applied to adjust the pheromone intensity on each edge. The pheromone on each edge (i, j) is adjusted using the following update rule:

$$\tau(i, j) \leftarrow (1 - \rho) \cdot \tau(i, j) + \sum_{k=1}^{n} \Delta\tau_k(i, j) \tag{12}$$

Here, n is the total number of ants, $\rho \in (0,1)$ denotes the rate of pheromone evaporation which is considered at 0.5 and $\Delta\tau_k(i, j)$ denotes the amount of pheromone that the k_{th} ant laid along the edge (i, j)

during the current cycle. If total the cost (e.g., unused volume or packing inefficiency) of the solution is defined as L_k , which is generated by the k_{th} ant, then $\Delta\tau_k(i,j)$ can be determined by:

$$\Delta\tau_k(i,j) = \begin{cases} \frac{Q}{L_k}, if\ edge\ (i,j)\ is\ used\ in\ solution\ by\ ant\ k \\ 0\ , otherwise \end{cases} \quad (13)$$

Here, a constant parameter is represented by Q (Stützle et al., 2012). Pheromone evaporation serves a similar function to forgetting in learning systems. By gradually reducing the influence of previously accumulated pheromone trails over time, it prevents the algorithm from converging too early to a poor region. Meanwhile, pheromone reinforcement by $\Delta\tau_k(i,j)$ enhances the directions that lead to better quality solutions, leading the searching process to more promising areas.

(iv) Stopping criterion

The stopping criterion for the Ant Colony Optimization is defined by either reaching the maximum number of operating cycles or the CPU time limit. ACO can be adapted to a wide range of optimization problems. In this research, ACO is applied to address packing problems that require significant time and resources, such as arranging items within containers of different sizes and shapes. The use of ACO in this scenario illustrates its suitability and effectiveness for efficiently solving complex problems. The procedural steps of the ACO algorithm are summarized in **Figure 5**.

4. Results and Discussion

This study aimed to evaluate the efficiency of the Simulated Annealing (SA) and the Ant Colony Optimisation (ACO), both integrated with the Axis Order Test (AOT) and Corner Point Placing (CPP) techniques. Three test scenarios were conducted to validate the proposed method. To assess probabilistic outcomes, a set of items and containers were designed based on standard shipping containers and parcel box dimensions commonly used in Thailand. In this scenario, the weight of the items and the pressure they might exert on each other were not considered. All items and containers were rectangular in shape, and item orientation was fixed, meaning the items could not be rotated. The boxes used in the experiments varied in size to reflect realistic packing scenarios.

Table 1. Summary of box instances and average volumes.

Problem	No. of boxes	Box size variants	Container size (W×L×H)	Avg. box volume
1	75	3	8×19×8	30.33
2	100	4	8×19×8	31.75
3	150	5	8×19×8	30.20

SA and ACO, applied to three-dimensional packing problem, were compared in order to determine the most efficient packaging method. In this research, we used a container with approximate dimensions of 8 feet (2.43 m) in width, 19 feet (5.79 m) in length, and 8 feet (2.43 m) in height. The problem was divided into three phases: Problem 1 consisted of 75 items, with 25 boxes for each of the three sizes: 2×3×5, 3×5×3, and 2×4×2. Problem 2 consisted of 100 items, with 25 boxes for each of the four sizes: 2×4×2, 3×4×3, 2×3×5, and 3×5×3. Problem 3 involved packing 150 items, with 30 boxes for each of the five sizes: 2×4×2, 3×4×3, 2×3×5, 3×5×3, and 2×3×4. The average box volumes for Problems 1, 2, and 3 were 30.33, 31.75, and 30.20, respectively, as shown in **Table 1**.

