E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia Science & Technology Asia
Vol.30 No.4 October - December 2025 Page: [28-54]
Original research article

HORL_2OPT: A Hybrid Reinforcement Learning and Hippopotamus Optimization Algorithm for Bottled Water Delivery Route Optimization

Wanatchapong Kongkaew[1,*], Phattara Khumprom[2], Thanathip Limna[3], Sirirat Suwatcharachaitiwong[1], Dollaya Buakum[1]

[1]*Department of Industrial and Manufacturing Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110 Thailand*
[2]*Graduate School of Management and Innovation, King Mongkut's University of Technology Thonburi, Bangkok 10140 Thailand*
[3]*Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla 90110 Thailand*

Received 24 April 2025; Received in revised form 23 June 2025
Accepted 5 July 2025; Available online 17 December 2025

ABSTRACT

This study presents HORL_2OPT, a hybrid optimization framework developed to address the bottled water delivery routing problem modeled as a Traveling Salesman Problem (TSP). The framework aims to minimize travel distance, enhance computational efficiency, and ensure consistent solutions. HORL_2OPT combines three key components: *Q*-learning for guided initialization, the Hippopotamus Optimization Algorithm (HOA) for global exploration, and a 2-opt heuristic for local route refinement. Tested on 15 TSPLIB benchmarks and 26 real-world cases from a bottled water distributor in southern Thailand, HORL_2OPT consistently produced the best or near-best results. For instance, it achieved a total distance of 8,034.2 in the berlin52 problem, outperforming HOA (12,953.2), DE (25,215.2), and PSO (23,187.0); and in lin318, it achieved 56,695.0 compared to HOA's 85,286.2 and DFA's 122,910.4. In real applications, it generated the shortest or equally optimal routes in 18 of 26 cases, occasionally surpassing LINGO, with most runs completed within 20 seconds. By integrating machine learning, metaheuristics, and local search, HORL_2OPT delivers robust, high-quality solutions suitable for practical logistics and dynamic routing scenarios.

Keywords: Bottled water logistics; Hybrid; Metaheuristic algorithms; Reinforcement learning; Route optimization

***Corresponding author:** wanatchapong.k@psu.ac.th **doi:** 10.14456/scitechasia.2025.68

1. Introduction

The distribution of bottled drinking water plays a crucial role in ensuring accessibility while maintaining cost-effectiveness within logistics systems. Efficient logistics and routing strategies help reduce transportation costs and improve service performance [1]. Traditional routing challenges such as the Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) have been extensively addressed using heuristic, metaheuristic, and mathematical programming techniques [2,3]. However, these approaches often struggle with scalability, adaptability to dynamic conditions, and computational efficiency, limiting their effectiveness in real-world logistics applications [4].

Recent literature has broadened the scope of logistics optimization. A review on logistics efficiency [5] highlights how technologies, including Artificial Intelligence (AI), Machine Learning (ML), and Internet of Things (IoT), can improve performance in transportation, warehousing, and inventory control. Another study [6] focuses on streamlining land transportation and categorizes optimization models. It highlights the use of mixed-integer linear programming and heuristic algorithms to improve supply chain operations.

Advancements in multi-agent systems (MAS) and artificial intelligence have introduced promising solutions for optimizing logistics and delivery networks. Multi-agent frameworks allow for decentralized decision-making, enhancing operational flexibility and efficiency in real-world applications [7]. Reinforcement learning (RL) techniques, particularly *Q*-learning, offer significant potential for improving adaptability in routing scenarios by iteratively refining decision-making based on historical experiences [8]. However, standalone RL solutions face limitations in convergence stability and computational overhead. Thus, hybrid optimization frameworks combining RL with metaheuristic techniques offer significant promise [9].

Metaheuristic algorithms have been widely studied in logistics optimization. Rajwar, Deep, and Das [2] reviewed over 500 metaheuristic algorithms and described their strengths and limitations in large-scale optimization problems. Popular methods, such as Genetic Algorithms (GA), Firefly Algorithm (FA), Particle Swarm Optimization (PSO), and Differential Evolution (DE), show good results but often fall short in dynamic and uncertain environments [4]. To overcome these challenges, researchers have explored multi-agent solutions. Hamrouni, Alutaybi, and Ouerfelli [7] proposed a MAS-based framework for optimizing vehicle routing in electrical vehicle networks, demonstrating improved adaptability and efficiency. Similarly, van der Zwan [10] investigated multi-agent task allocation and path planning strategies for autonomous logistics systems. These works support cooperative frameworks in solving routing problems.

RL applications in logistics decision-making continue to expand. Malathy et al. [8] showed that cooperative RL systems can improve decision-making in production control. Lin et al. [9] introduced a graph-based multi-agent RL model for pollution detection in underwater networks. Their model demonstrated the effectiveness of RL-based strategies in adaptive learning scenarios. Despite these successes, RL alone still faces challenges in real-world logistics. It requires better initialization strategies and stronger convergence properties.

Together, these studies show a need

for flexible, integrated models in logistics. This paper addresses that need by proposing HORL_2OPT. This hybrid model combines *Q*-learning for intelligent initialization, the Hippopotamus Optimization Algorithm (HOA) for search, and the 2-opt heuristic for route refinement. The goal is to improve adaptability and route quality in bottled water delivery.

Although past studies have explored RL and metaheuristics separately, few combine them effectively. The novel proposed model fills this research gap. It unifies the learning ability of RL with the search power of metaheuristics in a single framework. While the HOA has shown good performance in various optimization domains, its application in logistics, especially with enhancements for initial solution quality and local refinement, is limited. To addresses this gap, a novel HORL_2OPT hybrid framework is introduced. It builds on HOA by integrating *Q*-learning to generate high-quality initial solutions and applying the 2-opt heuristic further improvements. The model also introduces a multi-agent component to support decentralized decision-making. Its effectiveness is tested on benchmark datasets and real-world delivery scenarios. Results show that HORL_2OPT consistently outperforms existing methods. Paired t-tests confirm significant improvements in convergence speed, solution quality, and computational efficiency. This validates the proposed framework as a robust and scalable solution for logistics route optimization.

2. Literature Review

2.1 Broader perspectives on logistics optimization

The logistics optimization domain has witnessed extensive research spanning mathematical programming, heuristic strategies, and the emergence of advanced technologies. Recent literature underscores a shift toward integrated and technology-driven solutions. Saidi and Ayadi [5] review and examine key factors influencing logistics performance, including transportation efficiency, inventory management, warehousing, and supply chain coordination. Similarly, another literature survey from [6] outlines techniques for streamlining land transportation, highlighting the growing importance of hybrid and heuristic-based optimization methods.

Technological disruption in logistics has been extensively reviewed. A review in [11] identifies and categorizes disruptive and conventional technologies between 2011 and 2020, emphasizing their effect on operational areas like distribution and warehousing. Additionally, Feng and Ye [12] explores the operations management aspects of smart logistics systems supported by ICT, AI, and IoT, and proposes directions for future research on adaptive logistics systems.

Service-oriented reviews have also emerged. [13] explores the influence of logistics services on firm performance, noting how transportation and information management contribute to competitive advantage. To improve warehouse efficiency, routing and clustering strategies [14] are employed, specifically by developing order picking methods that reduce travel time and operational delays.

Innovative logistics methods are further represented by recent studies on emerging paradigms. Xiaoshan and Weiwei [15] addresses logistics path optimization using deep learning and blockchain, while Toro, Escobar, and Granada [16] offers a review of green VRP, reflecting the sector's alignment with sustainability goals. Likewise,

Singh [17] reviews logistics route planning in circular economy contexts, indicating a growing emphasis on eco-efficient transportation models.

Integrated models, particularly those for location inventory routing problems (LIRPs), were systematically reviewed in [18] through a PRISMA-based assessment. Another study in [19] focuses on logistics optimization during emergencies. These studies reinforce the trend of applying multi-faceted models that balance multiple objectives under complex constraints.

2.2 Metaheuristics and hybridization trends for routing optimization

Metaheuristic algorithms have gained significant attention in solving complex routing problems, such as the TSP and VRP. These approaches provide near-optimal solutions with acceptable computational efficiency, particularly in large-scale optimization scenarios. Among the most widely used metaheuristic methods are GA, PSO, FA, and DE. While these techniques exhibit promising capabilities, they also present specific limitations that necessitate further refinements and hybridization strategies.

GA is a robust evolutionary optimization method that mimics the principles of natural selection to generate high-quality solutions for routing problems. Indrianti et al. [20] applied GA to solve a green vehicle routing problem for LPG distribution, addressing complex constraints and aiming to reduce emissions. However, a major drawback of GA is real-world adaptability, particularly when dealing with social sustainability and uncertain environments. Furthermore, Okulewicz and Mańdziuk [21] found that while GA performs well in long-term optimization strategies, it often requires extensive computational resources, making real-world implementation challenging. This necessitates hybrid approaches that integrate GA with other metaheuristics to enhance convergence efficiency.

