Journal of Applied Engineering and Technological Science
Vol 6(2) 2025: 1019-1039

A HEURISTIC ENHANCING ARTIFICIAL IMMUNE SYSTEM FOR THREE-DIMENSIONAL LOADING CAPACITATED VEHICLE ROUTING PROBLEM

Peeraya Thapatsuwan[1], Warattapop Thapatsuwan[2*], Chaichana Kulworatit[3]
Department of Computational Science and Digital Technology, Faculty of Liberal Arts and Science, Kasetsart University Kamphaeng Saen Campus, Thailand[1,2]
Department of Computer Science, School of Science, King Mongkut's Institute of Technology Ladkrabang, Thailand[3]
warattapop.t@ku.th[2]

Received: 12 October 2024, Revised: 11 April 2025, Accepted: 13 April 2025
**Corresponding Author*

ABSTRACT
This study addresses the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP), a highly complex NP-hard problem that combines vehicle routing with spatially constrained three-dimensional bin packing. To tackle this challenge, we propose an enhanced Artificial Immune System (En-AIS) that integrates a novel local search heuristic called "Bring-i-to-j," designed to improve routing feasibility and loading efficiency. The En-AIS algorithm is further refined through rigorous parameter tuning using a full factorial design and ANOVA analysis. Comparative experiments were conducted against conventional AIS and the Firefly Algorithm (FA) across 27 benchmark instances. Results demonstrate that En-AIS consistently outperforms both baseline methods in terms of solution quality, achieving an average improvement of 15–20% while maintaining competitive computational times. These findings highlight the algorithm's robustness and its practical potential for application in logistics and supply chain optimization tasks involving joint routing and loading decisions.

Keywords: *Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP), Artificial Immune System, Firefly Algorithm, Metaheuristics, Bring-i-to-j Heuristic, Optimization Algorithms.*

1. Introduction

Logistics plays a crucial role in the global economy and society by facilitating the efficient movement of goods and people. Transportation costs significantly influence the price of consumables and commodities, thereby affecting overall economic efficiency. In 2021, the global logistics market was valued at over €8.4 trillion and is projected to exceed €13 trillion by 2027, accounting for approximately 10.7% of the global GDP (Placek, 2023). The escalating complexity of supply and distribution networks has amplified the importance of optimizing packing and routing processes, which are pivotal in determining shipping costs. Addressing these factors can significantly reduce logistical expenses and enhance overall supply chain performance.

Enhancing on-time delivery not only fosters improved customer service but also provides organizations with a substantial competitive advantage. In this context, the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP) has gained increasing attention due to its practical significance in various industries, including e-commerce, automotive, and express delivery. The 3L-CVRP is an NP-hard problem that combines the challenges of serving multiple customers with a fleet of vehicles and efficiently loading rectangular boxes into containers, all while minimizing the total travel distance. The objective is to achieve the lowest overall transportation cost by identifying the shortest route for each vehicle and optimizing the loading sequence.

The 3L-CVRP is not only theoretically complex but also practically impactful. It represents a class of logistics problems where the efficiency of delivery operations depends on the joint optimization of routing and space-constrained loading. Effectively solving the 3L-CVRP can directly lead to reduced transportation costs, improved vehicle utilization, and enhanced customer satisfaction—particularly in industries facing tight delivery schedules and spatial constraints, such as e-commerce, grocery distribution, and parcel logistics. Due to the problem's NP-hard nature and real-world relevance, it continues to attract significant attention in combinatorial optimization and logistics research.

The complexity of 3L-CVRP arises from simultaneously optimizing routing decisions and packing strategies within vehicle capacity and stability constraints (Fuellerer et al., 2010; Gendreau et al., 2006). Exact optimization methods become impractical for larger problem instances, prompting extensive research into metaheuristic algorithms as viable alternatives.

Nature-inspired metaheuristics have demonstrated robust performance in solving complex combinatorial problems, including vehicle routing problems (VRP). These algorithms can be categorized into physically-based methods (e.g., Simulated Annealing), socially-based methods (e.g., Tabu Search), and biologically-inspired methods (e.g., Genetic Algorithms, Ant Colony Optimization, Firefly Algorithm, and Artificial Immune System) (Engin & Doyen, 2004b; Yang, 2009). While substantial research has employed metaheuristics such as Genetic Algorithms (Chen et al., 2023), Tabu Search (Meliani et al., 2022), and Adaptive Large Neighborhood Search (Qi et al., 2023; Wang et al., 2025), relatively few studies have effectively applied Artificial Immune Systems (AIS) or Firefly Algorithms (FA) specifically to the 3L-CVRP, indicating a clear research gap in the current literature (Chi et al., 2025; Leloup et al., 2025).

Existing AIS implementations typically employ generic mutation and selection processes, often limiting their performance in solving highly constrained and complex loading and routing problems (Freitas & Timmis, 2003; Garrett, 2005; Hart & Timmis, 2008). Conventional AIS explores solution spaces via simplistic mutation mechanisms without targeted local optimization, frequently resulting in suboptimal solutions and slower convergence (Timmis, 2007). Recent literature indicates that targeted heuristic improvements and hybridization within AIS can substantially enhance its capability to handle real-world VRP complexities and constraints (Chi et al., 2025; Leloup et al., 2025; Wang et al., 2023).

This paper proposes a novel heuristic-enhanced AIS algorithm designed explicitly for the 3L-CVRP to bridge this research gap. Our approach differentiates itself from traditional AIS methods by integrating a specialized local search heuristic termed "Bring-i-to-j." This heuristic systematically exploits local improvements by efficiently relocating items within and across vehicle loading sequences, enhancing route feasibility, load stability, and space utilization. Additionally, our work rigorously fine-tunes algorithm parameters through factorial experimental design, systematically optimizing performance—an essential step frequently overlooked in previous AIS studies (Aytug et al., 2003).

Thus, this research significantly contributes by addressing the explicitly identified gap through advanced heuristic integration and rigorous parameter optimization, providing substantial performance improvements over existing approaches. The remaining sections of this paper are organized as follows. Section 2 reviews related literature, describes mathematical modeling and presents the details of the proposed heuristic-enhanced AIS. Section 3 presents experimental results and analysis, while Section 4 concludes the study and outlines directions for future research.

2. Literature Review

2.1 Three-Dimensional Loading Capacitated Vehicle Routing Problem

CVRP is known to be NP-hard, extending the scope of the classical Traveling Salesman Problem (TSP) by incorporating elements of the Bin Packing Problem (BPP) (Toth & Vigo, 2002). This extension involves finding the shortest route for visiting all customers and efficiently packing goods into vehicles, making it a more complex and realistic representation of logistics problems.

The 3L-CVRP combines the Three-Dimensional Loading Problem (3DLP) and the Capacitated Vehicle Routing Problem (CVRP). It involves determining a set of vehicle routes and a containment plan for efficiently loading the cargo on each route. Each route starts and ends at the warehouse, aiming to achieve the shortest total distance between shipments or the minimum traveling cost. While these objectives are often aligned, they can differ in certain scenarios where, for example, toll roads or fuel costs influence the cost efficiency differently from distance.

The rectangular cargo will be loaded into the truck's loading area for delivery to the customer. The loading area may not necessarily be square, aligning with the rectangular nature of the cargo. Packaged goods must not exceed the volume and weight that the vehicle can handle.

In the previous literature, 3L-CVRP considers the following assumptions (Bortfeldt, 2012; Doerner et al., 2007; Fuellerer et al., 2010; Gendreau et al., 2006):

1. On one route, only one vehicle will be used (one car per route).
2. Every vehicle will have a container volume, and the payload will be the same.
3. Boxes or products are rectangular with different sizes.
4. Products of the same customer must be in the same car or path.
5. The total weight of the box in each car must not exceed the weight the car can carry.
6. The total volume of all product boxes in each car must not exceed the volume the car can load.
7. The edge of the product box must be placed parallel to the edge of the container or parallel to the x, y, and z axes without any part of the box coming out of the container.
8. The top box supports the bottom and stabilizes it by filling the remaining space in the container with foam rubber.
9. Boxes that must be delivered to the last customer should not be placed so that it will prevent boxes from reaching the earlier customers by using a first-in, first-out rule.

A mathematical model is based on (Küçük & Topaloglu Yildiz, 2022), and a mathematical model for minimizing the efficiency of volume usage for vehicle containers has been developed by previous research (Chen et al., 1995; Christensen & Rousøe, 2009).

The 3L-CVRP is made up of two parts: the transport and the loading or packaging. Part of product transportation can be explained by the complete graph $G = (V, E)$, which consists of a set of vertices and edges. The problem can be described as follows: 1) Define the set of $N+1$ transmission points ($V = 0, 1, ..., N$), and let the single depot be at point 0 and the customer at point 1 to N. 2) Determine E as the set of edges (i, j) connecting all vertex pairs. 3) Let D_{ij} represent the distance between nodes i to j. 4) Set K to represent the number of homogeneous vehicles with the same weight and volume limitation. 5) Assign the volume of packing area as $Vol = L_c * W_c * H_c$, where L_c, W_c, and H_c are the respective length, width, and height of c^{th} vehicle containers. This problem aims to achieve the shortest total distance for the cargo.

In the loading or packaging section is to minimize the number of vehicle containers required to pack the product. The problem model is described as follows: 1) Define a set of rectangular-shaped cargo boxes, the total number of B boxes ($B = \sum_{i=1}^{N} B_i$). Each box has a length, width, and height in l_b, w_b, and h_b, respectively ($b = 1, 2,..., B$). 2) Set unlimited vehicle containers with length, width, and height in L_c, W_c, and H_c, respectively ($c = 1, 2,..., K$). The length, width, and height of the box must not exceed the length, width, and height of the vehicle containers ($l_b \leq L_c$, $w_b \leq W_c$, and $h_b \leq H_c$). In this research, the product box cannot be rotated for easy loading and unloading of boxes. Therefore, the last box delivered to the customer should not be blocked using the last-in, first-out (LIFO) rule. In terms of loading or packaging, assign i to represent each customer, where $i = 1, 2, ..., N$. Each customer has a demand for b boxes. Let B_i represent the total number of boxes of customer i^{th} ($b = 1, 2, ..., B_i$). Define I_{ib} instead of the item of customer i^{th}, the box b^{th} ($I_{ib} \in \{0,1\}$), which consists of l_{ib}, w_{ib}, and h_{ib} representing the length, width, and height of the box b^{th} of customer i^{th}. The total volume of each customer's product is $vo_i = \sum_{b=1}^{B_i} l_{ib} w_{ib} h_{ib}$. The formulation of the 3L-CVRP is as follows:

Indices:

i, j	Customers i^{th} and j^{th}	$(i, j = 1, 2,..., N)$
c	Vehicle containers c^{th} for each vehicle $(c = 1, 2,..., K)$	
b, a	Boxes b^{th} and a^{th}	$(b, a = 1, 2,..., B_i)$

Parameters:

N	Total number of customer service nodes ($0, 1, 2, ..., N$), 0 is the depot
K	Total number of vehicles

Vol Maximum vehicle container volume
vo_i Total volume of each customer's product
D_{ij} Distance from node i to j
F_c Fixed costs associated with using the c^{th} vehicle
B_i Total number of boxes of customer i^{th} ($b = 1, 2, ..., B_i$)
B Total number of boxes ($B = \sum_{i=1}^{N} B_i$)
l_b, w_b, h_b The length, width, and height of box b^{th}
l_{ib}, w_{ib}, h_{ib} The length, width, and height of box b^{th} of customer i^{th}
L_c, W_c, H_c The length, width, and height of vehicle containers c^{th}
x_b, y_b, z_b The coordinates of a back-left-bottom corner that specifies the placement of boxes b^{th} (x_b, y_b, $z_b \geq 0$)
M An arbitrarily large number used in Big-M constraints.