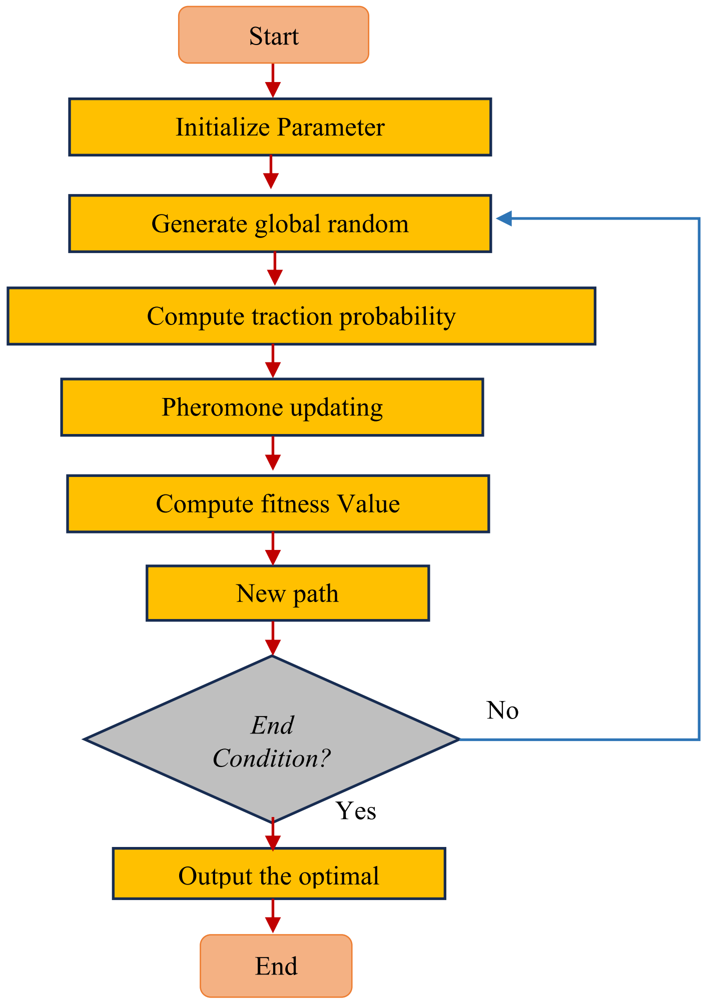


Figure 5. Illustration of flowchart of ant colony optimisation (ACO).

4.1 A Comparative Analysis of SA and ACO in 3D Bin Packing

Table 2 illustrates a comparison of four hybrid models: SA–AOT, SA–CPP, ACO–AOT, and ACO–CPP. These models combined two metaheuristic approaches SA and ACO — with two placement strategies: AOT and CPP. The test problems in each experiment were solved on a PC running Microsoft Windows 10 with 32 GB of RAM and an Intel Core i5 CPU running at 2.50 GHz. To guarantee the robustness and dependability of the results, each experiment was repeated 10 times. This article reports the statistical measures of the average (mean) and standard deviation (Mean ± Std) across these runs. In all problem cases, the ACO – CPP model obtained the best performance. Across the three problems, the model recorded the shortest average CPU times of 0.07 ± 0.01, 0.09 ± 0.01, and 0.17 ± 0.01 hours and also maintained the highest average container utilisation rates of 91.67% ± 0.01, 92.91% ± 2.82, and 98.19% ± 0.29, for Problems 1, 2, and 3, respectively, as shown in **Figure 6**. These results are based on 10 independent runs. Moreover, the memory usage for this problem was less than 2600 KB. These superior results are attributed to two key factors: (1) the CPP method provides a more flexible way to place the items by updating available Corner Point Placing, leading to more efficient use of space; and (2) the ACO algorithm, which bases its decisions on accumulated pheromone trail data, enables more adaptive and experience-driven optimisation compared to the random perturbations used in SA. Therefore, the combination of pheromone-guided search and corner point placement reduced search space and sped up convergence, outperforming the SA–AOT, SA–CPP, and ACO–AOT models, which required longer runtimes and slightly higher memory consumption.