PSO is a population-based optimization algorithm inspired by the social behavior of birds and fish. It is particularly effective in exploring solution spaces efficiently; however, it often converges too quickly to suboptimal solutions and struggles to adapt to the complexities of real-world problem environments [22]. Some technique like the update mechanism adjusts the velocity and location of particles exhibiting fitness values within the inferior half of each sub-population was presented through an adaptive mutation multi-particle swarm optimization (AMPSO) [23]. According to [24], PSO is advantageous in transit network optimization because of its computational efficiency and adaptability. However, its performance is highly sensitive to parameter tuning, making it difficult to generalize across various routing scenarios. To mitigate these limitations, recent studies propose hybrid methods to balance exploration and exploitation capabilities in optimization tasks [25].

FA is recognized for its strong multi-objective optimization capabilities. It requires extensive fine-tuning of parameters to avoid stagnation in local optima [25]. Given these challenges, hybrid approaches combining FA with PSO, DE, or other metaheuristics have been proposed to enhance its computational efficiency while preserving its multi-objective optimization strengths. DE is another prominent optimization technique well-suited for continuous optimization problems, including real-time vehicle routing. Okulewicz and Mańdziuk [21] found that while DE outperforms GA and PSO in dynamic vehicle routing scenarios,

its high processing demands restrict its applicability in large-scale logistics networks.

While metaheuristic algorithms have significantly advanced routing optimization, there's still a need for hybrid approaches that combine their strengths to tackle complex routing challenges. Our research addresses this by integrating metaheuristics with machine learning to boost exploration and learning efficiency in routing tasks. The HOA, a recent and promising metaheuristic, has shown competitive performance in general optimization tasks but remains underexplored in the logistics domain. This study introduces HOA into a hybrid framework, integrating it with RL and local search heuristics to enhance its performance in real-world applications.

2.3 Reinforcement learning in routing problems

Reinforcement learning (RL) has emerged as a promising approach for solving dynamic and complex routing problems, particularly the TSP and VRP. Traditional metaheuristic methods such as GA and PSO often struggle with adaptability in dynamic environments, whereas RL-based approaches continuously learn and refine solutions based on dynamic changes. Recent advancements in RL have further enhanced the ability to optimize routes under uncertain and evolving conditions [26]. This suggests that while RL-based routing can significantly enhance adaptability, solution initialization and training efficiency need further optimization. RL-based approaches, particularly Q-learning and its deep learning extensions, have demonstrated significant potential in addressing VRP variants. A deep reinforcement learning (DRL)-based heuristic has developed for solving the Vehicle Routing Problem with Backhauls (VRPBs), wherein an RL agent sequentially constructs optimal vehicle routes by learning from past routing decisions [27]. Their findings highlight the effectiveness of DRL in handling dynamic constraints and adapting to customer demands. However, they also noted key challenges, including high computational costs and solution initialization difficulties, which remain barriers to real-world implementation.

A comprehensive review by [28] analyzed various RL techniques applied to different VRP variants, including dynamic and stochastic VRPs. The authors highlighted that while Q-learning and policy gradient methods are effective for optimizing route planning, Q-learning in particular struggles with solution generalization when applied to large-scale logistics networks. To address these issues, they recommend hybrid models that integrate RL with metaheuristic algorithms to improve performance and scalability. Chen et al. [29] developed an RL-based routing algorithm, RL-Routing, to optimize network paths based on dynamic traffic conditions. Their results showed that RL can effectively manage congestion and improve network efficiency.

Given the challenges associated with standalone RL approaches, hybrid models with local search heuristics have been increasingly explored to balance computational efficiency and optimization quality [30]. Such hybrid strategies suggest a promising direction for future research, where RL can be augmented with traditional optimization methods to overcome its inherent limitations. This research explores hybrid HOA approach that integrate RL as solution initialization strategies to enhance the capability of traditional optimization method.

2.4 Hippopotamus optimization algorithm

The Hippopotamus Optimization Algorithm (HOA) is a recently introduced bio-inspired metaheuristic optimization technique that mimics the movement, social behaviors, and defensive mechanisms of hippopotamuses [31]. Initially designed to tackle benchmark optimization functions, HOA has demonstrated superior performance compared to traditional optimization algorithms. However, its potential applications in logistics, the TSP and VRP remain largely unexplored [32].

Recent studies have proposed modifications to the original HOA to improve its efficiency in various optimization scenarios. Han et al. [33] introduced a Modified Hippopotamus Optimization Algorithm (MHO) aimed at enhancing convergence speed and solution accuracy in global optimization tasks. Their findings indicate that MHO outperforms standard HOA in solving engineering design problems, yet its application in supply chain logistics and routing remains an open research question.

Although HOA has not yet been extensively explored in logistics, recent studies have applied it to network and energy system optimization. Maurya, Tiwari, and Pratap [34] implemented HOA for distribution network reconfiguration, optimizing power flow under different load conditions. Their results highlight HOA's strong performance in multi-objective optimization, indicating its potential applicability in logistics routing problems where multiple constraints and objectives need to be considered. Our preliminary study, Kongkaew et al. [35], presented the HOA for drinking water routing optimizing delivery by mimicking hippopotamus behaviors, improving route efficiency and reducing costs. Despite its efficacy, the proposed method requires fine-tuning of parameters, robustness, and lacks extensive real-world testing.

The integration of HOA with machine learning and network optimization presents another promising avenue for logistics applications. Mamatha and Sateesh Kumar [36] incorporated HOA into deep learning models for channel estimation in millimeter-wave MIMO-OFDM systems, demonstrating the algorithm's capability to enhance optimization in communication networks.

Despite its promising performance in various optimization domains, HOA has yet to be systematically tested in logistics, TSP applications. Key research gaps include the need for empirical validation in logistics optimization, as HOA has been successful in energy and network optimization, but its performance in transportation logistics remains limited. Additionally, hybridization with reinforcement learning, metaheuristics, and local refinement presents a significant research opportunity to enhance adaptability and performance in dynamic routing problems.

By incorporating broader literature insights and addressing the limitations of isolated methods, the proposed hybrid method aims to contribute a robust and generalizable approach to logistics optimization.

3. Methodology

3.1 Mathematical formulation of the bottled water delivery problem

The bottled drinking water routing problem addressed in this study is formulated based on the TSP using the Miller–Tucker–Zemlin formulation [37]. The objective is to determine an optimal route that minimizes the total travel distance required to deliver bottled drinking water to a set

of customer locations, subject to logistical constraints. The mathematical formulation of TSP is expressed as follows.

$$\text{Min } Z = \sum_{i \in N} \sum_{j \in N, j \neq i} d_{ij} x_{ij} \qquad (3.1)$$

Subject to:

$$\sum_{j \in N, j \neq i} x_{ij} = 1, \ \forall i \in N, \qquad (3.2)$$

$$\sum_{i \in N, j \neq i} x_{ij} = 1, \ \forall j \in N, \qquad (3.3)$$

$$u_i - u_j + (|N| - 1)x_{ij} \leq |N| - 2, \qquad (3.4)$$
$$\forall i \in N, j \in N \backslash \{1\}, i \neq j,$$

$$1 \leq u_i \leq |N| - 1, \ \forall i \in N \backslash \{1\}, \qquad (3.5)$$

$$x_{ij} \in \{0, 1\}, \ \forall i, j \in N, j \neq i, \qquad (3.6)$$

where d_{ij} represents the distance between nodes i and j based on the geographical coordinates, x_{ij} is a binary decision variable, where $x_{ij} = 1$ if the delivery route travels directly from node i to node j, and $x_{ij} = 0$ otherwise. The u_i and u_j values are arbitrary real numbers, but they can be converted into a ranked sequence of non-negative integers to represent the order in which nodes are visited. N is the set of nodes, including the depot and all customer locations. The $\{1\}$ is the set of node 1 (representing the depot). In the model, the objective function in Eq. (3.1) aims to minimize the overall travel distance, while Eqs. (3.2)-(3.3) guarantee that each node is entered and exited exactly once. Eqs. (3.4)-(3.5) serve to eliminate subtours, ensuring a single continuous route through all nodes, and Eq. (3.6) defines the binary nature of the decision variable.

3.2 Proposed HORL_2OPT hybrid algorithm

The HOA operates through a population-based evolutionary mechanism that mimics the cooperative behavior of hippopotamuses in search of food and optimal living conditions [31]. It is one of the recent powerful metaheuristics in solving complex combinatorial problems due to its adaptability, particularly in routing and logistics. This work developed the hybridization of the HOA with reinforcement learning and 2-opt heuristic (HORL_2OPT) framework to optimize the bottled drinking water delivery problem. The proposed method combines three components: (i) model-free reinforcement learning via Q-learning for guided initialization, (ii) the Hippopotamus Optimization Algorithm (HOA) for population-based global and local search, and (iii) a 2-opt local search heuristic for solution refinement. The overall goal of the hybridization is to enhance solution quality, improve convergence speed, and increase robustness against local optima. The framework of the proposed HORL_2OPT is shown in Fig. 1.