Decision variables (a binary variable):
X_{ij}^{c} Equals 1 if the vehicle containers c^{th} travels from node i^{th} to j^{th} ; 0 otherwise.
V_c Equals 1 if the vehicle containers c^{th} is selected; 0 otherwise.
l_b^X, l_b^Y, l_b^Z Equals 1 if the length of the box b^{th} is parallel to the X, Y, or Z axes; 0 otherwise.
w_b^X, w_b^Y, w_b^Z Equals 1 if the width of the box b^{th} is parallel to the X, Y, or Z axes; 0 otherwise.
h_b^X, h_b^Y, h_b^Z Equals 1 if the height of the box b^{th} is parallel to the X, Y, or Z axes; 0 otherwise.
le_{ba} Equals 1 if the box b^{th} is placed on the left side of the box a^{th}; 0 otherwise.
ri_{ba} Equals 1 if the box b^{th} is placed on the right side of the box a^{th}; 0 otherwise.
be_{ba} Equals 1 if the box b^{th} is placed on the back of the box a^{th}; 0 otherwise.
fr_{ba} Equals 1 if the box b^{th} is placed in front of the box a^{th}; 0.
ab_{ba} Equals 1 if the box b^{th} is placed above the box a^{th}; 0 otherwise.
un_{ba} Equals 1 if the box b^{th} is placed under the box a^{th}; 0 otherwise.
I_{ib}^{c} Equals 1 if the item of customer i^{th}, box b^{th} is selected to be placed in the vehicle containers c^{th}; 0 otherwise.

Objective function:

$$\left(Minimize\ \sum_{c=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} D_{ij}X_{ij}^{c}\right) + \left(Minimize \sum_{c=1}^{K} F_c V_c\right) \tag{1}$$

Subject to:

$$\sum_{c=1}^{K}\sum_{i=0}^{N} X_{ij}^{c} = 1,\ \forall j\ \in N \cup \{0\}\ and\ i \neq j \tag{2}$$

$$\sum_{c=1}^{K}\sum_{j=0}^{N} X_{ij}^{c} = 1,\ \forall i\ \in N \cup \{0\}\ and\ i \neq j \tag{3}$$

$$\sum_{j=1}^{N} X_{ij}^{c} = \sum_{j=1}^{N} X_{ji}^{c},\ \forall c \in K\ and\ i = 0 \tag{4}$$

$$\sum_{i=1}^{N} X_{ih}^{c} - \sum_{j=1}^{N} X_{hj}^{c} = 0,\ \forall c \in K\ and\ h \in N \tag{5}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} vo_i\, X_{ij}^{c} \leq Vol,\ \forall c \in K \tag{6}$$

$$X_{ij}^{c} \in \{0,1\},\ \forall c \in K\ and\ \forall i,j \in N \tag{7}$$

$$x_b + (l_b * l_b^X) + (w_b * w_b^X) + (h_b * h_b^X) \leq x_a + (1\text{-}\ be_{ba}) * M,\ \forall b, a, b<a \tag{8}$$

$$x_a + (l_a * l_a^X) + (w_a * w_a^X) + (h_a * h_a^X) \leq x_b + (1\text{-}\ fr_{ba}) * M, \forall b, a, b<a \tag{9}$$

$$y_b + (l_b * l_b^Y) + (w_b * w_b^Y) + (h_b * h_b^Y) \leq y_a + (1\text{-}\ le_{ba}) * M,\ \forall b, a, b<a \tag{10}$$

$$y_a + (l_a * l_a^Y) + (w_a * w_a^Y) + (h_a * h_a^Y) \leq y_b + (1\text{-}\ ri_{ba}) * M,\ \forall b, a, b<a \tag{11}$$

$$z_b + (l_b * l_b^Z) + (w_b * w_b^Z) + (h_b * h_b^Z) \leq z_a + (1\text{-}\ un_{ba}) * M,\ \forall b, a, b<a \tag{12}$$

$$z_a + (l_a * l_a^Z) + (w_a * w_a^Z) + (h_a * h_a^Z) \leq z_b + (1\text{-}\ ab_{ba}) * M,\ \forall b, a, b<a \tag{13}$$

$$le_{ba} + ri_{ba} + be_{ba} + fr_{ba} + ab_{ba} + un_{ba} \geq I_{ib}^{c} + I_{ia}^{c}\ \text{-}\ 1,\ \forall b, a, b<a \tag{14}$$

$$x_b + (l_b * l_b^X) + (w_b * w_b^X) + (h_b * h_b^X) \leq L_c + (1\text{-}\ I_{ib}^{c}) * M,\ \forall b,c \tag{15}$$

$$y_b + (l_b * l_b^Y) + (w_b * w_b^Y) + (h_b * h_b^Y) \leq W_c + (1\text{-}\ I_{ib}^{c}) * M,\ \forall b,c \tag{16}$$

$$z_b + (l_b * l_b^Z) + (w_b * w_b^Z) + (h_b * h_b^Z) \leq H_c + (1\text{-}\ I_{ib}^{c}) * M,\ \forall b,c \tag{17}$$

$$\sum_{c=1}^{K} I_{ib}^{c} = 1\ ,\ \forall b \in B \tag{18}$$

$$\sum_{i=1}^{N} \sum_{b=1}^{B_i} I_{ib}^{c} \leq M \times C_c, \ \forall c \tag{19}$$

$$l_b^X, \ l_b^Y, \ l_b^Z, \ w_b^X, \ w_b^Y, \ w_b^Z, \ h_b^X, \ h_b^Y, \ h_b^Z, \ le_{ba}, \ ri_{ba}, \ be_{ba}, \ fr_{ba}, \ ab_{ba}, \ un_{ba}, \ C_c, \ I_{ib}^{c} \in \{0,1\}, \ \forall b, a, b < a \tag{20}$$

The objective function in (1) is to minimize the total distance of all routes and to reduce vehicles used. Equations (2) and (3) ensure that a route in an undirected graph visits each node exactly once and returns to the depot. In (4), the vehicle must only start and end at the depot. Equation (5) ensures that vehicles entering the node must exit the node. In (6), the packaged goods must not exceed the vehicle volume and the range of decision variables as in (7). Equations (8) to (13) ensure that the boxes loaded into the containers do not overlap. Equation (14) is applied when box b^{th} and box a^{th} are placed in container c^{th}, with at least one side of box b^{th} and box a^{th} related by six variables (le_{ba}, ri_{ba}, be_{ba}, fr_{ba}, ab_{ba}, and un_{ba}). Equations (15) to (17) ensure that all the boxes fit snugly into the container without overflowing, and (18) assures that one carton will be loaded into only one vehicle container. In (19), container c^{th} will be selected when any box is assigned. Finally, (20) specified the range of the variables. The variables le_{ba}, ri_{ba}, be_{ba}, fr_{ba}, ab_{ba}, and un_{ba} are only defined for $b<a$. The container will have the origin point coordinates at the back-left-bottom corner. The length L_c of the container is placed along the X-axis, the width W_c along the Y-axis, and the height H_c along the Z-axis. The position of the box b^{th} has the coordinates (x_b, y_b, z_b) at the back-left-bottom corner.

2.2 Metaheuristics Methods

Solving real-world problems such as scheduling, packing, and routing can be effective for small-scale instances. However, as problem size increases, more sophisticated approaches are required. Nature-inspired metaheuristics have emerged as powerful tools, drawing inspiration from natural processes. These algorithms can be categorized into three groups (Engin & Doyen, 2004b): physically-based methods like Simulated Annealing (SA); socially-based methods like Tabu Search (TS); and biologically-based methods such as Genetic Algorithms (GA), Neural Networks (NN), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Shuffled Frog Leaping (SFL), Firefly Algorithm (FA), and Artificial Immune System (AIS).

These nature-inspired methods have been widely applied to the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP). For instance, Gendreau et al. (2006) developed a TS algorithm tailored for 3L-CVRP, while Doerner et al. (2007) combined ACO and TS with fast approximation algorithms to address loading constraints. Fuellerer et al. (2010)further advanced this by implementing an ACO algorithm with efficient loading heuristics for 3L-CVRP. Despite the success of TS and ACO, the literature reveals limited use of FA and AIS in this domain, presenting a research gap that this study aims to address by enhancing AIS for 3L-CVRP.

Recent advancements have shifted toward hybrid metaheuristics to tackle increasingly complex 3L-CVRP variants. Meliani et al. (2022) proposed a TS approach with extreme point-based first-fit decreasing loading, effectively handling heterogeneous fleet 3L-CVRP with intricate constraints. Similarly, Küçük and Yildiz (2022) introduced a constraint programming model hybridized with evolutionary algorithms, decomposing routing and loading phases to solve benchmark instances efficiently. Wang et al. (2023) developed a clustering-based NSGA-II approach for multi-depot VRP with time windows and 3D loading constraints, optimizing vehicle utilization and reducing costs. Chen et al. (2023) proposed a hybrid Biogeography-Based Optimization (HBBO) algorithm, co-optimizing bin size and 3D loading layouts, validated with real-world industrial cases. Qi et al. (2023) integrated Multiobjective Evolutionary Algorithms (MOEAs) with Adaptive Large Neighborhood Search (ALNS) to address transportation planning with unloading constraints and multi-frequency visits. These hybrid approaches highlight the growing complexity of 3L-CVRP and the need for robust, adaptable solutions.