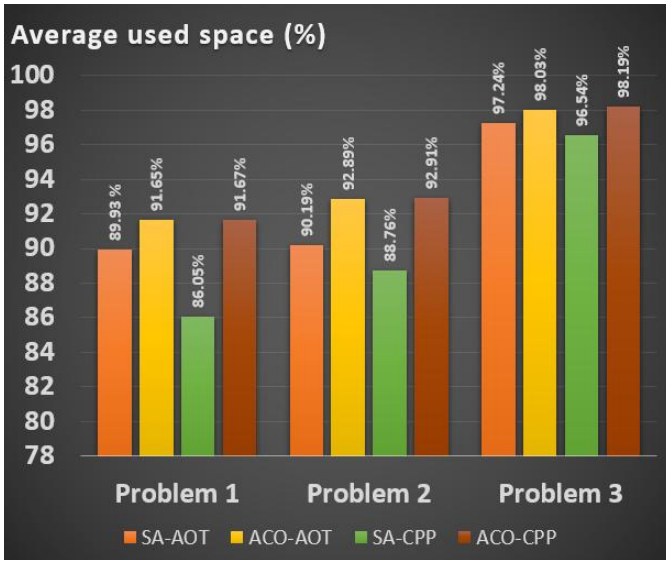


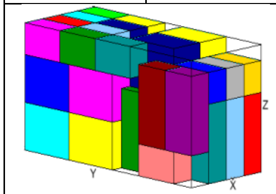
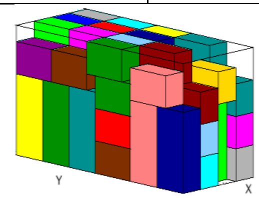
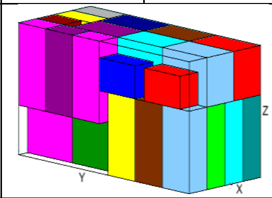
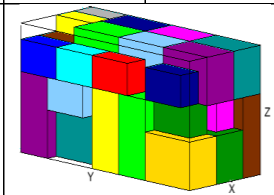
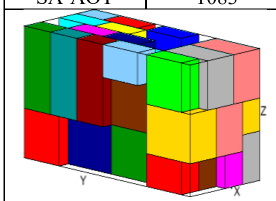
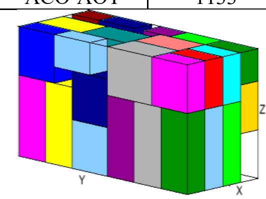
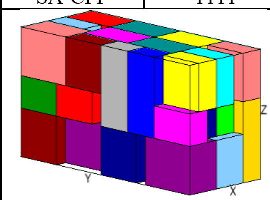
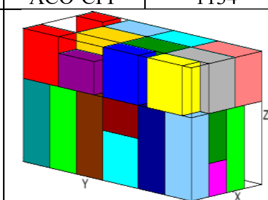
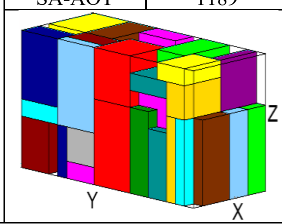
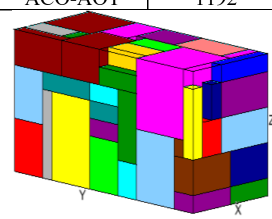
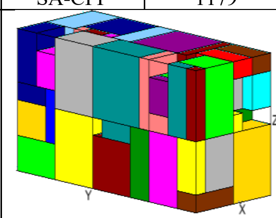
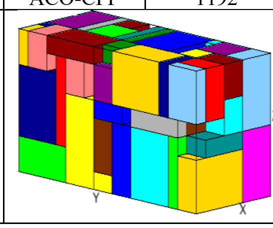
Figure 6. Comparison of space utilization across algorithms.

Table 2. Comparison of simulated annealing (SA) and ant colony optimisation (ACO) across packing problems.

Model	Average CPU time (h) ± SD	Average used space (%) ± SD	Memory usage (KB)
Problem 1 SA-AOT	1.27 ± 0.09	89.93 ± 0.01	1958
ACO-AOT	0.73 ± 0.04	91.65 ± 0.01	1972
SA-CPP	0.10 ± 0.01	86.05 ± 0.01	1954
ACO-CPP	0.07 ± 0.01	91.67 ± 0.01	1921
Problem 2 SA-AOT	1.85 ± 0.11	90.19 ± 1.23	2190
ACO-AOT	0.74 ± 0.06	92.89 ± 1.96	2176
SA-CPP	0.11 ± 0.01	88.76 ± 2.28	2180
ACO-CPP	0.09 ± 0.01	92.91 ± 2.82	2150
Problem 3 SA-AOT	2.06 ± 0.13	97.24 ± 0.01	2526
ACO-AOT	1.21 ± 0.23	98.03 ± 0.47	2588
SA-CPP	0.28 ± 0.31	96.54 ± 0.01	2566
ACO-CPP	0.17 ± 0.01	98.19 ± 0.29	2480