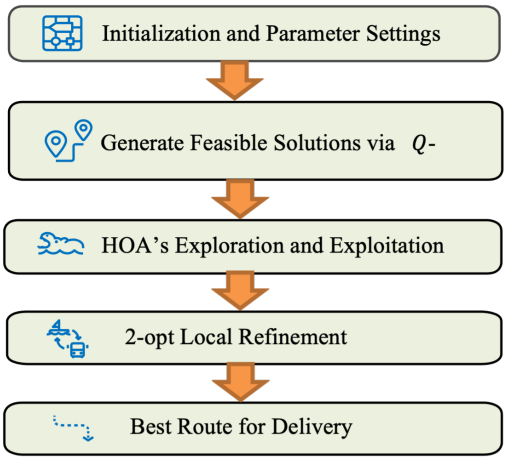

Fig. 1. HORL_2OPT framework for route optimization in bottled water delivery.

3.2.1 Initialization and parameter settings

The initial population includes one Q-learned solution and $N - 1$ randomly

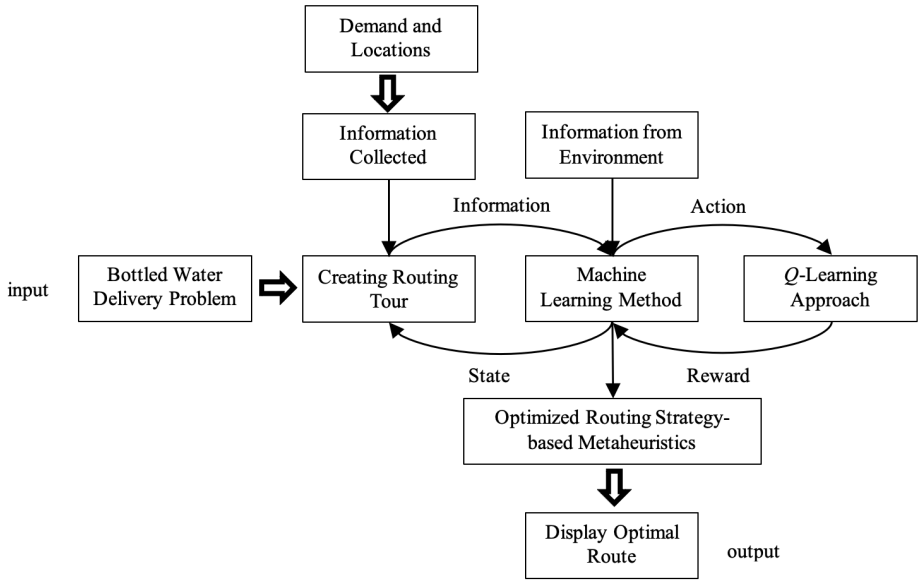

Fig. 2. Reinforcement learning for bottled water delivery routing.

generated agents. Each solution undergoes evaluation, and performance metrics such as best distance and iteration runtime are logged.

The initial population is constructed using a combination of Q-learning and random initialization. Each agent is represented by a real-valued vector within the range [-100,100] , with dimensionality equal to the number of delivery nodes. One agent is guided by Q-learning, trained over 1,000 iterations, and the rest are generated randomly to ensure diversity. Position vectors are converted to valid delivery sequences using the smallest position value (SPV) rule. Fitness is evaluated based on the total travel distance calculated from the distance matrix.

Key parameters include the population size ($nPop$), maximum number of iterations ($MaxGen$), learning rate (α_{RL}), and discount factor (γ_{RL}). These parameters are optimized using the Taguchi method to improve performance and stability. Further explanation of this tuning process is provided in Section 3.4.

3.2.2 Generation of initial feasible solutions

This step focuses on generating a population of initial candidate solutions. To improve early-stage convergence and solution diversity, the method combines Q-learning and random initialization.

In the Q-learning component, the delivery network is modeled as a Markov Decision Process (as in Fig. 2), where

- States (S) represent the set of customer nodes.
- Actions (A) denote the possible transitions from one node to another (i.e., feasible delivery steps.
- Rewards (r) are assigned as the negative distance between two connected nodes, encouraging the agent to find shorter paths.

- Transition $s \rightarrow s'$ occurs upon selecting a delivery from one customer to another.
- Policy is implicitly encoded in the Q-table and updated via the Bellman equation, given by

$$Q(s,a) \leftarrow Q(s,a) + \alpha_{RL}[r + \gamma_{RL} \max_{a'} Q(s',a') - Q(s,a)] \quad (3.7)$$

where $\alpha_{RL} \in (0,1)$ is the learning rate, $\gamma_{RL} \in (0,1)$ is the discount factor, and $\max_{a'} Q(s',a')$ represents the best future reward expected from the next state.

The agent follows an ϵ-greedy policy, where with probability $1-\epsilon$, it selects the action with the highest Q-value, and with probability ϵ, it chooses a random action to ensure exploration. Over iterations, ϵ decays gradually to emphasize exploitation. After a sufficient number of training episodes, a near-optimal route is extracted from the trained Q-table using a greedy rollout. This route is mapped into a real-valued position vector and inserted into the HOA's population as an elite initial candidate.

The remaining population is generated randomly using Eq. (3.8), where each agent's position is represented as a real-valued vector of size equal to the number of customer nodes.

$$X_i : x_{i,j} = lb_j + rand \cdot (ub_j - lb_j), \quad (3.8)$$

where lb_j and ub_j denote the lower and upper bounds of the jth decision variable in D-dimensional space ($j = 1,2,,m$), and $rand$ is a random number between 0 and 1. These vectors are later converted into permutation-based routes using the SPV rule, which transforms each position vector into a discrete sequence within the delivery.

3.2.3 Optimization via HOA

The core iterative optimization of HORL_2OPT is governed by the HOA, a recent metaheuristic inspired by hippopotamus territorial behavior. Each iteration of HOA consists of three behavioral phases, namely exploration, defensive movement, and exploitation [31]. These behaviors are mathematically translated into update rules that govern the movement of candidate solutions in the search space. Let $X \in R^{N \times D}$ represent the population matrix of N agents over a D-dimensional space (number of delivery points), F_i is the fitness of agent $X_i, i = 1,2,,n$, evaluated as the total distance via route and a distance matrix. Each iteration includes the following steps.

Phase 1: Exploration (river and pond movement)

This phase encourages wide search by half the population. Each agent X_i is influenced by the location of the dominant hippopotamus D_{hippo} (i.e., the best position corresponding to the best route) and $\mathcal{MG}_i$ refers to the mean of a randomly selected hippopotamus, including the current considered hippopotamus with equal probability. Two mathematically stochastic position updates are generated as follows:

1. The location of the herd's male hippopotamus in a lake or pond:

$$X_i^{M\,hippo} : x_{i,j}^{M\,hippo} = x_{i,j} + y_1 \cdot (D_{hippo} - I_1 \cdot x_{i,j}), \quad (3.9)$$

for $i = 1,2,...,[\frac{N}{2}]$ and $j = 1,2,...,m$.

2. The position of the female or imma-

ture hippopotamus in the herd:

$$h = \begin{cases} I_2 \times \vec{r}_1 + (\sim \rho_1) \\ 2 \times \vec{r}_2 - 1 \\ \vec{r}_3 \\ I_1 \times \vec{r}_4 + (\sim \rho_2) \\ r_5 \end{cases}, \quad (3.10)$$

$$T = \exp\left(-\frac{t}{MaxGen}\right), \quad (3.11)$$

$$X_i^{FB\ hippo} : x_{i,j}^{FB\ hippo} = \begin{cases} x_{i,j} + h_1 \cdot (D_{hippo} \\ -I_2 \cdot \mathcal{MG}_i), & T > 0.6 \\ \Xi, & else \end{cases}, \quad (3.12)$$

$$\Xi = \begin{cases} x_{i,j} + h_2 \cdot (\mathcal{MG}_i \\ -D_{hippo}), & r_6 > 0.5 \\ lb_j + r_7 \cdot (ub_j \\ -lb_j), & else \end{cases},$$

for $i = 1, 2, ..., [\frac{N}{2}]$ and $j = 1, 2, ..., m$.

where y_1, r_5, r_6, r_7 are random numbers between 0 and 1, and, $\vec{r}_{1,...,4}$ is a random vector between 0 and 1. In Eq. (3.12), h_1 and h_2 are numbers or vectors randomly selected from the five scenarios in the h equation in Eq. (3.10), where ρ_1 and ρ_2 are integer random numbers that can be either 0 or 1. In Eqs. (3.9)-(3.10), $I_1, I_2 \in \{1, 2\}$ are influence weights. In Eq. (3.11), $MaxGen$ denotes the maximum number of iterations. The better of the two candidate positions is retained if it improves fitness. Eqs. (3.13)-(3.14) describe the position update of male and female or immature hippopotamus within the herd. F_i refers to the objective function value (total distance) calculated from the route after decoding using the SPV rule and a distance matrix.

$$X_i = \begin{cases} X_i^{M\ hippo}, & F_i^{M\ hippo} < F_i \\ X_i, & else \end{cases}, \quad (3.13)$$

$$X_i = \begin{cases} X_i^{FB\ hippo}, & F_i^{FB\ hippo} < F_i \\ X_i, & else \end{cases}. \quad (3.14)$$

Using h vectors, I_1 and I_2 scenarios enhance the algorithm's global search and exploration capabilities.