In a notable study, Ferreira et al. (2024) tackled the Green Vehicle Routing Problem with Two-Dimensional Loading Constraints and Split Delivery (G2L-SDVRP) using a Variable Neighborhood Search (VNS) metaheuristic. By integrating environmental objectives like minimizing CO_2 emissions with routing and loading decisions, their custom local search and constraint programming approach improved benchmark solutions, offering insights applicable to 3L-CVRP's operational and sustainability challenges. Building on this trend, Gimenez-Palacios et al. (2023) addressed multidrop and split delivery conditions in 2D loading, incorporating axle weight and center-of-gravity constraints relevant to 3D loading extensions.

Among the latest contributions, Leloup et al. (2025) proposed a three-phase algorithm for the Three-Dimensional Loading Vehicle Routing Problem with Split Pickups and Time Windows (3L-SPVRP-TW). Combining a savings-based heuristic, route reduction strategies, and General Variable Neighborhood Search (GVNS), their approach optimizes cost and feasibility under time-dependent travel, reachability, and stability constraints. Wang et al. (2025) introduced a hybrid method for the Collaborative Multidepot Split Delivery Network Design with Three-Dimensional Loading Constraints (CMDSDN-TDLC), using K-nearest neighbor clustering and Adaptive Nondominated Sorting Genetic Algorithm III (ANSGA-III) to balance operating costs, vehicle numbers, and loading rates. Chi et al. (2025) advanced 3L-CVRP with relocation constraints (3L-PCVRP-RC) through a hybrid framework integrating Mixed Integer Linear Programming (MILP) with an improved branch-and-price algorithm and Backward Dynamic Programming (BDP), achieving significant cost savings and volume utilization. These studies collectively demonstrate the evolution of hybrid metaheuristics, providing a foundation for further innovation in 3L-CVRP.

The Artificial Immune System (AIS) offers unique advantages for such problems. Unlike conventional single-directional search methods, AIS employs a multi-directional search via a population of antibodies (candidate solutions). Its duplication process clones only high-affinity antibodies, and a double mutation mechanism provides suboptimal solutions a second chance for improvement. However, researchers like Freitas and Timmis (2003), Garrett (2005), Hart and Timmis (2008), and Timmis (2007) have noted that traditional AIS struggles to guarantee global optima in large-scale combinatorial optimization due to its stochastic nature. In the context of 3L-CVRP, its limited ability to efficiently handle complex 3D loading constraints—such as stability, stacking, and orientation—and the lack of mechanisms to manage routing-loading interdependencies may contribute to its underutilization. Preliminary studies, such as Engin and Doyen (2004a) applying AIS to hybrid flow shop scheduling, demonstrate its potential for multi-dimensional optimization problems, suggesting that with proper adaptation, AIS could address 3L-CVRP challenges. Two strategies have been proposed to overcome these limitations: parameter fine-tuning through statistical design (Aytug et al., 2003; Engin & Doyen, 2004b) and hybridization with techniques like local search or genetic algorithms (Jaradat & Langari, 2009; Sinha & Goldberg, 2003). In the context of 3L-CVRP, such hybridization remains underexplored, offering a promising avenue for this research.

The Firefly Algorithm (FA), introduced by Yang (2008), mimics the light-emitting behavior of fireflies with three key rules: unisex attraction, brightness-based attractiveness decreasing with distance, and optimization driven by light intensity (Yang, 2009). While effective for continuous optimization, FA has seen limited application in 3L-CVRP as it is designed for continuous rather than discrete problems. Specifically, determining 3D box placements and managing constraints like stability or unloading sequences are challenges FA was not inherently designed to address, likely contributing to its underuse in this domain. However, preliminary studies like Altabeeb et al. (2021; 2019) and Trachanatzi et al. (2020) have shown FA's effectiveness in solving capacitated VRP, particularly for routing with capacity constraints. These applications suggest that with modifications to handle discrete 3D loading, FA could offer valuable insights for 3L-CVRP, though such adaptations remain scarce.

This research proposes an enhanced AIS framework (En-AIS) integrating domain-specific local search, such as the "Bring-i-to-j" heuristic, and problem decomposition to address 3L-CVRP's complex routing and loading constraints. By building on the strengths of AIS and addressing its limitations, this approach aims to fill the gap in applying biologically inspired methods to 3L-CVRP, offering a robust alternative to existing metaheuristics like TS, ACO, and FA.

3. Research Methods

3.1 Natural Clonal Selection

In the body's immune system, clonal selection is the process by which immune cells, specifically B cells, that identify certain antigens are chosen to proliferate and differentiate. When a B cell connects with an antigen, it activates and begins clonal expansion, generating multiple identical cells. These clones undergo somatic hypermutation, introducing random variations to their receptors. This mutation enhances receptor diversity and improves the immune system's efficiency in identifying and countering the antigen (Burnet, 1959).

1. Antigen Recognition: Each B cell possesses distinct receptors that can attach to specific antigens. When a matching antigen is encountered, the B cell binds to it.
2. Activation and Proliferation: Upon binding to the antigen, the B cell is activated and enters a phase of rapid division, creating numerous identical clones.
3. Differentiation: The cloned B cells then specialize into plasma cells and memory B cells. Plasma cells produce and release antibodies that directly target the antigen, while memory B cells persist in the body, ready to provide a quicker response if the antigen reappears.
4. Affinity Maturation: Throughout proliferation, somatic hypermutation introduces minor mutations in the receptors of cloned B cells, enhancing receptor diversity. B cells with receptors of higher affinity for the antigen are preferentially expanded.
5. Elimination of Self-reactive Cells: B cells that strongly bind to self-antigens (normal body molecules) are generally eliminated through negative selection, helping to prevent autoimmune reactions.

3.2 Artificial Immune System for 3L-CVRP

AIS (Artificial Immune System) is a biologically inspired method that mimics the immune system's ability to recognize and adapt to foreign pathogens. It is a branch of computational intelligence used for solving complex optimization problems by simulating immune responses (Dasgupta, 2006). CLONALG is a variant of AIS based on clonal selection theory, which focuses on the adaptability of B and T cells to detect and eliminate foreign invaders (non-self antigens or foreign molecules) (Burnet, 1959).

In the context of this study, each antibody encodes a complete solution to the 3L-CVRP by representing a randomized customer visiting sequence. For example, an antibody such as C1–C2–C3 indicates that the vehicle will visit Customer 1 first, followed by Customer 2, and then Customer 3. Based on this sequence, the items associated with each customer are packed into vehicles in order.

The algorithm begins loading from Vehicle 1 by attempting to pack all items belonging to the current customer using a three-dimensional packing heuristic.

Specifically, a bottom-left-front (BLF)-inspired spatial heuristic is employed. The algorithm begins placing items at the bottom-left corner of the container and fills space horizontally to the right. Once the width is exhausted, it stacks vertically upward. When vertical space is also filled, it progresses front-to-back along the container's depth. This placement strategy continues until the entire container space is used.

During the optimization process, all solution candidates (antibodies) are evaluated based on strict feasibility rules. The 3L-CVRP incorporates two main categories of constraints: vehicle capacity and loading feasibility.

Vehicle capacity constraints are enforced by ensuring that the total volume of items assigned to each vehicle does not exceed the vehicle's container capacity. If an attempted packing exceeds the available space, the algorithm assigns the entire customer's shipment to the next vehicle.

Loading feasibility—such as placement orientation, stackability, and collision avoidance—is inherently handled within the packing heuristic. The 3D placement routine enforces all physical constraints during the packing process, ensuring that items are placed only when they fit without overlap, orientation violations, or unstable stacking. As a result, all constructed solutions are feasible by design, and no additional penalty or rejection mechanism is required.

To preserve delivery consistency and operational efficiency, the system is designed to pack all items from the same customer within a single vehicle whenever possible. If the current vehicle does not have enough capacity to accommodate the entire set of items, none of them will be partially loaded. Instead, the algorithm moves all that customer's items to the next available vehicle.

This grouping constraint not only ensures spatial feasibility but also enhances logistical practicality, aligning the solution with real-world delivery requirements such as last-mile logistics and simplified unloading processes.

This encoding strategy enables the algorithm to jointly optimize both routing and loading in a unified solution representation. Fig. 1 illustrates this encoding approach, demonstrating how customer-to-item relationships are translated into vehicle-level 3D packing layouts.

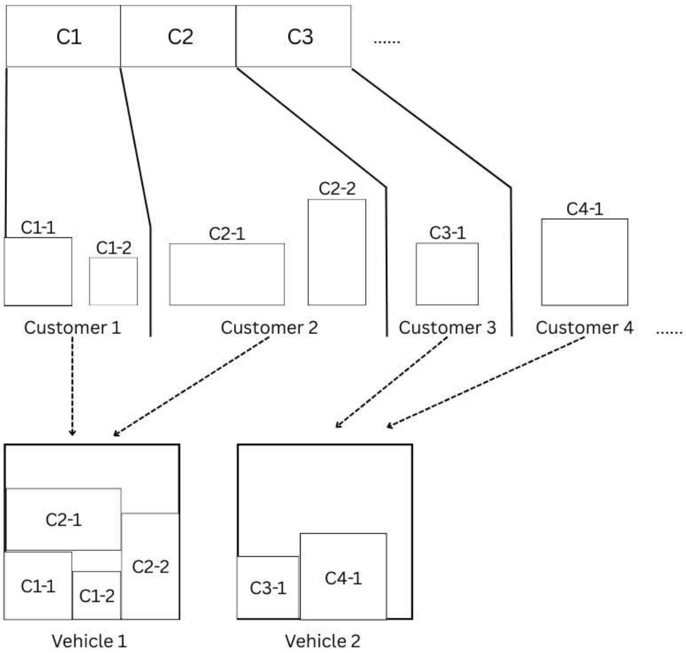


Fig. 1. Antibody encoding in the AIS framework for 3L-CVRP, illustrating customer-to-item mapping and 3D packing layout.

The CLONALG algorithm can be described as follows:

1. Randomly initialize the antibody population.
2. For each antibody (*Ab*), calculate its affinity value (*Aff_Val*) from (21). The fitness value of *Ab* can be calculated as the main objective function (1).

$$Aff\_Val = 1 \text{ / Fitness value of } Ab \tag{21}$$

3. Rank the population in descending order of *Aff_Val*, then calculate a large group of clones according to the clonal factor (*β*) and the maximum population from (22).

$$Num\_Clones = \left(\frac{\beta * Max\ population}{Rank\_index}\right) \tag{22}$$

4. Generate clones of these antibodies based on their affinity; antibodies with higher affinity will produce more clones.
5. Differentiate these clones by implementing mutation on each clone (Engin & Doyen, 2004b).
6. Add these mutated clones to the *Ab* population, sort them by their affinity, and select the number of *Ab* in the top rank of the population for the next generation.
7. Repeat steps 2 to 6 until a termination criterion is satisfied.