On the other hand, ACO-AOT produced acceptable results, achieving a similar utilisation of 91.65 ± 0.01, 92.89 ± 1.96, and 98.03 ± 0.47, for Problems 1, 2, and 3, respectively (see **Figure 6**). However, it required a significantly longer processing time (over 0.73, 0.74, 1.21 hours for Problems 1, 2, and 3, respectively) and produced less compact packing arrangements, particularly in larger problem cases. The example of best packing performance is in **Table 3**. This inefficiency can be attributed to the AOT method, which scans items in a linear order, often resulting in fragmented and underutilized space. Generally, the combination of CPP and ACO yielded higher performance in terms of efficiency and accuracy, indicating the strength of integrating heuristic-driven search strategies with placement-based techniques.

Table 3. Sample packing outcomes from 10 independent runs for three case studies using a container of size 8 ft × 19 ft × 8 ft.

Problem	Method 1	Volume	Method 2	Volume	Method 3	Volume	Method 4	Volume
1	SA-AOT	1088	ACO-AOT	1096	SA-CPP	1097	ACO-CPP	1099
								
2	SA-AOT	1083	ACO-AOT	1133	SA-CPP	1111	ACO-CPP	1134
								
3	SA-AOT	1189	ACO-AOT	1192	SA-CPP	1179	ACO-CPP	1192
								

4.2 Comparative Analysis of Related Work in 3D Bin Packing

In this section, we compare our proposed model (ACO-CPP) with previous studies to evaluate its suitability for application in the transportation industry. **Table 4** provides a comparison overview of ACO-CPP and other methods in 3D bin packing.

Zuo et al. (2022) studied the three-dimensional bin packing problem (3D-BPP) involving irregularly shaped items using a constructive heuristic algorithm. Their research introduced a new 3D bin packing model that accommodated rectangular-shaped and non-rectangular items. The main objective was to optimize packing performance and reduce the operational costs for the company logistics. Their approach achieved an average utilization (~87.2%), which was less than some deep learning-based techniques. Nevertheless, the method demonstrated practical applicability, especially in real-world fresh food delivery (Zuo et al., 2022).

Yang et al. (2024) addressed three-dimensional bin design and packing problem (3D-BDPP) using a two-layer heuristic approach. This approach integrates an outer heuristic framework and an inner deterministic constructive heuristic to determine optimal container dimensions and generate effective box placements. The goal of this research was to reduce packing costs. Benchmarking data was derived from an e-commerce company. Their findings indicated that the average packing utilization was around 86.1%. Moreover, redesigning bin sizes significantly reduced overall costs. The proposed two-layer heuristic proved effective for solving the 3D-BDPP, with GA providing superior solution quality, while differential evolution algorithm (DEA) achieved faster computation times (Yang et al., 2024).

Nguyen & Nguyen (2023) introduced a novel method for online three-dimensional bin packing using space splitting and merging technique. This approach was specifically designed for robotic packing applications and demonstrated real-time capability, making it suitable for industrial use. The results indicated that the method was both fast and straightforward due to its simple structural design, achieving an average space utilisation of up to 83.0% (Nguyen & Nguyen, 2023). More recently, Zhang et al. (2024) proposed a technique combining Generative Adversarial Networks (GAN) with GA to generate efficient packing sequences. The purpose of this study was to produce high quality solutions while enhancing exploration and exploitation capabilities. Their model achieved approximately 90% space utilization. However, it required long training times and was sensitive to parameter selection (Zhang et al., 2024). In the same year, Wong et al. (2024) employed a hybrid heuristic approach integrated with Proximal Policy Optimization (PPO) to maximize space utilization and enable real-time packing in actual circumstances using robot manipulators. Their results showed a space utilization of up to 83%. In addition, their concept supported real-time interaction in robotic settings. However, this approach depended on environment-specific reinforcement learning frameworks, making it less flexible in unseen instances (Wong et al., 2024).