Phase 2: Defensive behavior (predator avoidance)

In this phase, the second half of the population simulates escape behavior by reacting to synthetic predators (random positions):

$$Predator_j = lb_j + \vec{r}_8 \cdot (ub_j - lb_j), \quad \text{for } j = 1, 2, ..., m, \quad (3.15)$$

where $\vec{r}_8$ is a random vector between 0 and 1. Let $\vec{D} = |Predator_j - x_{i,j}|$ denotes the distance of the ith hippopotamus to the predator, and $F_{Predator_j}$ represents the factor of the hippopotamus adopting a defensive behavior to protect itself against the predator. A hippopotamus's defensive behavior changes based on how close a predator is. As in Eq. (3.16), if the predator is in very close proximity ($F_{Predator_j} < F_i$), the hippopotamus will immediately turn and charge to force a retreat. If the predator is at a greater distance, suggesting it's near the territory's boundary, the hippopotamus will turn towards the predator but with restricted movement, aiming to signal its presence within its territory.

$$X_i^{HippoR} : x_{i,j}^{HippoR} = \begin{cases} (\vec{RL} \oplus Predator_j \\ +(\frac{b}{(c - d \times \cos(2\pi g))}) \\ \cdot(\frac{1}{\vec{D}}), & F_{Predator_j} < F_i \\ \vec{RL} \oplus Predator_j \\ +(\frac{b}{(c - d \times \cos(2\pi g))}) \\ \cdot(\frac{1}{2 \times \vec{D} + \vec{r}_g}), & F_{Predator_j} \geq F_i \end{cases}, \quad (3.16)$$

for $i = [\frac{N}{2}] + 1, [\frac{N}{2}] + 2, ..., N$ and $j = 1, 2, ..., m$.

where X_i^{HippoR} represents the hippopotamus's position while facing the predator; $Predator_j$ is defined by Eq. (3.15); $\vec{RL}$ is a random vector with a Lévy distribution described in [31], the values of b, c, d, g are constants drawn from predefined random ranges (referred to in [31]), and $\vec{r}_g$ is a random vector with D-dimensional space. Equation (3.17) outlines a survival mechanism: if F_i^{HippoR} is greater than F_i, the hippopotamus (representing a path in the TSP) has been hunted and is replaced in the herd; otherwise, the hunter flees and the hippopotamus (representing the current of TSP path) remains a viable part of the solution herd. Notable improvement in global search was observed in the second phase. The two phases work together to effectively avoid local minima.

$$X_i = \begin{cases} X_i^{HippoR}, & F_i^{HippoR} < F_i \\ X_i, & F_i^{HippoR} \geq F_i \end{cases}. \quad (3.17)$$

Phase 3: Exploitation (localized foraging)

When a hippopotamus faces multiple predators or cannot defend itself, it flees to the nearest lake or pond. This strategy, which exploits predators' aversion to water, helps the hippo find safety close to its current location. This escape behavior is modeled to enhance local search exploitation within the algorithm. A random position is generated near the hippo's current location using Eqs. (3.18)–(3.20). If this new position improves the cost function, the hippo's position updates, signifying a safer discovery. This approach enhances the algorithm's local search capabilities, crucial for efficiently finding high-quality solutions by enabling effective exploration of promising routes for the bottled drinking water routing problem.

$$ub_j^{local} = \frac{ub_j}{t},$$
$$lb_j^{local} = \frac{lb_j}{t},$$
$$t = 1, 2, ..., MaxGen$$
$$X_i^{Hippo\mathcal{E}} : x_{i,j}^{Hippo\mathcal{E}} = x_{i,j} + r_{10} \cdot (lb_j^{local} + \lambda c \cdot (ub_j^{local} - lb_j^{local}))$$
$$\text{for } i = 1, 2, ..., N, j = 1, 2, ..., m. \quad (3.18)$$

$$\lambda = \begin{cases} 2 \times \vec{r}_{11} - 1 \\ r_{12} \\ r_{13} \end{cases}, \quad (3.19)$$

where $X_i^{Hippo\mathcal{E}}$ denotes the hippopotamus's position (represented the current city in route), which was the starting point for finding the nearest safe place or the most optimal next city. In Eq. (3.18), λ represents a random vector or number, selected from three possible scenarios in Eq. (3.19). r_{10} and r_{13} are random numbers between 0 and 1, while r_{12} follows a normally distributed random number. Additionally, $\vec{r}_{11}$ is a random vector between 0 and 1. These λ approaches enhance local search, leading to better delivery route exploitation. Eq. (3.20) dictates solution updates when fitness improves.

$$X_i = \begin{cases} X_i^{Hippo\mathcal{E}}, & F_i^{Hippo\mathcal{E}} < F_i \\ X_i, & F_i^{Hippo\mathcal{E}} \geq F_i \end{cases}. \quad (3.20)$$

3.2.4 Route refinement via 2-opt local search

Following Phase 3, a 2-opt local optimization procedure refines the best solution obtained in each iteration. This algorithm enhances route quality by iteratively swapping two non-adjacent edges within the tour to reduce the total travel distance. Given a tour with edges (i, j) and (k, l), 2-opt evaluates the benefit of replacing these edges.

The change in total distance is calculated by subtracting the lengths of the old edges and adding the lengths of the new edges. The change in total distance (Δd) is calculated by

$$\Delta d = (d_{ik} + d_{jl}) - (d_{ij} + d_{kl}). \quad (3.21)$$

According to Eq. (3.21), if $\Delta d < 0$ (i.e., the new configuration results in a shorter route), the edges are replaced. This operation is repeated until no further improving swaps are possible. The resulting route is locally optimal and is retained as the refined solution for the current iteration, contributing to a more efficient and higher-quality exploitation of potential routes.

Fig. 3 demonstrates the pseudocode of the HORL_2OPT framework, integrating three optimization layers to ensure global exploration, adaptive learning, and refined local solutions. The combined approach significantly improves convergence speed, route efficiency, and cost-effectiveness for bottled drinking water distribution logistics.

3.3 Data sources and preparation

The experimental analysis was conducted using both benchmark and real-world datasets to validate the proposed hybrid framework. The datasets were carefully selected to ensure the robustness, scalability, and applicability of the HORL_2OPT framework in solving real-world routing and logistics problems. A total of 15 TSP instances were obtained from the TSPLIB library, a widely used benchmark dataset for evaluating optimization algorithms. These instances varied in size, ranging from 16 to 783 nodes, to assess the scalability and efficiency of the proposed approach. The TSPLIB dataset [38] provides predefined coordinates for each node along with an optimal or near-optimal solution for comparison. The selection of instances was based on the complexity and diversity of problem structures, ensuring a comprehensive performance evaluation. Moreover, a real-world dataset was collected over a 10-day period (with 26 problems in total) from a bottled drinking water distribution factory in Southern Thailand. On a daily basis, a company representative will assign the number of customers to be served on each route, ensuring that the total volume of drinking water does not exceed the 100-gallon capacity of the drinking water delivery pickup truck. Each problem was identified using a naming convention. For example, "D1W1T1" designates the first tour for day 1 of week 1. The dataset contained detailed records of delivery operations, distributed across urban and rural areas.

To construct the problem instance for optimization, the distance matrix was constructed using distances between delivery points collected from Google Maps. While acknowledging that real-time traffic conditions can influence actual travel times, these static distances provide a practical and widely accepted basis for initial route optimization studies, particularly for a bottled water delivery service where routes might be planned in advance or predominantly during off-peak hours. Future research will explore integrating dynamic traffic data for enhanced real-world applicability. By combining benchmark datasets with real-world logistics data, the experimental setup ensures that the HORL_2OPT framework is rigorously evaluated, and it is described in the next subsection.