In this study, key parameters of the AIS—namely, population size (Ab), number of generations (G), clonal factor (β), and elimination rate (%E)—were not arbitrarily assigned. Instead, a parameter tuning process was conducted as part of the experimental framework, detailed in Section 4.1. A full factorial design was employed to evaluate different parameter levels systematically. The selected configurations were chosen based on their statistical significance using Analysis of Variance (ANOVA) across 27 problem instances.

Although the mutation rate was not explicitly isolated as a standalone parameter, its influence is inherently captured through the clonal selection and variation mechanisms. The mutation strength is implicitly controlled by the number of clones (influenced by β) and the proportion of eliminated antibodies (%E), which determine the extent of diversity introduced per iteration. These settings enable a balance between exploration and exploitation, thereby improving both convergence speed and solution quality.

Furthermore, the parameter tuning process can be considered a form of sensitivity analysis, as it involves testing multiple combinations of parameters and observing their impact on performance across various problem sizes. The results from the factorial experiments revealed how different levels of each parameter (AbG, β, and %E) influenced solution quality and convergence behavior. These insights helped identify parameter configurations that consistently performed well, thereby contributing to the robustness of the proposed algorithm.

This study incorporates the “Bring-i-to-j” local search heuristic into the core optimization process to improve the baseline AIS. After the mutation step in each generation, a subset of elite antibodies is refined by repositioning one customer within its route. This local adjustment helps reduce unnecessary detours, shortens the total travel distance, and improves vehicle utilization. The algorithm intensifies its search around high-potential solutions by embedding this heuristic within the evolutionary loop—before the affinity evaluation and selection steps. As a result, it converges more quickly and consistently produces higher-quality solutions. This enhancement transforms the AIS into a more efficient and problem-specific solver for the 3L-CVRP.

3.3 Local Search: *Bring-i-to-j*

Bring-i-to-j is a proposed heuristic that is based on the idea of mutation. In this heuristic, a random customer *i* in a solution is randomly selected, and then a random another customer *j* is selected. Finally, move customer *i* to the adjacent location next to customer *j*. The concept of *Bring-i-to-j* is to move only one customer while another customer stands for waiting. This approach differs from traditional mutation operators that randomly swap two customers' positions, potentially causing disruptions to the vehicle routing plan. By moving only one customer, the *Bring-i-to-j* heuristic reduces the likelihood of disrupting the vehicle routing plan and improves the algorithm's efficiency.

Bring-i-to-j will be applied to the antibody population after merging with the mutated clones. Fig. 2 describes the idea of this approach to help antibodies find solutions with only one move. For example, suppose a route where the vehicle will leave depot *0* to deliver goods to customers 1, 3, 4, and 2 at different locations and then return to the depot. The route improvement will initially randomize two customers as *i* and *j*, respectively, and customer *i* will be moved adjacency to customer *j*.

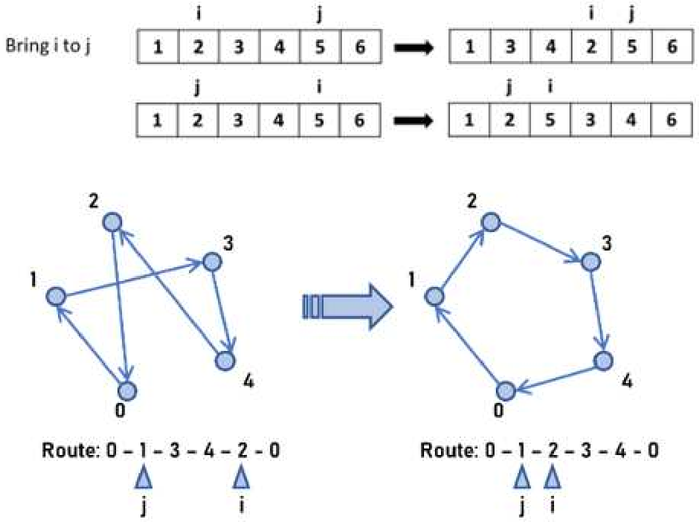
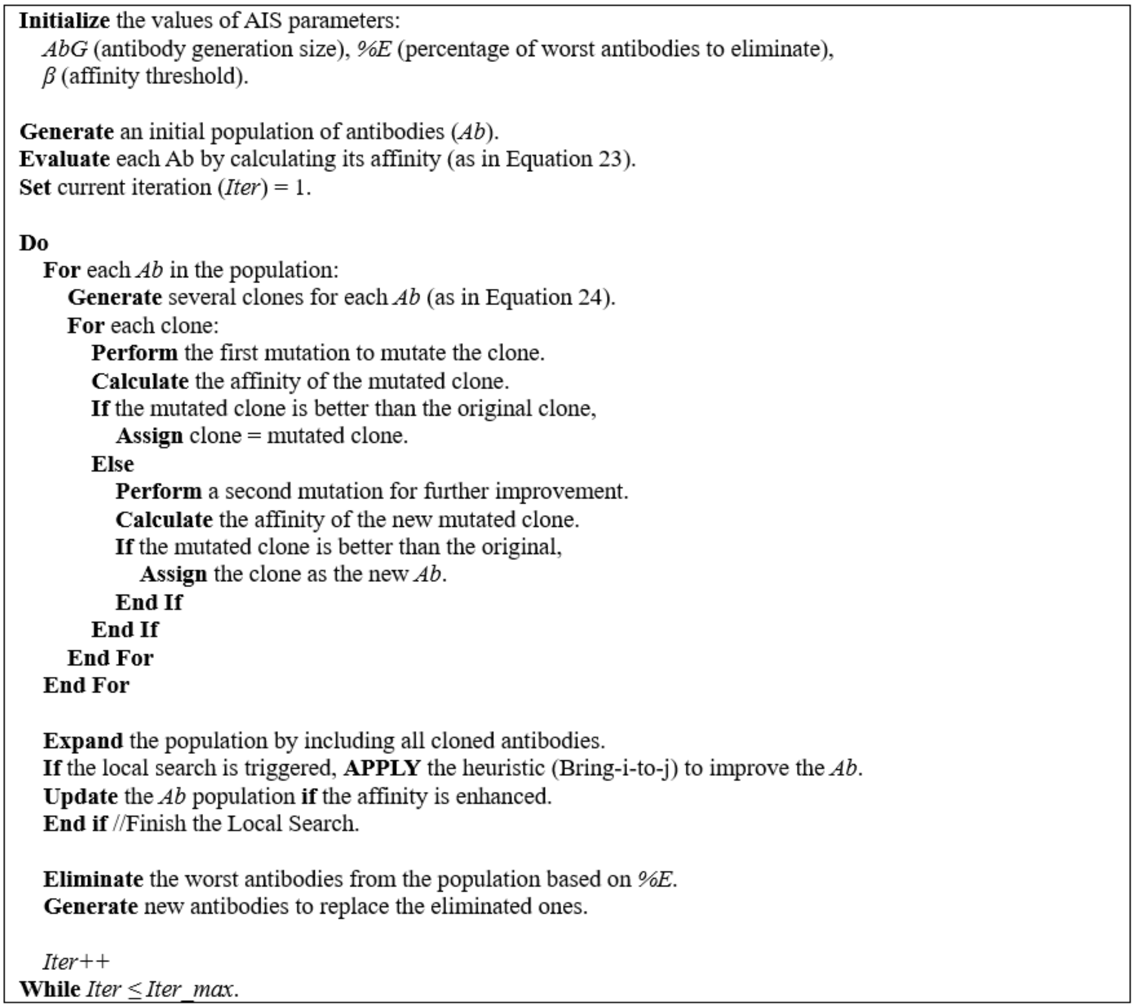


Fig. 2. Bring-i-to-j heuristic illustration: moving customer i closer to j to improve solution feasibility.

Compared to the proposed En-AIS, the Firefly Algorithm (FA) follows a simpler movement-based mechanism with fewer problem-specific enhancements. While FA is generally effective for continuous optimization problems, its adaptation to combinatorial scenarios like the 3L-CVRP lacks embedded local search strategies. This absence limits its capability to refine solutions iteratively, especially in high-dimensional or constraint-heavy instances. As summarized in Table 3, En-AIS consistently outperforms FA in both average and best-case solution quality across 27 benchmark instances while maintaining a comparable computation time. These results highlight the importance of integrating domain-specific heuristics, such as Bring-i-to-j, to guide metaheuristic search processes more effectively.

3.4 AIS enhancing with LS for 3L-CVRP

This paper uses the *Bring-i-to-j* as the local search for the original AIS. The pseudo-code of AIS enhanced with local search (En-AIS) is displayed in Fig. 3.



```
Initialize the values of AIS parameters:
  AbG (antibody generation size), %E (percentage of worst antibodies to eliminate),
  β (affinity threshold).

Generate an initial population of antibodies (Ab).
Evaluate each Ab by calculating its affinity (as in Equation 23).
Set current iteration (Iter) = 1.

Do
  For each Ab in the population:
    Generate several clones for each Ab (as in Equation 24).
    For each clone:
      Perform the first mutation to mutate the clone.
      Calculate the affinity of the mutated clone.
      If the mutated clone is better than the original clone,
        Assign clone = mutated clone.
      Else
        Perform a second mutation for further improvement.
        Calculate the affinity of the new mutated clone.
        If the mutated clone is better than the original,
          Assign the clone as the new Ab.
        End If
      End If
    End For
  End For

  Expand the population by including all cloned antibodies.
  If the local search is triggered, APPLY the heuristic (Bring-i-to-j) to improve the Ab.
  Update the Ab population if the affinity is enhanced.
  End if //Finish the Local Search.

  Eliminate the worst antibodies from the population based on %E.
  Generate new antibodies to replace the eliminated ones.

  Iter++
While Iter ≤ Iter_max.
```

Fig. 3. Pseudo-code of En-AIS.

Compared to the traditional AIS, the proposed En-AIS explicitly integrates the Bring-i-to-j heuristic into its evolutionary process. Specifically, En-AIS applies this local search heuristic after the mutation step in each generation but before evaluating the affinity and selecting the best antibodies. This refinement step significantly improves the solution quality by reducing unnecessary travel distances and enhancing loading feasibility, enabling the algorithm to converge faster and achieve superior solutions, especially in complex or larger problem instances.