In comparison to the aforementioned approaches, the ACO-CPP model proposed in this study offers several distinct advantages. To begin with, this approach achieved approximately 92% of space utilization without the need for any model training. Therefore, this makes it simple, effective for immediate implementation in real-world applications. In addition, unlike black-box neural network-based approaches, the algorithm maintains transparency and interoperability. Furthermore, its design enables flexible item placement by utilizing corner point-based strategies alongside pheromone-guided search, which facilitates rapid identification of near-optimal packing configurations. This adaptability makes the model suitable for containers of varying sizes and shapes. The operational benefits of ACO-CPP model are particularly evident in real-time decision-making problems with minimal computational load. This also reflects our selection of SA and ACO in this work. This is because it offers a strong balance between performance and practical implementation. These methods are well-suited for real-world logistics applications and provide a strong baseline for hybrid optimization. While many metaheuristic algorithms have gained traction in recent years, such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), or deep reinforcement learning (DRL), they often require complex parameter settings, extensive training, higher computational overhead, or difficulties in designing and debugging, which may limit their suitability in real-world logistics environments.

Table 4. Comparison of ACO-CPP with related work in 3D bin packing based on space utilisation, training requirements, and processing time.

Method	Used space	Training	Time	Notes	References
ACO + CPP	~92.0%	No	Low	Fast, interpretable, real-time	Our Work
Constructive Heuristic	~87.0%	No	Moderate	Designed for irregular shaped items with packing	Zuo et al. (2022)
Two-layer heuristic (GA/DE outer,heuristic inner)	~86.1%	No	Moderate	Bin design + 3D packing with support area space structure	Yang et al. (2024)
Space Splitting+ Merging	~83.0%	No	Low-Moderate	Online packing using space splitting. Good placement flexibility.	Nguyen & Nguyen (2023)
GAN + GA	~90.0%	Yes	High	Needs GAN training good convergence	Zhang et al. (2024)
HHPPO (Heuristic + PPO)	~83.0%	Yes	Moderate-High	Requires RL env setup	Wong et al. (2024)

5. Conclusion

This study presents an innovative approach to the container loading problem, a key challenge in logistics optimisation. By integrating Simulated Annealing (SA) and Ant Colony Optimisation (ACO) with the Axis Order Test (AOT) and corner point placement (CPP), we aimed to improve container packing efficiency and reduce transportation storage costs. Three problem instances were evaluated across 12 test cases. Each test case was assessed over 10 independent runs (**Tables 2** and **3**). The ACO-CPP method outperformed others, achieving the highest space utilisation (over 85%) and the shortest processing times –0.07, 0.09, and 0.17 hours for Problems 1, 2, and 3, respectively. While our method offers advantages in speed, space utilisation, and ease of implementation, it does not guarantee global optimality, as performance depends on parameter settings (e.g., pheromone persistence, temperature, cooling rate). In future work, complex real-world constraints such as container size (according to international container specifications), irregular shape of items, and item rotation will be integrated into the model. Moreover, we will consider incorporating learning-based components, such as deep reinforcement learning and Grey Wolf Optimiser, to implement adaptive packing mechanisms that learn and improve over time based on feedback. It may allow the model to adaptively modify its decision policy to different packing scenarios, making it more universal and usable in industries. In addition, we will consider a dataset based on ISO-standard container sizes, allowing more powerful analysis across logistics systems. Further, a comparison with other methods from recent literature indicates the high performance of the heuristic in terms of both computational efficiency and volume utilisation.

5.1 Limitations and Wider Applicability

In this topic, we will explain limitations and wider applicability of our proposed algorithm. In spite of the good results of ACO-CPP model in term of achieving the highest space utilisation and the shortest processing times, there are a number of limitations of concern as follows:

(1) The model has not been assessed in scenarios where environmental factors vary because the model uses static parameters without adaptive tuning. This may limit its applicability across different problem scales.
(2) Our model does not handle temperature-sensitive items, fragile goods, or balanced loading. This point indicates that the current model assumes rectangular boxes and fixed item orientations, which may fail to account for the full complexity of real-world cargo. These issues can affect packing feasibility and safety in actual logistics operations.

However, the proposed approach holds good chances to be adopted on a wider applicability because it can be extended to real-time logistics systems and optimisation problems. Future work will be spent on its various operational environments adaptation and limitation overcoming.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research project is supported by Science Research and Innovation Fund. Agreement No. FF67/P1-019. The study has also been technically supported by Rajamangala University of Technology Isan Surin Campus, Public University “Kadri Zeka”, and Rajabhat Maha Sarakham University. The authors are grateful to the responsible editor and the anonymous reviewers for their valuable comments and suggestions, which have greatly improved this paper.