3.4 Experimental setup

The implementation was conducted in a controlled computing environment to ensure reproducibility, reliability, and rig-

Algorithm HORL_2OPT

Input: distance_matrix, learning_rate (α_{RL}), discount_coefficient (γ_{RL}), ε-greedy, RL-training, SearchAgents ($nPop$), Max_iterations ($MaxGen$)
Output: Best_route, Best_fitness

Initialize bounds, dimension, parameters
    Initial_tour ← randperm(dimension)
    RL_tour ← Q_Learning(initial_tour, distance_matrix, alpha, gamma) // Using Eq. (3.7)
    X[1, :] ← DiscreteToPosition(RL_tour)
    X[2:SearchAgents, :] ← Generate population X randomly within bounds using Eq. (3.8)

For each agent i
    fit[i] ← total_distance(distance_matrix, SPV(X[i, :])) // SPV is a smallest position value rule
End For
    Xbest ← best of X, fbest ← best value of fit

For t = 1 to Max_iterations

  # *Phase 1: Exploration*
  For i = 1 **to** SearchAgents / 2
    Generate random movement coefficients
    X_P1 ← explore toward leader via Eq. (3.9)
    X_P2 ← explore toward or away from random group or random reset via Eq. (3.12)
    Evaluate X_P1, X_P2 // Calculate tour distance via "total_distance(·)" function
    Update X[i, :] and fit[i] if improved // Using Eq. (3.13)-(3.14)
  End For

  # *Phase 2: Defense*
  For i = SearchAgents/2+1 **to** SearchAgents
    predator ← random
    X_P3 ← move based on predator distance and Lévy step via Eq. (3.16)
    Evaluate X_P3 // Calculate tour distance via "total_distance(·)" function
    Update X[i, :] and fit[i] if improved // Using Eq. (3.17)
  End For

  # *Phase 3: Exploitation*
  For i = 1 **to** SearchAgents
    X_P4 ← local search within shrinking bounds via Eq. (3.18)
    Evaluate X_P4 // Calculate tour distance via "total_distance(·)" function
    Update X[i, :] and fit[i] if improved // Using Eq. (3.20)
  End For
    Update Xbest and fbest
    (opt2_tour, F_2opt) ← opt_2_iteration(SPV(X[best_index]), distance_matrix, fbest)
    Best_route ← opt2_tour
    Best_fitness ← F_2opt

End For

Return Best_route, Best_fitness

Fig. 3. Pseudocode of the proposed method.

Table 1. Taguchi factor level design for algorithm parameter tuning.

Coded Factor	Level 1	Level 2	Level 3	Level 4	Name of Factor
A	50	100	150	200	Number of Search Agents ($nPop$)
B	100	200	300	500	Number of iterations ($MaxGen$)
C	0.1	0.3	0.6	0.9	learning rate (α_{RL})
D	0.1	0.45	0.75	0.9	discount factor (γ_{RL})

Table 2. Taguchi L16 orthogonal array design matrix for parameter tuning of the proposed algorithm.

Run	Number of populations	Number of iterations	Learning rate	Discount coefficient
1	50	100	0.1	0.1
2	50	200	0.3	0.45
3	50	300	0.6	0.75
⋮	⋮	⋮	⋮	⋮
15	200	300	0.3	0.9
16	200	500	0.1	0.75

orous performance evaluation. The computational experiments were designed to assess the efficiency, robustness, and scalability of the proposed HORL_2OPT algorithm. The proposed algorithm and the compared methods were implemented and tested using MATLAB, while LINGO optimization software (version 20) was used to formulate and solve for optimal or best-known solutions of the mixed-integer linear programming (MILP) model for comparative analysis. This ensured that the performance of HORL_2OPT was benchmarked against exact optimization techniques. The computations were carried out on a personal computer with an Intel(R) Core(TM) i5-12400 processor 2.5GHz and 16GB RAM.

To comprehensively evaluate the proposed HORL_2OPT algorithm, a range of computational parameters was considered to conduct Parameter Tuning using the Taguchi method. The primary HOA parameters, including number of populations (coded as Factor A) and number of iterations (Factor B), and the Q-learning Hyperparameters: learning rate α (Factor C) and discount factor γ (Factor D) were optimized. Each factor was examined. Each factor was examined at four distinct levels, providing a broad exploration space to capture the influence of parameter variation on algorithm performance.

The ranges were chosen based on preliminary testing and domain knowledge to ensure practical relevance and performance sensitivity. A factorial experimental design was employed to identify optimal parameter settings that improve convergence speed and solution accuracy. Table 1 presents the experimental factor levels used in the Taguchi method for tuning the parameters of the proposed algorithm.

Table 2 details the L16 orthogonal array used to systematically explore the parameter space with only 16 experimental runs instead of testing all 256 possible combinations (4^4). This matrix efficiently balances the interactions of four factors across four levels, significantly reducing computational effort. Each run represents a unique combination of values for the number of populations (search agents), the number of iterations, the learning rate, and the discount coefficient. The L16 design provides

a robust foundation for estimating the influence of each parameter on the objective function, such as minimizing total distance in a routing problem.

Fig. 4 illustrates the average total distance obtained from each factor level, offering a visual interpretation of how each parameter influences algorithmic performance. The trend lines suggest which levels of each parameter are associated with better (i.e., lower) distance outcomes. For instance, the optimal zone may be visually observed where the mean distance reaches its minimum, indicating more efficient routing performance. This visual aid complements the numerical results by highlighting sensitivity and response patterns.

Table 3 presents the signal-to-noise (S/N) ratios calculated using the "smaller-the-better" criterion, which is suitable for minimization problems like total distance reduction. The S/N values across four levels of each parameter show relatively small variation, but subtle differences are captured using the delta values. The discount coefficient (Factor D) has the highest delta (0.04), indicating it is the most influential parameter, followed by the number of search agents, learning rate, and number of iterations. The ranking further supports this, placing discount coefficient as the most critical. The prediction result, derived from the optimal levels of each factor (150 SearchAgents, 300 iterations, learning rate of 0.9, and discount coefficient of 0.45), forecasts a minimized total distance value of 61,870 units.

Fig. 5 graphically displays the S/N ratios across the four levels for each parameter. This plot helps in visually identifying the most stable and robust levels that lead to minimum variation in the performance measure. The steepness of the slope for each parameter line indicates its impact; a steeper slope implies higher sensitivity. This figure confirms that the discount coefficient has the strongest effect on algorithm performance, validating the ranking observed in Table 3.

The interaction plots (Fig. 6) offer critical insights into the complex relationships among the optimized parameters: SearchAgents, number of iterations, learning rate, and discount coefficient. Contrary to the Taguchi method's assumption of negligible interactions, these plots clearly demonstrate significant and often strong interdependencies. For instance, the non-parallel and crossing lines between SearchAgents and number of iterations (Fig. 6, top-left), and number of iterations and learning rate (Fig. 6, middle-left), indicate that the optimal setting for one factor heavily depends on the level of another. These pervasive interactions highlight the L16 Taguchi orthogonal array's restricted ability to fully capture complex interdependencies, as its primary focus is on main effects, not higher-order interactions.

While the Taguchi L16 design efficiently narrowed down the optimal parameter ranges, leading to the identified combination of 150 SearchAgents, 300 iterations, a learning rate of 0.9, and a discount coefficient of 0.45 for a minimized total distance, the strong interactions observed suggest this might represent a good local minimum rather than a definitive global optimum. The non-uniform performance across different factor levels, as shown by the non-parallel lines, highlights that the "best" setting for one factor is contingent on others. Therefore, although the L16 design provides a robust solution within its experimental constraints, a more comprehensive optimization explicitly accounting for these crucial interdependencies would be necessary for true global optimality. This might

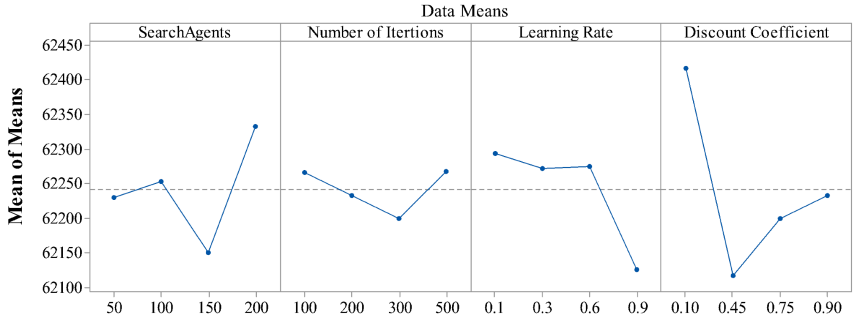

Fig. 4. Plot of the mean of total distance versus considering parameters.

Table 3. Signal to noise ratios (smaller is better) and prediction result.