3.5 Firefly Algorithm for 3L-CVRP

The main steps of FA begin with determining the intensity of the flashing light for each firefly. In the Firefly Algorithm, light intensity is used as a metaphor for the quality of the solution—fireflies with higher light intensity represent better solutions, and other fireflies are attracted to them to explore the search space effectively.

During the dual light intensity comparison loop, each firefly compares its light intensity with others. Fireflies with lower intensity will move towards those with higher light intensity to improve their solution quality. The moving distance depends on the attractiveness. The movement of a firefly i, which is attracted by a more attractive (i.e., brighter) firefly j, is determined by the equation (23). This equation models how the position of firefly i changes in response to the attraction, ensuring that it moves towards better solutions in the optimization process.

The firefly attractiveness function calculation is displayed in (24), and the distance between any two fireflies i and j at x_i and x_j, can be defined as a Cartesian distance (r_{ij}) using (25). After migration, the new fireflies are assessed based on the quality of their solutions, such as distance minimization or cost optimization, and their light intensity is adjusted accordingly to reflect the updated solution quality.

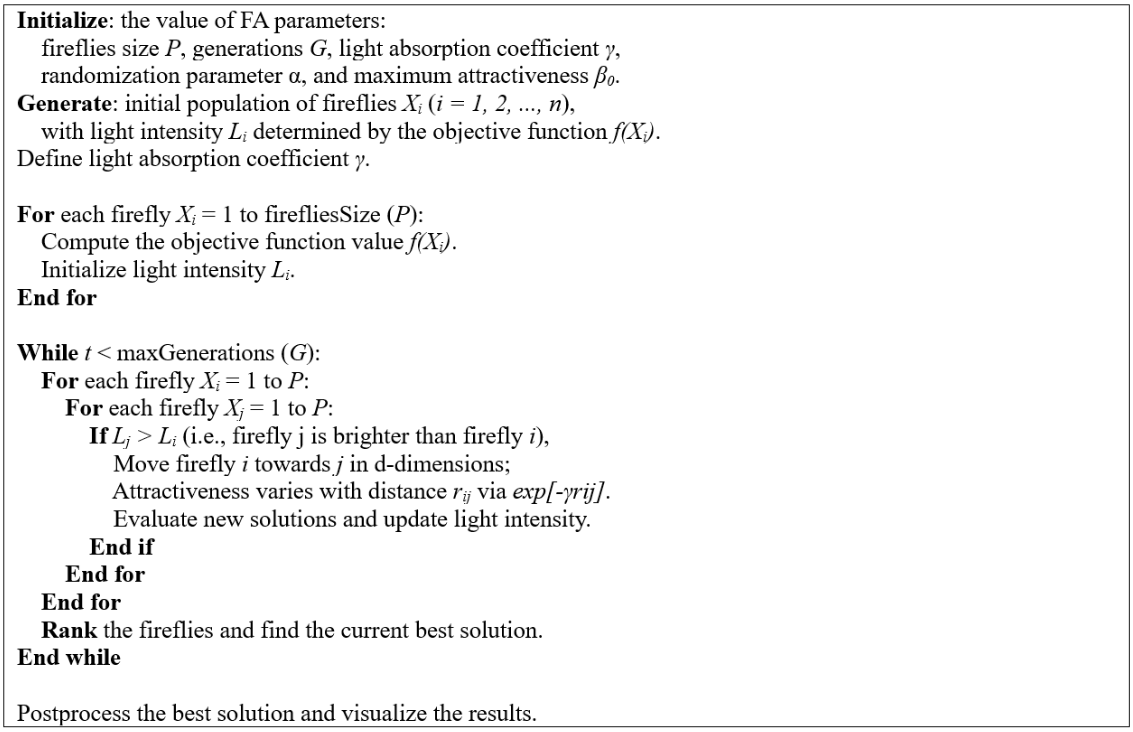
During the pairwise comparison loop, the best-so-far solution is iteratively updated. The pairwise comparison process is repeated until the termination criteria are satisfied. Finally, the best-so-far solution is visualized.

The pseudo-code of the FA applied to solve the 3L-CVRP is illustrated in Fig. 4. In summary, the key steps include initializing the population of fireflies, calculating their light intensity based on solution quality, iteratively moving fireflies towards brighter ones, updating the best-so-far solution, and repeating this process until convergence criteria are met.

$$x_i = x_i + \beta_0 \times \exp\left(-\gamma r_{ij}^2\right) \times \left(x_j - x_i\right) + \alpha\left(rand - \frac{1}{2}\right) \tag{23}$$

$$\beta_{(r)} = \beta_0 \times \exp(-\gamma r^m)\,, with\ m \geq 1 \tag{24}$$

$$r_{ij} = \left\|x_i - x_j\right\| = \sqrt{\sum_{k=1}^{d}\left(x_{i,k} - x_{j,k}\right)^2} \tag{25}$$



```
Initialize: the value of FA parameters:
  fireflies size $P$, generations $G$, light absorption coefficient $\gamma$,
  randomization parameter $\alpha$, and maximum attractiveness $\beta_0$.
Generate: initial population of fireflies $X_i$ ($i = 1, 2, ..., n$),
  with light intensity $L_i$ determined by the objective function $f(X_i)$.
Define light absorption coefficient $\gamma$.

For each firefly $X_i$ = 1 to firefliesSize ($P$):
  Compute the objective function value $f(X_i)$.
  Initialize light intensity $L_i$.
End for

While $t$ < maxGenerations ($G$):
  For each firefly $X_i$ = 1 to $P$:
    For each firefly $X_j$ = 1 to $P$:
      If $L_j > L_i$ (i.e., firefly j is brighter than firefly $i$),
        Move firefly $i$ towards $j$ in d-dimensions;
        Attractiveness varies with distance $r_{ij}$ via $exp[-\gamma rij]$.
        Evaluate new solutions and update light intensity.
      End if
    End for
  End for
  Rank the fireflies and find the current best solution.
End while

Postprocess the best solution and visualize the results.
```

Fig. 4. Pseudo-code of FA.

4. Results and Discussions

This section was designed to verify the appropriate AIS parameter settings for 3L-CVRP. It also benchmarks the performance of AIS with the proposed heuristic in terms of computational efficiency and the quality of the solutions obtained. All algorithms were coded in

C# language, chosen due to its balance of performance, ease of use, and familiarity among the research team. These algorithms were tested on a set of 27 instances introduced by (Gendreau et al., 2006). The number of customers ranged from 15 to 100, vehicles from 4 to 23, and items from 32 to 198. The computational tests were performed on a computer with an Intel Core i5-8400 CPU @2.80GHz with 8 GB of RAM, running Windows 10 as the operating system. Each run was repeated 10 times, using different random seeds for each run.

4.1 Experiment A

Due to random volatility in evolutionary algorithms such as AIS, which refers to fluctuations in solution quality and variability in convergence, the parameter settings were investigated to fit the 3L-CVRP. Table 1 displays the full factorial design of AIS parameters. It includes the combination of the number of antibodies and number of generations (*AbG*), referring to the number of solutions generated and the required processing time. Therefore, the combination of AbG was defined at 22,500 to limit the computational time required based on prior experimentation and practical limitations observed in similar optimization problems. The percentage of antibody elimination (*%E*) and clonal factor (*β*) are defined as scaling factors for the number of clones created for selected antibodies. These parameters are important as they control the population's diversity and the algorithm's convergence rate, impacting both the exploration of the search space and the refinement of candidate solutions.

Table 1 - Experimental factors and levels.

Factors	Levels	Values		
		Low (-1)	**Medium (0)**	**High (+1)**
AbG	3	90*250	150*150	250*90
%E	3	25%	50%	75%
β	3	0.5	1.0	1.5

The computational results were analyzed using analysis of variance (ANOVA), a general linear model. ANOVA was selected because it effectively determines the influence of multiple factors on outcomes and identifies statistically significant differences among tested instances. The individual P-values for all 27 instances are summarized in Table 2.

The first four columns of Table 2 include the instance name, total number of customers (*Cus*), vehicles (*Veh*), and items (*Item*). The results indicate that all factors have a statistically significant impact on small to medium-sized problems. For small problems, assigning values to AbG tends to produce a large number of antibodies, leading to efficient convergence. However, as the problem size increases, the number of antibodies decreases, requiring more generations to reach convergence.

The clonal factor setting was found to be significant across almost every problem size and becomes increasingly crucial for optimizing very large problems. A high clonal factor value generates a substantial number of clones, significantly enhancing the search capability but requiring longer computational time. Depending on the problem size, the computational time can range from several hours to multiple days.

The hypermutation rate (*β*) also proved statistically significant, specifically for larger problem sizes. As the problem complexity and solution space expand, *β* becomes essential for maintaining diversity among solutions, thereby preventing premature convergence. Without explicit local search, a high *β* value facilitates exploration of a broader solution space, allowing AIS to search deeper and find competitive solutions even without specialized heuristics.

Overall, Table 2’s findings underscore the distinct role of each parameter. For small to medium-sized problems, a high *AbG* setting accelerates convergence by producing a larger antibody population that efficiently explores smaller solution spaces. In contrast, larger instances benefit significantly from a high clonal factor, which enables deeper solution exploration, albeit with increased computational time. *β* remains a critical parameter for the largest problem sizes, ensuring effective exploration across the expanded solution landscape by preventing premature convergence and enhancing search diversity.

Table 2 also suggests the appropriate setting for each parameter (shown in parentheses). Although the p-value for these settings was insignificant, the main effect plot provides insight into the most effective parameter configurations for different problem sizes, guiding parameter selection for improved optimization outcomes.

Table 2 - P-values and parameter effects based on main effect plot.