AI Disclosure

During the preparation of this work the author(s) used generative AI in order to improve the language of the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Almasarwah, N., Abdelall, E., Suer, G., Egilmez, G., Singh, M., & Ramadan, S. (2023). Pallet loading optimization considering storage time and relative humidity. *Journal of Industrial Engineering and Management*, *16*(2), 453-471. https://doi.org/10.3926/jiem.4613.

Barykin, S.Y., Bochkarev, A.A., Kalinina, O.V., & Yadykin, V.K. (2020). Concept for a supply chain digital twin. *International Journal of Mathematical, Engineering and Management Sciences*, *5*(6), 1498-1515. https://doi.org/10.33889/ijmems.2020.5.6.111.

Chien, C.F., Lan, Y.B., Sethanan, K., & Peng, C.C. (2024). Digital system for dynamic container loading with neural network-based memory exploiting hybrid genetic algorithm for carbon reduction. *Computers & Industrial Engineering*, *191*, 110149. https://doi.org/10.1016/j.cie.2024.110149.

De Souza, E.D., Kerber, J.C., Bouzon, M., & Rodriguez, C.M.T. (2022). Performance evaluation of green logistics: paving the way towards circular economy. *Cleaner Logistics and Supply Chain*, *3*, 100019. https://doi.org/10.1016/j.clscn.2021.100019.

Delahaye, D., Chaimatanan, S., & Mongeau, M. (2019). Simulated annealing: from basics to applications. In: Gendreau, M., Potvin, J.Y. (eds) *Handbook of Metaheuristics*. Springer International Publishing, pp. 1-35. https://doi.org/10.1007/978-3-319-91086-4_1.

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28-39. https://doi.org/10.1109/MCI.2006.329691.

Doukas, H., Spiliotis, E., Jafari, M.A., Giarola, S., & Nikas, A. (2021). Low-cost emissions cuts in container shipping: Thinking inside the box. *Transportation Research Part D: Transport and Environment*, *94*, 102815. https://doi.org/10.1016/j.trd.2021.102815.

Fang, J., Chen, X., Rao, Y., Peng, Y., & Yan, K. (2024). A new approach for bin packing problem using knowledge reuse and improved heuristic. *Scientific Reports*, *14*(1), 31708. https://doi.org/10.1038/s41598-024-81749-5.

Ferrari, E., Christidis, P., & Bolsi, P. (2023). The impact of rising maritime transport costs on international trade: estimation using a multi-region general equilibrium model. *Transportation Research Interdisciplinary Perspectives*, *22*, 100985. https://doi.org/10.1016/j.trip.2023.100985.

Henderson, D., Jacobson, S.H., & Johnson, A.W. (2003). The theory and practice of simulated annealing. In: Glover, F., Kochenberger, G.A. (eds) *Handbook of Metaheuristics.* Springer, Bostan, US, pp. 287-319. https://doi.org/10.1007/0-306-48056-5_10.

Houssein, E.H., Gafar, M.H.A., Fawzy, N., & Sayed, A.Y. (2025). Recent metaheuristic algorithms for solving some civil engineering optimization problems. *Scientific Reports*, *15*(1), 7929. https://doi.org/10.1038/s41598-025-90000-8.

Jugović, A. (2020). The economic impact of container-loading problem. *Transactions on Maritime Science*, *9*(2), 271-278. https://doi.org/10.7225/toms.v09.n02.010.

Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671-680. https://doi.org/10.1126/science.220.4598.671.

Levine, J., & Ducatelle, F. (2004). Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, *55*(7), 705-716. https://doi.org/10.1057/palgrave.jors.2601771.

Liu, Y., & Jiang, X. (2024). Dynamic feedback algorithm based on spatial corner fitness for solving the three-dimensional multiple bin-size bin packing problem. *Complex & Intelligent Systems*, *10*(1), 4055-4081. https://doi.org/10.1007/s40747-024-01368-5.

McKendall, A., & Dhungel, Y. (2024). A simulated annealing algorithm for the generalized quadratic assignment problem. *Algorithms*, *17*(12), 540. https://doi.org/10.3390/a17120540.