Level	SearchAgents	Number of iterations	Learning rate	Discount coefficient
1	-95.88	-95.89	-95.89	-95.91
2	-95.88	-95.88	-95.89	-95.86
3	-95.87	-95.88	-95.89	-95.88
4	-95.89	-95.89	-95.87	-95.88
Delta	0.03	0.01	0.02	0.04
Rank	2	4	3	1

Prediction Result				
SearchAgents	Number of iterations	Learning rate	Discount coefficient	Predicted value
150	300	0.9	0.45	61,870

Fig. 5. Plot of the signal-to-noise ratios versus considering parameters.

involve future exploration with a full factorial or a specifically designed fractional factorial experiment.

3.5 Performance metrics and comparisons

To objectively assess the effectiveness of the HORL_2OPT framework, several performance metrics were employed. The total distance (TD) served as the primary objective function, reflecting the overall efficiency of the routing solution, where lower values denote superior performance. Computational time was also measured to evaluate the time required to reach a near-optimal solution, which is particularly important for assessing the practicality of the framework in operational settings. Additionally, convergence speed was considered, defined as the number of iterations needed for the algorithm to stabilize at an optimal or near-optimal solution. To quan-

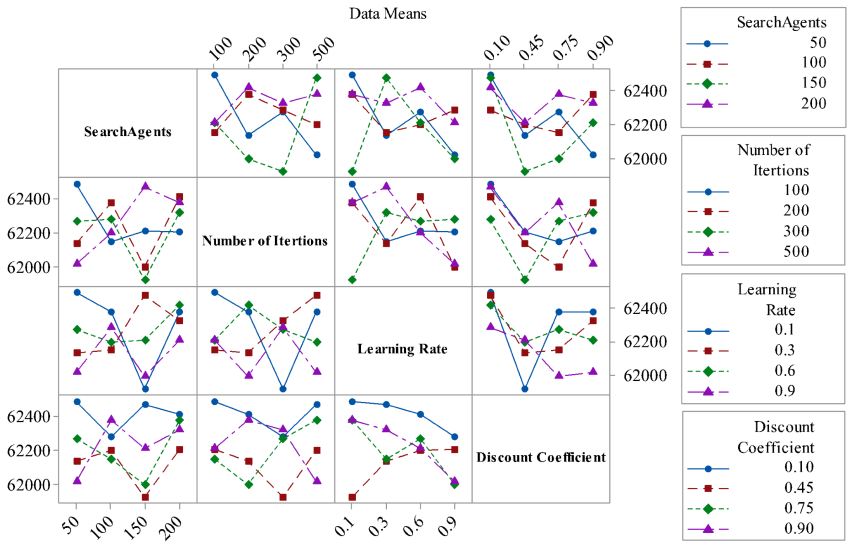

Fig. 6. Interaction plot of the total distance versus considering parameters.

titatively compare performance, the relative improvement of HORL_2OPT over a baseline algorithm A was calculated using a specific mathematical formulation, as follows:

$$\text{Improvement} = \frac{TD_A - TD_{HORL\_2OPT}}{TD_A} \times 100\%, \quad (3.22)$$

where TD_A is the total distance of the baseline algorithm, and $TD_{TD_{HORL\_2OPT}}$ is the total distance obtained using the proposed approach.

To ensure robustness and reliability, the validation technique was employed by the comparison with baseline models. The HORL_2OPT algorithm was benchmarked against traditional metaheuristic approaches, including DE [39, 40], discrete-FA (DFA) [41, 42], PSO [23, 43], AMPSO [23], grey wolf optimization (GWO) [44], and HOA [31, 35]. The comparative analysis focused on solution quality, computational efficiency, and convergence behavior. The experimental setup ensured a rigorous evaluation of the HORL_2OPT hybrid framework. A structured evaluation process using well-defined computational parameters, performance metrics, and validation techniques confirmed the efficacy of HORL_2OPT in optimizing the bottled drinking water distribution problem.

4. Computational Results

This section presents a comprehensive evaluation of the proposed HORL_2OPT framework for optimizing bottled water delivery, using both benchmark and real-world datasets. The performance is assessed through four key lenses: total distance (solution quality), convergence behavior, computational efficiency, and statistically comparative performance against baseline algorithms. According to fairness in algorithm comparisons, especially with AMPSO, the original AMPSO only refers to initializing positions and velocities but does not specify whether a heuristic or random initialization method was used. This lack of detail introduces uncertainty in how much initialization quality may have influenced reported results. Since initialization can significantly impact

Table 4. Parameter settings for different optimization methods.

Method	Population size ($nPop$)	Termination criteria ($MaxGen$)	Key parameters	
			Configuration	Reference
DE	150	300 iterations	F= 1.5, CR = 0.5	[39]
DFA	150	300 iterations	α= 0.2, β_0= 1, γ= 1	[41, 42]
PSO, PSO(G)	150	300 iterations	w_{max}= 0.9, w_{min}= 0.35, C_1= 2, C_2 = 2	[43]
AMPSO, AMPSO(G)	150	300 iterations	w_{max}= 0.9, w_{min}= 0.35, C_1= 2, C_2 = 2	[43]
GWO	150	300 iterations	-	[44]
HOA	150	300 iterations	-	[31, 35]
HORL_2OPT	150	300 iterations	α_{RL}= 0.9, γ_{RL}= 0.45 RL-training = 1,000 iterations	Section 3.4
LINGO	-	Time-bound	Branch-and-bound solver with 43,200 seconds time limit	
Initialization	Random feasible solutions or greedy method (G); permutation encoding of TSP; fixed seed for replicability; five independent replications; $nPop$ and $MaxGen$ were optimized as in Section 3.4.			
System Configuration	Personal computer with an Intel(R) Core(TM) i5-12400 processor 2.5GHz and 16GB RAM.			

optimization, especially in complex combinatorial problems like TSP, this ambiguity affects reproducibility and comparability. To ensure fairness, we implemented two variants of AMPSO: AMPSO (with random initialization) and AMPSO(G) with a greedy heuristic initialization used to test potential performance gains from stronger starting conditions. This logic was applied to PSO, generating two versions: PSO based random initialization and PSO(G) based greedy initialization. Here, the comparative analysis includes nine state-of-the-art algorithms: DE, DFA, PSO, PSO(G), AMPSO, AMPSO(G), GWO, HOA, and the proposed HORL_2OPT. All algorithms were tuned with best-practice or literature-validated parameters (see Table 4), ensuring a fair and reproducible comparison. Notably, HORL_2OPT utilized parameter settings optimized using the Taguchi L16 orthogonal design, which effectively reduced the required number of experimental runs. Although interaction analysis (Fig. 6) revealed interdependencies between factors, the selected parameters provided robustness and consistent computational results.

4.1 Benchmark dataset performance

This section evaluates the performance of the proposed HORL_2OPT algorithm using 15 standard TSP benchmark problems from the TSPLIB library. These problems vary in size and complexity, ranging from 16 to 783 cities. The goal is to compare the solution quality and runtime of HORL_2OPT against eight well-known metaheuristic algorithms: DE, DFA, PSO, PSO(G), AMPSO, AMPSO(G), GWO, HOA, and the known best solutions. Table 5 presents the total distances obtained by each method. HORL_2OPT provided the best or near-best solutions in every case.

For smaller problems, such as ulysses16 and eil76, HORL_2OPT achieved 6,866.6 and 571.2, which are close to the optimal values of 6,859 and 538, respectively. In medium-sized problems like berlin52 (52 nodes), HORL_2OPT produced a route of 8,034.2. This is much better than DE (25,215.2), DFA (19,068.6), and HOA (12,953.2). Only PSO(G) and AMPSO(G) came relatively close with 8,181.0 and 8,186.0

Table 5. The comparative performance on the solution across nine metaheuristic algorithms and the optimal or best-known solutions.