Instances				**P-Values (Minimum Values)**		
Name	***Cus***	***Veh***	***Item***	***AbG***	***%E***	***β***
E016-03m	15	5	32	**0.001 (250*90)**	**0.006 (25%)**	0.456 **(1.5)**
E016-05m	15	5	26	**0.000 (250*90)**	**0.000 (25%)**	**0.034 (1.5)**
E021-04m	20	5	37	**0.000 (250*90)**	0.158 **(25%)**	0.070 **(1.5)**
E021-06m	20	6	36	**0.000 (250*90)**	**0.002 (25%)**	**0.033 (1.5)**
E022-04g	21	7	45	0.182 **(250*90)**	**0.022 (25%)**	0.053 **(1.5)**
E022-06m	21	6	40	0.143 **(250*90)**	**0.001 (25%)**	**0.002 (1.0)**
E023-03g	22	6	46	**0.000 (250*90)**	**0.006 (25%)**	0.153 **(1.5)**
E023-05s	22	8	43	**0.037 (250*90)**	**0.031 (25%)**	**0.045 (1.5)**
E026-08m	25	8	50	**0.002 (250*90)**	**0.000 (50%)**	**0.002 (1.5)**
E030-03g	29	10	62	**0.000 (250*90)**	0.088 **(25%)**	0.068 **(1.0)**
E030-04s	29	9	58	**0.000 (250*90)**	**0.001 (50%)**	**0.006 (1.5)**
E031-09h	30	9	63	**0.007 (250*90)**	**0.046 (25%)**	**0.007 (1.0)**
E033-03n	32	9	61	**0.000 (250*90)**	**0.002 (25%)**	**0.046 (1.0)**
E033-04g	32	11	72	**0.000 (250*90)**	0.346 **(50%)**	**0.013 (1.0)**
E033-05g	32	10	68	**0.001 (250*90)**	0.066 **(25%)**	0.643 **(1.0)**
E036-11h	35	11	63	**0.015 (250*90)**	0.350 **(25%)**	0.069 **(1.5)**
E041-14h	40	14	79	0.226 **(250*90)**	0.052 **(50%)**	**0.014 (1.5)**
E045-04f	44	14	91	0.368 **(250*90)**	0.356 **(25%)**	**0.019 (1.5)**
E051-05e	50	13	99	**0.000 (250*90)**	0.964 **(50%)**	0.363 **(1.5)**
E072-04f	71	20	147	0.454 **(150*150)**	0.369 **(50%)**	**0.000 (1.5)**
E076-07s	75	18	155	0.168 **(250*90)**	0.138 **(25%)**	**0.000 (1.5)**
E076-08s	75	19	146	0.646 **(150*150)**	0.585 **(50%)**	**0.000 (1.5)**
E076-10e	75	18	150	0.994 **(150*150)**	0.658 **(50%)**	**0.000 (1.5)**
E076-14s	75	18	143	0.083 **(150*150)**	0.791 **(25%)**	**0.001 (1.5)**
E101-08e	100	24	193	0.427 **(150*150)**	0.375 **(50%)**	**0.000 (1.5)**
E101-10c	100	28	199	0.187 **(150*150)**	0.416 **(25%)**	**0.000 (1.5)**
E101-14s	100	25	198	**0.024 (90*250)**	**0.014 (50%)**	**0.000 (1.5)**

A functional analysis was also conducted to better interpret the impact of each parameter on the algorithm's behavior. The number of antibodies and generations (*AbG*) directly controls the population size and overall search breadth. A high *AbG* value allows broader exploration but increases computational time, whereas a lower value accelerates the search but may lead to premature convergence due to limited diversity.

The elimination rate (*%E*) manages how much of the population is replaced in each iteration. A high *%E* ensures greater diversity by discarding more low-quality antibodies but may disrupt the convergence process. Conversely, a low *%E* value helps preserve promising solutions but might cause stagnation.

The clonal factor (β) governs the number of clones produced from high-affinity antibodies. A higher β increases local exploitation by allowing deeper refinement of reasonable solutions, though it adds computational cost. Lower β values reduce the computational burden but limit the fine-tuning of elite solutions.

Understanding these behavioral impacts helps fine-tune parameter settings beyond statistical significance, ensuring robust performance across a wide range of problem instances.

4.2 Experiment B

This experiment aimed to compare three algorithms: AIS, En-AIS, and FA, in terms of minimum (Min), average (Avg), and maximum (Max) solutions and computational time usage (Time). Both AIS and En-AIS used the parameter settings obtained from Experiment A, while the FA parameters were based on (Yang, 2008).

Table 3 summarizes the experimental results for all three algorithms across the 27 benchmark instances. Each row presents the minimum, maximum, and average total travel distances obtained in 10 independent runs and the average computational time. These numerical values clearly show performance differences among AIS, En-AIS, and FA.

Table 3 - Experimental results of all algorithms.

Problems	**Solutions' Quality of Algorithms**					**Problems**	**Solutions' Quality of Algorithms**				
	Alg	**Min**	**Max**	**Avg**	**Time(s)**		**Alg**	**Min**	**Max**	**Avg**	**Time(s)**
E016-03m	AIS	**344.12**	344.41	344.38	6.798	E033-05s	AIS	1661.44	1880.89	1767.90	14.06
	En-AIS	**344.12**	**344.12**	**344.12**	9.112		En-AIS	**1551.42**	**1649.02**	**1603.80**	19.47
	FA	**344.12**	346.70	344.85	7.840		FA	1753.99	1952.10	1864.72	16.43
E016-05m	AIS	**340.55**	**340.55**	**340.55**	5.750	E036-11h	AIS	725.38	806.06	767.58	13.52
	En-AIS	**340.55**	**340.55**	**340.55**	7.953		En-AIS	**709.55**	**757.20**	**735.56**	19.11
	FA	**340.55**	**340.55**	**340.55**	6.649		FA	747.87	847.03	804.18	15.61
E021-04m	AIS	397.89	**408.46**	402.40	7.727	E041-14h	AIS	957.13	1057.01	1009.28	16.20
	En-AIS	**386.52**	410.38	**396.52**	11.537		En-AIS	**916.55**	**971.83**	**955.54**	22.27
	FA	392.55	436.81	410.20	8.884		FA	1014.44	1137.08	1047.54	18.75
E021-06m	AIS	**440.68**	459.45	449.99	7.945	E045-04f	AIS	1507.82	1673.24	1599.41	22.09
	En-AIS	**440.68**	**440.94**	**440.76**	10.75		En-AIS	**1424.21**	**1506.48**	**1484.06**	26.38
	FA	**440.68**	464.90	455.65	9.06		FA	1578.52	1728.03	1662.97	25.36
E022-04g	AIS	479.82	519.83	496.71	9.33	E051-05e	AIS	1095.07	1245.54	1165.53	23.18
	En-AIS	**475.48**	**495.30**	**478.92**	12.78		En-AIS	**921.29**	**984.66**	**952.62**	29.51
	FA	479.82	576.53	512.75	10.94		FA	1124.20	1245.78	1171.36	26.83
E022-06m	AIS	509.36	531.30	518.15	8.28	E072-04f	AIS	907.34	1032.89	965.38	33.02
	En-AIS	**504.39**	**516.04**	**509.62**	11.65		En-AIS	**736.78**	**792.66**	**759.07**	42.96
	FA	514.50	543.50	521.50	9.41		FA	888.67	1006.82	954.96	38.11
E023-03g	AIS	891.07	**920.17**	902.93	9.75	E076-07s	AIS	1669.05	1843.45	1750.59	33.01

Problems	Solutions' Quality of Algorithms					Problems	Solutions' Quality of Algorithms				
	Alg	Min	Max	Avg	Time(s)		Alg	Min	Max	Avg	Time(s)
	En-AIS	**883.48**	930.88	**894.91**	13.18		En-AIS	**1343.22**	**1415.37**	**1379.26**	44.95
	FA	896.38	957.64	918.78	11.22		FA	1683.39	1848.84	1767.51	38.01
E023-05s	AIS	886.11	983.42	924.71	9.27	E076-08s	AIS	1634.09	1832.78	1712.18	31.53
	En-AIS	**882.06**	**939.09**	**897.82**	12.75		En-AIS	**1385.08**	**1432.16**	**1417.25**	43.91
	FA	892.57	988.65	933.13	10.73		FA	1761.14	1905.48	1826.67	36.37
E026-08m	AIS	687.82	760.89	720.96	10.32	E076-10e	AIS	1689.84	1824.79	1745.02	31.66
	En-AIS	**672.35**	**715.52**	**685.72**	14.15		En-AIS	**1341.75**	**1389.39**	**1364.64**	43.92
	FA	712.01	799.81	739.33	11.88		FA	1730.84	1854.86	1789.97	36.52
E030-03g	AIS	952.42	1079.63	1026.75	12.50	E076-14s	AIS	1675.04	1845.43	1761.30	33.08
	En-AIS	**918.20**	**972.76**	**953.26**	17.50		En-AIS	**1289.18**	**1394.28**	**1341.93**	42.87
	FA	942.27	1091.18	1024.61	14.62		FA	1775.90	1930.58	1831.96	37.69
E030-04s	AIS	951.96	1074.34	1009.66	12.50	E101-08e	AIS	2285.44	2631.79	2434.78	43.25
	En-AIS	980.65	**967.15**	**933.36**	16.60		En-AIS	**1745.38**	**1898.22**	**1810.02**	51.53
	FA	**948.16**	1098.22	1028.17	14.25		FA	2357.58	2573.67	2466.56	49.58
E031-09h	AIS	643.83	701.38	674.82	13.19	E101-10c	AIS	2813.29	3109.39	2951.42	47.07
	En-AIS	**639.65**	**650.06**	**647.06**	17.82		En-AIS	**2024.56**	**2141.21**	**2072.24**	58.57
	FA	660.91	738.75	687.63	15.46		FA	2750.89	3213.27	2980.37	53.51
E033-03n	AIS	2953.63	3291.73	3011.44	12.84	E101-14s	AIS	2468.47	2589.54	2530.23	45.61
	En-AIS	**2918.24**	**3010.08**	**2961.43**	18.45		En-AIS	**1876.06**	**2040.93**	**1942.61**	60.38
	FA	3014.23	3480.72	3315.50	14.81		FA	2505.91	2763.47	2633.82	51.99
E033-04g	AIS	1802.82	1902.69	1852.95	14.40						
	En-AIS	**1588.81**	**1711.49**	**1658.05**	19.77						
	FA	1809.83	2061.82	1927.22	16.55						

The experimental results in Table 3 reported that En-AIS's minimum, maximum, and average solutions were superior to those produced by other algorithms for all problem sizes, with an average improvement of 15-20% in solution quality compared to AIS and FA. This indicates that En-AIS finds high-quality solutions across various scenarios more effectively. The Bring-i-to-j heuristic significantly improved the quality of the obtained solution by enhancing local search capabilities and allowing the algorithm to escape local optima more efficiently. Specifically, the heuristic moves a selected customer to a new position in the route, creating a new neighborhood solution that helps explore different configurations and overcome local stagnation. On average, En-AIS achieved a 17.42% improvement in solution quality over AIS and a 13.97% improvement over FA across all test instances. The most significant improvement was observed in instance E033-04g, where En-AIS outperformed AIS by 10.50% and FA by 13.98%. These results validate the statistical superiority of En-AIS in both minimum and average performance metrics.