Nguyen, T.H., & Nguyen, X.T. (2023). Space splitting and merging technique for online 3D bin packing. *Mathematics*, *11*(8), 1912. https://doi.org/10.3390/math11081912.

Peng, W., Tong, R., Tang, M., & Dong, J. (2005). Ant colony search algorithms for optimal packing problem. In: Wang, L., Chen, K., Ong, Y.S. (eds) *Advances in Natural Computation*. Springer, Berlin, Heidelberg, pp. 1229-1238. https://doi.org/10.1007/11539117_164.

Penpark, S., Thawatchai, B., & Sanchai, Y. (2021). The management of food boxes packing problems in rectangular containers using the heuristic method. *Journal of Sustainability Science and Management*, *16*(2), 129-137. http://doi.org/10.46754/jssm.2021.02.014.

Pietri, N.O., Chou, X., Loske, D., Klumpp, M., & Montemanni, R. (2021). The buy-online-pick-up-in-store retailing model: optimization strategies for in-store picking and packing. *Algorithms*, *14*(12), 350. https://doi.org/10.3390/a14120350.

Raj, R., Singh, A., Kumar, V., De, T., & Singh, S. (2024). Assessing the e-commerce last-mile logistics' hidden risk hurdles. *Cleaner Logistics and Supply Chain*, *10*, 100131. https://doi.org/10.1016/j.clscn.2023.100131.

Romero, V.S., Osaba, E., Villar-Rodriguez, E., Oregi, I., & Ban, Y. (2023). Hybrid approach for solving real-world bin packing problem instances using quantum annealers. *Scientific Reports*, *13*(1), 11777.

Sahab, M.G., Toropov, V.V., & Gandomi, A.H. (2013). A review on traditional and modern structural optimization: problems and techniques. *Metaheuristic Applications in Structures and Infrastructures*, 25-47. https://doi.org/10.1016/b978-0-12-398364-0.00002-4.

Stützle, T., López-Ibánez, M., Pellegrini, P., Maur, M., Montes de Oca, M., Birattari, M., & Dorigo, M. (2012). Parameter adaptation in ant colony optimization. In: Hamadi, Y., Monfroy, E., Saubion, F. (eds) *Autonomous Search.* Springer, Berlin, Heidelberg, pp. 191-215. https://doi.org/10.1007/978-3-642-21434-9_8.

Statista Research Department. (2025). *Retail e-commerce sales worldwide from 2017 to 2030. Statista.* https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/.

Tjandra, S., Kraus, S., Ishmam, S., Grube, T., Linßen, J., May, J., & Stolten, D. (2024). Model-based analysis of future global transport demand. *Transportation Research Interdisciplinary Perspectives*, *23*, 101016.

Tomar, V., Bansal, M., & Singh, P. (2023). Metaheuristic algorithms for optimization: a brief review. *Engineering Proceedings*, *59*(1), 238. https://doi.org/10.3390/engproc2023059238.

Wong, C.C., Tsai, T.T., & Ou, C.K. (2024). Integrating heuristic methods with deep reinforcement learning for online 3D bin-packing optimization. sensors, *24*(16), 5370. https://doi.org/10.3390/s24165370.

Yang, Y., Wu, Z., Hao, X., Liu, H., & Qi, M. (2024). Two-layer heuristic for the three-dimensional bin design and packing problem. *Engineering Optimization*, *56*(10), 1601-1638.

Zhang, Y., Yao, Y., Kan, H.K., & Luo, W. (2024). A GAN-based genetic algorithm for 3D bin packing problem. *Scientific Reports*, *14*(1), 7775. https://doi.org/10.1038/s41598-024-56699-7.

Zhao, X., Bennell, J.A., Bektaş, T., & Dowsland, K. (2016). A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, *23*(1-2), 287-320. https://doi.org/10.1111/itor.12094.

Zuo, Q., Liu, X., & Chan, W.K.V. (2022). A constructive heuristic algorithm for 3D bin packing of irregular shaped items. In: Qiu, R., Chan, W.K.V., Chen, W., Badr, Y., Zhang, C. (eds) *City, Society, and Digital Transformation.* Springer International Publishing, pp. 393-406. https://doi.org/10.1007/978-3-031-15644-1_29.