No.	Problem	# Node	Optimal / BKFS	DE	DFA	PSO	PSO (G)	AMPSO	AMPSO (G)	GWO	HOA	HORL_2OPT
1	ulysses16	16	6,859	10,234.0	7,342.0	10,001.6	7,943.0	9,722.4	7,943.0	7,102.8	6,974.6	**6,866.6**
2	berlin52	52	7,542	25,215.2	19,068.6	25,419.2	8,181.0	25,120.8	8,186.0	13,515.4	12,953.2	**8,034.2**
3	eil76	76	538	2,232.0	1,754.0	2,213.0	608.8	2,117.6	608.0	1,368.0	1,119.2	**571.2**
4	lin105	105	14,379	107,698.8	86,014.4	108,519.4	16,938.0	69,716.8	16,938.0	61,908.8	65,744.8	**15,702.6**
5	pr124	124	59,030	618,191.8	517,850.2	625,107.2	67,055.0	431,082.6	67,055.0	426,517.4	375,163.6	**62,461.2**
6	ch150	150	6,528	48,768.4	43,559.2	49,238.6	7,122.6	46,528.0	7,113.0	40,997.8	44,201.4	21,620.8
7	si175	175	21,407	46,029.2	43,793.4	45,925.2	22,022.4	39,076.0	22,022.4	1,308,535.8	1,309,952.4	136,310.6
8	d198	198	15,780	169,693.4	144,918.4	170,900.8	17,735.0	95,257.4	17,756.2	27,890.6	28,443.2	**2,765.0**
9	a280	280	2,579	31,261.0	29,125.6	31,512.6	2,982.6	20,835.0	2,975.2	499,956.4	498,655.8	45,428.8
10	lin318	318	42,029	550,084.6	515,741.2	552,139.4	49,595.6	351,640.6	49,338.0	690,558.0	687,381.4	56,695.0
11	pcb442	442	50,778	729,135.2	697,586.8	735,256.2	59,183.8	585,662.0	59,024.2	25,455.2	40,131.2	**8,340.2**
12	d493	493	35,002	425,295.6	397,830.8	425,652.8	41,099.0	236,830.2	40,897.4	28,653.8	43,185.8	**6,983.4**
13	si535	535	48,450	153,352.2	147,991.0	152,584.2	50,082.6	138,794.2	50,082.4	142,337.4	22,472.6	**16,466.0**
14	d657	657	48,912	818,688.4	777,895.4	819,949.2	58,960.0	510,339.0	59,246.2	382,474.2	112,339.4	**37,534.4**
15	rat783	783	8,806	172,060.0	160,853.2	172,190.8	11,102.8	83,140.0	11,064.0	146,584.2	87,141.8	49,012.6
% improvement over other method			22.4	-723.1	-656.4	-727.0	11.4	-457.3	11.5	-701.2	-602.6	0.0

Remark: BKFS denotes Best-Known Feasible Solution. Bold values in the "HORL_2OPT" column indicate the superior solutions among all algorithms.

Table 6. The comparative performance on runtime of nine metaheuristic algorithms.

No.	Problem	# Node	DE	DFA	PSO	PSO (G)	AMPSO	AMPSO (G)	GWO	HOA	HORL_2OPT
1	ulysses16	16	0.40	10.97	0.12	0.12	0.14	0.14	0.71	0.72	4.86
2	berlin52	52	0.47	14.02	0.17	0.16	0.18	0.18	1.98	1.98	37.36
3	eil76	76	0.52	16.25	0.22	0.20	0.22	0.24	3.27	3.17	76.60
4	lin105	105	0.59	18.90	0.27	0.23	0.26	0.28	4.71	4.68	191.04
5	pr124	124	0.64	20.69	0.30	0.25	0.29	0.31	5.70	5.51	213.56
6	ch150	150	0.71	22.84	0.36	0.36	0.34	0.44	8.10	8.30	493.45
7	si175	175	0.77	25.16	0.41	0.38	0.39	0.45	10.69	10.75	646.34
8	d198	198	0.83	27.19	0.46	0.40	0.45	0.47	13.96	13.90	613.49
9	a280	280	1.04	34.46	0.62	0.55	0.61	0.65	15.96	16.00	2,582.24
10	lin318	318	1.14	39.24	0.71	0.66	0.71	0.80	23.26	23.25	3,069.10
11	pcb442	442	1.49	51.45	0.98	1.01	1.05	1.20	4.47	14.59	166.82
12	d493	493	1.62	55.87	1.09	1.16	1.16	1.36	6.91	21.17	405.32
13	si535	535	1.74	61.05	1.23	1.28	1.36	1.56	9.49	27.12	521.26
14	d657	657	2.08	73.33	1.50	1.66	1.63	1.91	26.23	65.27	5,982.44
15	rat783	783	2.47	89.44	1.84	2.14	1.99	2.45	28.56	70.79	9,080.10

respectively, but HORL_2OPT still performed better. Larger instances, such as lin318, d657, and rat783, further show the advantage of the proposed method. In lin318 (318 nodes), HORL_2OPT recorded 56,695.0, while DE, FA, PSO and GWO exceeded 500,000. On rat783, HORL_2OPT achieved 49,012.6, much lower than DFA (160,853.2) and HOA (87,141.8). While the original HOA already demonstrates strong performance, HORL_2OPT consistently outperforms it, as well as DE, DFA, PSO, and GWO, across almost all benchmark instances in terms of total distance.

These improvements are the result of combining three key components: *Q*-learning for intelligent initialization, HOA for exploration and exploitation, and 2-opt for refining the best routes. The hybrid structure allows the algorithm to avoid poor local solutions and focus on high-quality paths from the early stages of the search.

Table 6 shows the runtime for each method. HORL_2OPT generally requires more time than basic algorithms such as DE and PSO. This substantial computational overhead is an expected trade-off, resulting from combining *Q*-learning processes and the added complexity of local

search techniques like 2-opt, which further contribute to the increased computational time. For example, lin105 took 191.04 seconds, while DE, PSO, PSO(G), AMPSO and AMPSO(G) needed less than one second. However, the solutions from HORL_2OPT are significantly more accurate.

Although HORL_2OPT is slower, its performance remains acceptable for offline route planning. The extra time is a trade-off for higher solution quality. In most cases, the runtime remains under practical limits, even for larger datasets. In summary, HORL_2OPT consistently produces better routing results than the other methods tested. It offers a strong balance between accuracy and robustness, especially for cases where route quality matters more than speed.

4.2 Evaluation on real-world routing problems

To test the proposed method in practical conditions, we used 26 routing problems from a bottled water distributor in Southern Thailand. These problems were collected over 10 operational days and included various delivery node configurations and route complexities.

Table 7 presents the total distances obtained by each algorithm. HORL_2OPT produced the shortest or equally optimal routes in almost all 26 cases. In 18 of these, it matched or outperformed the best-known feasible solutions provided by LINGO. For instance, in instance D10W2T1, HORL_2OPT achieved a total distance of 56.9 km, which is equal to PSO(G) and AMPSO(G). This improved upon LINGO's result of 58.7 km and was significantly better than HOA (59.1 km) and DFA (114.2 km).

Furthermore, HORL_2OPT outperformed both AMPSO and AMPSO(G) in most cases, demonstrating that the hybridization of reinforcement learning, HOA, and 2-opt search effectively improves route quality even when compared to algorithms using heuristic initialization.

Runtime performance is summarized in Table 8. While HORL_2OPT required more computational time than DE, PSO, AMPSO and GWO, it completed all problem instances in less than 20 seconds. In contrast, LINGO failed to return solutions within the allowed 12-hour time limit for more than half of the instances. This makes HORL_2OPT a feasible option for real-world deployment, balancing speed and solution accuracy.

The findings confirm that HORL_2OPT is capable of solving practical logistics problems with high reliability and competitive efficiency. Its ability to outperform both commercial solvers and established metaheuristics across a diverse set of real-world cases highlights its potential for use in delivery route planning and related logistics applications.

4.3 Convergence analysis

The convergence behavior of HORL_2OPT was assessed using two representative cases: the benchmark instance si175 and the real-world case D10W2T1. Fig. 7 illustrates the convergence trend of HORL_2OPT on the benchmark instance si175. The algorithm demonstrates rapid progress during the early iterations and quickly reaches a stable solution. Compared to AMPSO, PSO, and DE, the HORL_2OPT curve shows a smoother and more direct descent toward a lower-cost solution. This indicates that the combination of Q-learning initialization and HOA-guided global search effectively positions the algorithm for faster con-

Table 7. The total distances of case studies across different methods.