From a computational perspective, the proposed En-AIS demonstrates a favorable balance between solution quality and processing time. Although the integration of the Bring-i-to-j local search and dual mutation introduces additional overhead compared to the baseline AIS, the performance gains justify the moderate increase in computation time. On average, En-AIS required approximately 15–25% more time than standard AIS or FA (as shown in Table 3), but consistently delivered better minimum and average solution values across all instances. This efficiency suggests that En-AIS is suitable for mid-sized and large-scale problems, particularly in planning environments where solution quality is prioritized over runtime optimality.

To statistically verify the observed differences, a one-way ANOVA was conducted using the average values from all 27 benchmark instances for AIS, En-AIS, and FA. The results, summarized in Table 4, indicated that the differences among the three algorithms were not statistically significant at the 0.05 level (p = 0.519). Furthermore, a Tukey post-hoc test confirmed that the mean solution qualities of all algorithms fell into the same grouping, suggesting that the average performance differences were not statistically distinguishable. En-AIS consistently produced lower mean values than the other algorithms, indicating practical advantages in solution quality across diverse scenarios. Visual summaries of the mean comparisons and confidence intervals are provided in Fig. 5 and 6.

Table 4 - One-way ANOVA results comparing solution quality across 27 instances.

Source	DF	SS	MS	F	p-value
Algorithm	2	753902	376951	0.66	0.519
Error	78	44396230	569182		
Total	80	45150132			

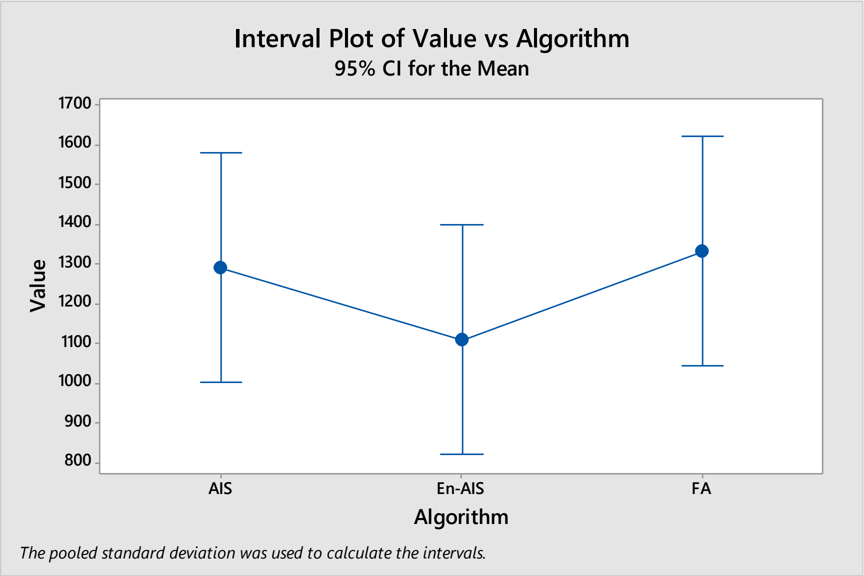


Fig. 5. Mean solution quality of AIS, En-AIS, and FA with 95% confidence intervals.

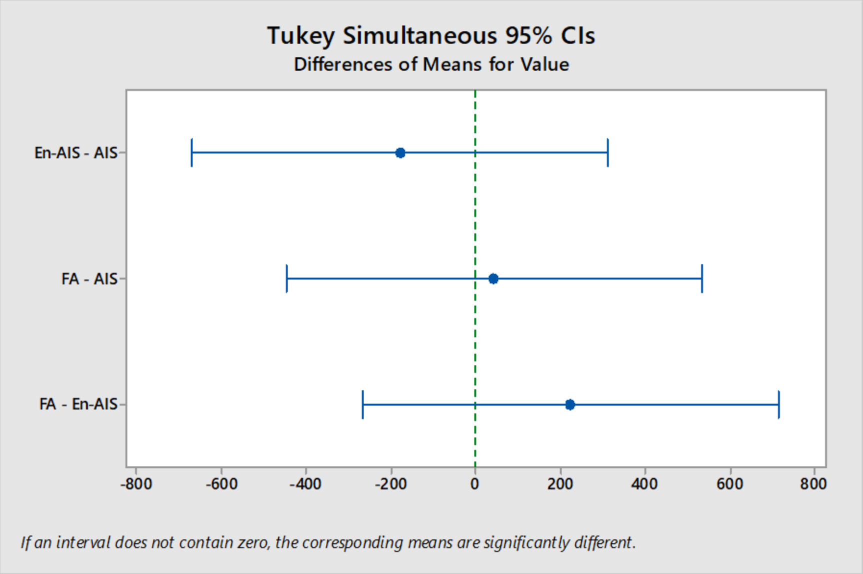


Fig. 6. Tukey HSD test results comparing algorithms' mean performance.

En-AIS also demonstrated better scalability than other algorithms, consistently providing superior solutions even as the problem size increased. Specifically, En-AIS was tested on problem sizes ranging from 15 customers and 4 vehicles to 100 customers and 23 vehicles, consistently outperforming other algorithms. This scalability makes En-AIS particularly well-suited for complex real-world logistics problems, such as the 3L-CVRP, where the number of customers, vehicles, and items can vary significantly.

However, En-AIS required longer computational time than AIS and FA. While En-AIS may require more time, it is essential to note that computational time is just one of several factors to consider when selecting an algorithm. The increased computational time is a trade-off for achieving higher solution quality, making En-AIS a better choice when accuracy is a priority. Other factors, such as accuracy, scalability, and complexity, should also be considered when making a final decision.

To further illustrate En-AIS's impact, Fig. 7 presents the percentage improvement of En-AIS over AIS across various problem instances. Notable improvements are observed, particularly in high-dimensional problems, with peaks as high as 29.79% for specific instances. This visual representation underscores En-AIS's ability to consistently achieve significant performance gains over AIS across all tested scenarios, providing a clear advantage in complex problems where precision and stability are critical.

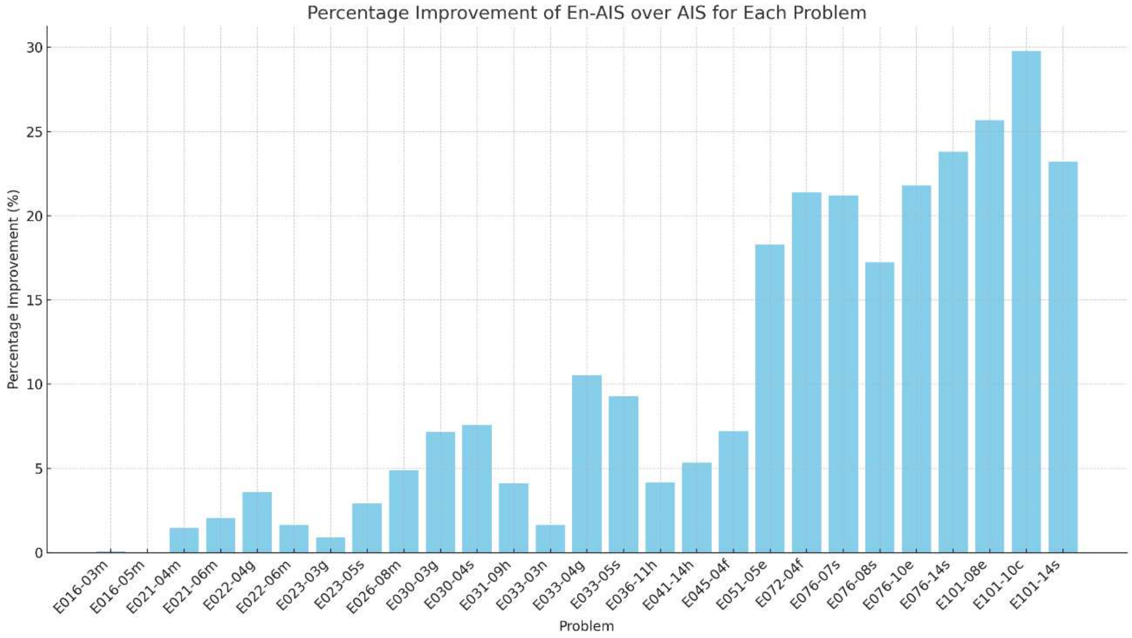


Fig. 7. Percentage Improvement of En-AIS over AIS across various problem instances.

These findings underscore the critical role of incorporating local search mechanisms within heuristic algorithms like AIS. Without local search, AIS often converges prematurely or becomes trapped in suboptimal regions of the solution space, particularly in high-dimensional and highly constrained optimization problems. Integrating local search in En-AIS not only refines the solution quality but also expedites convergence by effectively navigating local optima. This enhancement enables AIS-based algorithms to perform competitively with other advanced metaheuristics, proving that local search is essential for balancing exploration and exploitation, ultimately elevating AIS from a baseline heuristic to a more versatile and powerful optimization tool.

Table 5 presents a consolidated view of the core design elements that distinguish En-AIS from conventional AIS frameworks and typical metaheuristic methods.

Table 5 - Design Features and Performance Benefits of En-AIS

Aspect	Advantage in En-AIS
Flexibility	Dual-mutation strategy increases the ability to escape local optima.
Precision	Bring-i-to-j local search refines routing and loading feasibility directly.
Convergence Speed	Selectively accepts only improved solutions, speeding up convergence.
Diversity Control	Partial elimination and regeneration help maintain population diversity.
Hybrid Design	Integrates local search inside AIS loop, not just as post-processing
Problem Adaptability	Tailored specifically for 3L-CVRP, balancing exploration and exploitation effectively

These combined features contribute to the proposed approach's overall robustness, adaptability, and effectiveness.

Beyond benchmark performance, En-AIS holds significant practical potential in real-world logistics and transportation applications. The algorithm's ability to simultaneously optimize delivery routing and three-dimensional loading makes it suitable for industries such as e-commerce, retail replenishment, and freight logistics, where both spatial constraints and delivery efficiency are critical. For instance, companies involved in multi-stop deliveries or operating vehicle fleets with space limitations can benefit from En-AIS by reducing the number of required trips, improving vehicle utilization, and adapting to diverse item sizes and delivery orders. Its moderate runtime and scalable structure also enable integration into real-time or semi-real-time decision support systems, enhancing operational flexibility in dynamic logistics environments. This adaptability positions En-AIS as a viable solution for complex and evolving challenges in modern supply chains.

Therefore, the results of this benchmarking exercise indicate that En-AIS is the most suitable choice for solving the 3L-CVRP, despite its higher computational time. The algorithm's ability to consistently provide superior solutions and its enhanced local search capabilities—

such as more profound exploration of solution neighborhoods and effective diversification strategies—make it a more robust option for tackling complex routing and loading problems.

Furthermore, although comparative analysis with popular baseline models such as Genetic Algorithm (GA) or Ant Colony Optimization (ACO) is highly valuable, we deliberately limited our comparisons to algorithms implemented under the same experimental framework—namely AIS and FA. This is due to the methodological differences among studies in terms of packing heuristics, constraint formulations, and implementation transparency, which may make direct comparisons with external works less interpretable or fair. Nonetheless, benchmarking with standard metaheuristics remains an important direction for future work, particularly when harmonized loading strategies and benchmark sets become more available.

5. Conclusion

This research introduced an enhanced Artificial Immune System (En-AIS) integrated with the "Bring-i-to-j" local search heuristic to effectively address the complexities of the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP). The En-AIS framework demonstrated significant improvements in solution quality compared to traditional AIS and the Firefly Algorithm (FA) across a comprehensive set of benchmark instances. Key structural features—including dual-mutation strategies, selective update mechanisms, embedded local search, and controlled antibody replacement—enabled En-AIS to achieve faster convergence and superior solution stability, particularly in large and tightly constrained problem scenarios.

A rigorous experimental analysis, supported by statistical validation using ANOVA and Tukey tests, confirmed the practical advantage of En-AIS despite the lack of statistically significant mean differences at the 0.05 level. The practical implications highlighted indicate that En-AIS is well-suited for real-world logistics, particularly for e-commerce, retail replenishment, and freight transportation, where simultaneous optimization of routing and loading under space and delivery constraints is critical. The method's computational efficiency and adaptability make it a strong candidate for integration into dynamic, real-time logistics planning systems.

However, several potential limitations must be acknowledged. First, the benchmark instances considered are static and may not fully represent the dynamic and uncertain conditions inherent in real-world logistics, such as fluctuating customer demands, traffic variability, or unexpected operational disruptions. Second, this study assumes fixed orientations for loaded items and does not explicitly handle practical constraints such as fragility, stacking stability, or loading sequence preferences. Third, while En-AIS performs well on the evaluated scale, computational time may still present challenges when scaled to extremely large or highly time-sensitive operations.

Future research directions include addressing these limitations by extending En-AIS to dynamic and stochastic variants of the 3L-CVRP, incorporating adaptive or machine-learning-based parameter tuning, and exploring hybridization with other optimization frameworks to enhance scalability and robustness. Additionally, practical testing on real-world datasets and explicit handling of item-specific constraints such as fragility or orientation preferences would substantially strengthen the method's industrial applicability and validation.

In conclusion, the proposed En-AIS represents a significant methodological advancement, providing a robust, flexible, and efficient optimization tool for the complex, multi-dimensional challenges encountered in modern logistics and transportation management.

Acknowledgment

The work was supported by the Kasetsart University Research and Development Institute (KURDI), Kasetsart University.

References

Altabeeb, A. M., Mohsen, A. M., Abualigah, L., & Ghallab, A. (2021). Solving capacitated vehicle routing problem using cooperative firefly algorithm. *Applied Soft Computing*, *108*, 107403. https://doi.org/https://doi.org/10.1016/j.asoc.2021.107403

Altabeeb, A. M., Mohsen, A. M., & Ghallab, A. (2019). An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing*, *84*, 105728. https://doi.org/https://doi.org/10.1016/j.asoc.2019.105728

Aytug, H., Knouja, M., & Vergara, F. E. (2003). Use of genetic algorithms to solve production and operations management problems: a review. *International Journal of Production Research*, *41*(17), 3955–4009. http://zerlina.ingentaselect.com/vl=776754/cl=41/nw=1/fm=docpdf/rpsv/cw/tandf/00207543/v41n17/s1/p3955

Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, *39*(9), 2248-2257. https://doi.org/10.1016/j.cor.2011.11.008

Burnet, F. M. (1959). The clonal selection theory of acquired Immunity. *Cambridge University Press*.

Chen, C. S., Lee, S. M., & Shen, Q. S. (1995). An analytical model for the container loading problem. *European Journal of Operational Research*, *80*(1), 68-76. https://doi.org/10.1016/0377-2217(94)00002-t

Chen, M., Huo, J., & Duan, Y. (2023). A hybrid biogeography-based optimization algorithm for three-dimensional bin size designing and packing problem. *Computers & Industrial Engineering*, *180*, 109239. https://doi.org/https://doi.org/10.1016/j.cie.2023.109239

Chi, J., He, S., & Song, R. (2025). Solving capacitated vehicle routing problem with three-dimensional loading and relocation constraints. *Computers & Operations Research*, *173*, 106864. https://doi.org/https://doi.org/10.1016/j.cor.2024.106864

Christensen, S. G., & Rousøe, D. M. (2009). Container loading with multi-drop constraints. *International Transactions in Operational Research*, *16*(6), 727-743. https://doi.org/10.1111/j.1475-3995.2009.00714.x

Dasgupta, D. (2006). Advance in artificial immune systems. *IEEE computational intelligence magazine*, *1*(4), 40-49.

Doerner, K., Fuellerer, G., Hartl, R., Gronalt, M., & Iori, M. (2007). Metaheuristics for the vehicle routing problem with loading constraints. *Networks*, *49*, 294-307. https://doi.org/10.1002/net.20179

Engin, O., & Doyen, A. (2004a). Artificial immune systems and applications in industrial problems. *G. U. Journal of Science*, *17*(1), 71-84.

Engin, O., & Doyen, A. (2004b). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, *20*(6), 1083-1095. http://www.sciencedirect.com/science/article/B6V06-4CT62DH-1/2/765bb875fea11e60b8b71a7fac507c7d

Ferreira, K. M., de Queiroz, T. A., Munari, P., & Toledo, F. M. B. (2024). A variable neighborhood search for the green vehicle routing problem with two-dimensional loading constraints and split delivery. *European Journal of Operational Research*, *316*(2), 597-616. https://doi.org/https://doi.org/10.1016/j.ejor.2024.01.049

Freitas, A., & Timmis, J. (2003). Revisiting the foundations of artificial immune systems: A problem-oriented perspective. In J. Timmis, P. J. Bentley, & E. Hart (Eds.), *Lecture Notes in Computer Science* (Vol. 2787, pp. 229-241). Springer.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, *201*(3), 751-759. https://doi.org/10.1016/j.ejor.2009.03.046

Garrett, S. M. (2005). How do we evaluate artificial immune systems? *Evol Comput*, *13*(2), 145-177. https://doi.org/10.1162/1063656054088512

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transportation Science*, *40*(3), 342-350. https://doi.org/10.1287/trsc.1050.0145

Gimenez-Palacios, I., Alonso, M. T., Alvarez-Valdes, R., & Parreño, F. (2023). Multi-container loading problems with multidrop and split delivery conditions. *Computers & Industrial Engineering*, *175*, 108844. https://doi.org/https://doi.org/10.1016/j.cie.2022.108844

Hart, E., & Timmis, J. (2008). Application areas of AIS: The past, the present and the future. *Applied Soft Computing*, *8*(1), 191-201. http://www.sciencedirect.com/science/article/B6W86-4N1T1KX-2/2/9c970acc97e5f21f3167ce10f7fff74f

Jaradat, M. A. K., & Langari, R. (2009). A hybrid intelligent system for fault detection and sensor fusion. *Applied Soft Computing*, *9*(1), 415-422. http://www.sciencedirect.com/science/article/B6W86-4SJ2WR9-1/2/f4c842db7bc71cb35de51854d5fd8853

Küçük, M., & Topaloglu Yildiz, S. (2022). Constraint programming-based solution approaches for three-dimensional loading capacitated vehicle routing problems. *Computers & Industrial Engineering*, *171*, 108505. https://doi.org/https://doi.org/10.1016/j.cie.2022.108505

Leloup, E., Paquay, C., Pironet, T., & Oliveira, J. F. (2025). A three-phase algorithm for the three-dimensional loading vehicle routing problem with split pickups and time windows. *European Journal of Operational Research*, *323*(1), 45-61. https://doi.org/https://doi.org/10.1016/j.ejor.2024.12.005

Meliani, Y., Hani, Y., Lissane Elhaq, S., & El Mhamedi, A. (2022). A tabu search based approach for the Heterogeneous Fleet Vehicle Routing Problem with three-dimensional loading constraints. *Applied Soft Computing*, *126*, 109239. https://doi.org/10.1016/j.asoc.2022.109239

Placek, M. (2023). *Logistics industry worldwide - statistics & facts*. Retrieved 19 December from https://www.statista.com/topics/5691/logistics-industry-worldwide/

Qi, R., Li, J.-q., & Liu, X.-f. (2023). A knowledge-driven multiobjective optimization algorithm for the transportation of assembled prefabricated components with multi-frequency visits. *Automation in Construction*, *152*, 104944. https://doi.org/https://doi.org/10.1016/j.autcon.2023.104944

Sinha, A., & Goldberg, D. (2003). A survey of hybrid genetic and evolutionary algorithms. *ILLIGAL Technical Report 2003004*. https://doi.org/citeulike-article-id:1460878

Timmis, J. (2007). Artificial Immune Systems - today and tomorrow. *Natural Computing*, *6*, 1-18.

Toth, P., & Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, *123*(1), 487-512. https://doi.org/https://doi.org/10.1016/S0166-218X(01)00351-1

Trachanatzi, D., Rigakis, M., Marinaki, M., & Marinakis, Y. (2020). A firefly algorithm for the environmental prize-collecting vehicle routing problem. *Swarm and Evolutionary Computation*, *57*, 100712. https://doi.org/https://doi.org/10.1016/j.swevo.2020.100712

Wang, Y., Wei, Y., Wang, X., Wang, Z., & Wang, H. (2023). A clustering-based extended genetic algorithm for the multidepot vehicle routing problem with time windows and three-dimensional loading constraints. *Applied Soft Computing*, *133*, 109922. https://doi.org/https://doi.org/10.1016/j.asoc.2022.109922

Wang, Y., Wei, Y., Wei, Y., Zhen, L., & Deng, S. (2025). Collaborative multidepot split delivery network design with three-dimensional loading constraints. *Transportation Research Part E: Logistics and Transportation Review*, *196*, 104032. https://doi.org/https://doi.org/10.1016/j.tre.2025.104032

Yang, X.-S. (2008). *Nature-inspired metaheuristic algorithms*. Luniver press.

Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. Stochastic Algorithms: Foundations and Applications, Berlin, Heidelberg.