Problem	# Node	Current	LINGO solution type	LINGO	DE	DFA	PSO	PSO (G)	AMPSO	AMPSO (G)	GWO	HOA	HORL_2OPT
D1W1T1	19	56.86	Optimal	**48.4**	104.4	65.8	109.9	50.0	89.3	50.0	53.8	55.9	**48.4**
D1W1T2	33	75.09	BKFS	52.5	168.8	78.3	161.8	48.3	148.3	48.3	62.7	52.9	51.8
D1W1T3	24	52.42	BKFS	**50.1**	95.1	63.7	97.0	51.5	79.1	51.5	52.2	50.9	**50.1**
D2W1T1	27	54.16	Optimal	41.9	94.8	56.9	95.3	42.5	82.8	42.5	53.0	47.1	44.5
D2W1T2	31	38.23	BKFS	33.2	62.7	47.0	62.4	33.3	51.7	33.3	38.6	32.8	**32.3**
D2W1T3	32	34.00	BKFS	28.2	69.9	36.7	70.8	28.8	52.7	28.8	32.6	28.8	28.5
D3W1T1	25	57.80	Optimal	50.1	90.0	67.5	95.1	50.3	75.4	50.3	55.7	53.7	53.1
D3W1T2	20	48.72	Optimal	**44.7**	84.5	55.8	89.0	**44.7**	70.4	**44.7**	48.1	44.8	**44.7**
D3W1T3	21	32.86	Optimal	**32.4**	53.2	36.0	53.1	33.5	45.9	33.5	32.8	32.6	**32.4**
D4W1T1	22	54.36	Optimal	**50.6**	76.7	61.9	76.4	51.5	67.9	51.3	54.8	50.9	**50.6**
D4W1T2	29	56.73	BKFS	**44.5**	121.6	77.8	131.9	**44.5**	83.7	**44.5**	52.9	45.0	**44.5**
D5W1T1	19	45.73	Optimal	**45.1**	88.6	51.4	90.3	**45.1**	79.1	**45.1**	**45.1**	**45.1**	**45.1**
D5W1T2	35	36.86	BKFS	32.9	95.1	55.5	95.5	**32.8**	65.0	**32.8**	38.6	**32.8**	**32.8**
D6W2T1	28	61.87	BKFS	44.0	133.3	71.4	127.0	**45.8**	106.2	**45.8**	53.4	46.8	46.5
D6W2T2	33	39.63	BKFS	**35.3**	92.7	60.9	96.9	**34.7**	72.2	**34.7**	**39.6**	**34.7**	**34.7**
D6W2T3	14	29.57	Optimal	**29.5**	31.9	29.7	31.9	**29.5**	30.8	**29.5**	**29.5**	**29.5**	**29.5**
D7W2T1	23	64.43	Optimal	**55.9**	79.1	65.4	80.1	**56.9**	77.0	**56.9**	59.6	62.4	**55.9**
D7W2T2	20	59.32	Optimal	**44.1**	84.0	49.9	84.2	**44.1**	69.2	**44.1**	**44.4**	**44.1**	**44.1**
D7W2T3	16	34.06	Optimal	**31.4**	38.7	31.5	37.4	**31.4**	33.6	**31.4**	**31.4**	**31.4**	**31.4**
D8W2T1	20	52.75	Optimal	**48.2**	78.2	57.1	80.5	**48.8**	69.8	**48.8**	51.8	52.5	51.9
D8W2T2	39	78.15	BKFS	**47.8**	165.7	100.8	158.0	**47.8**	113.0	**47.8**	77.3	48.0	**47.8**
D8W2T3	27	28.26	BKFS	23.3	52.0	32.6	53.2	24.8	42.0	24.8	27.1	23.4	23.5
D9W2T1	28	56.80	Optimal	48.1	85.3	62.5	84.1	48.8	77.6	48.8	55.6	52.9	51.7
D9W2T2	39	61.31	BKFS	36.7	91.3	58.7	86.7	**39.3**	79.0	**39.3**	47.4	45.1	36.9
D10W2T1	40	101.26	BKFS	58.7	213.6	114.2	215.5	**56.9**	148.3	**56.9**	84.9	59.1	**56.9**
D10W2T2	38	45.47	BKFS	34.4	74.6	50.0	75.4	**32.9**	66.5	**32.9**	41.3	34.4	**32.9**
% improvement over other method				1.0	-120.0	-39.6	-121.3	0.4	-79.3	0.4	-14.7	-3.2	-

Remark: BKFS denotes Best-known feasible solution. The bold values in the table indicate the superior or equivalent solutions across methods.

Table 8. The computational time of case studies across different methods.

Problem	LINGO	DE	DFA	PSO	PSO (G)	AMPSO	AMPSO (G)	GWO	HOA	HORL_2OPT
D1W1T1	908.40	0.41	11.58	0.122	0.125	0.139	0.145	0.84	3.89	5.73
D1W1T2	43,200	0.43	12.16	0.141	0.139	0.148	0.157	1.30	5.71	11.31
D1W1T3	43,200	0.40	11.00	0.123	0.124	0.132	0.139	0.96	4.56	7.37
D2W1T1	7,985.81	0.41	11.25	0.127	0.130	0.137	0.149	1.09	4.92	10.81
D2W1T2	43,200	0.41	11.99	0.135	0.135	0.144	0.155	1.24	5.44	12.60
D2W1T3	43,200	0.41	11.60	0.136	0.136	0.146	0.156	1.34	5.58	13.07
D3W1T1	22.03	0.40	11.30	0.126	0.127	0.133	0.145	1.05	4.66	8.05
D3W1T2	10,754.69	0.39	11.81	0.117	0.120	0.125	0.133	0.86	3.96	6.19
D3W1T3	33,337.49	0.39	10.91	0.119	0.131	0.129	0.134	0.90	4.16	6.81
D4W1T1	11,409.62	0.39	11.26	0.120	0.129	0.130	0.129	0.94	4.28	6.99
D4W1T2	43,200	0.41	11.47	0.130	0.134	0.140	0.148	1.15	5.20	11.02
D5W1T1	1,913.27	0.39	11.54	0.117	0.123	0.126	0.136	0.82	3.83	5.66
D5W1T2	43,200	0.42	11.97	0.142	0.139	0.149	0.156	1.29	5.98	12.78
D6W2T1	43,200	0.40	11.78	0.129	0.129	0.139	0.145	1.15	5.05	8.99
D6W2T2	43,200	0.42	12.25	0.138	0.137	0.146	0.156	1.22	5.72	13.49
D6W2T3	8.28	0.37	11.84	0.112	0.117	0.120	0.126	0.63	3.20	4.09
D7W2T1	39.45	0.39	10.95	0.121	0.122	0.132	0.139	0.94	4.38	8.18
D7W2T2	10,985.68	0.39	11.11	0.117	0.121	0.126	0.135	0.86	3.96	5.89
D7W2T3	26.56	0.38	10.77	0.113	0.114	0.123	0.129	0.72	3.44	4.70
D8W2T1	12.58	0.39	11.57	0.117	0.121	0.128	0.137	0.84	3.95	5.91
D8W2T2	43,200	0.43	13.19	0.148	0.143	0.157	0.163	1.52	6.48	17.76
D8W2T3	43,200	0.40	11.12	0.128	0.129	0.137	0.145	1.10	4.88	7.91
D9W2T1	43,200	0.40	11.32	0.128	0.132	0.138	0.146	1.14	5.05	9.25
D9W2T2	43,200	0.43	12.95	0.149	0.146	0.160	0.163	1.43	6.46	17.05
D10W2T1	43,200	0.43	13.08	0.149	0.143	0.158	0.162	1.48	6.62	18.59
D10W2T2	43,200	0.43	12.15	0.145	0.142	0.151	0.163	1.45	6.36	15.19

vergence. Additionally, the use of 2-opt refinement helps reduce oscillations and accelerates the stabilization process. The result confirms the algorithm's ability to find high-quality solutions in fewer iterations.

Fig. 8 presents the convergence behavior on the real-world case D10W2T1. HORL_2OPT again shows superior performance, reaching a stable route distance in the early iterations. In contrast, other algorithms either stagnate early or take longer to reach suboptimal solutions. The consistent performance across both benchmark and real-world datasets highlights the robustness of the hybrid design. The convergence patterns provide further evidence that HORL_2OPT can maintain strong efficiency and solution quality across varied problem types and scales.

4.4 Statistical validation

To evaluate the significance of the observed performance differences, paired *t*-tests were conducted between HORL_2OPT and each comparison algorithm. The tests only focused on the total distance obtained across all real-world instances. The results of the statistical tests are summarized in Table 8, where each *p*-value indicates the probability of observing the given difference under the null hypothesis of no significant difference. Negative values in parentheses denote the average difference in favor of HORL_2OPT.

Results indicated that HORL_2OPT significantly outperformed most competing methods, with *p*-values less than 0.05 in the majority of cases. The improvements were consistent, especially when compared to algorithms using random initialization such as DE, PSO, and AMPSO. Although HORL_2OPT required more time for computation, its solution quality justified the added effort.

Table 9. Paired-*t* test among two considering methods.

Methods	Costs	Times
HORL_2OPT vs. Current	1.000 (-4.73)	-
HORL_2OPT vs. LINGO	0.081 (1.44)	1.000 (-6.23)
HORL_2OPT vs. DE	1.000 (-7.33)	0.000 (11.53)
HORL_2OPT vs. DFA	1.000 (-6.08)	0.992 (-2.59)
HORL_2OPT vs. PSO	1.000 (-7.57)	0.000 (11.86)
HORL_2OPT vs. PSO(G)	0.293 (0.55)	0.000 (11.85)
HORL_2OPT vs. AMPSO	1.000 (-7.48)	0.000 (11.84)
HORL_2OPT vs. AMPSO(G)	0.714 (0.57)	0.000 (11.83)
HORL_2OPT vs. GWO	1.000 (-4.29)	0.000 (11.30)
HORL_2OPT vs. HOA	0.996 (-2.93)	0.000 (7.77)

In particular, HORL_2OPT outperformed AMPSO with random initialization (p=1.000) and showed competitive results even against AMPSO(G), which used heuristic-based initialization (p=0.714). This indicates that HORL_2OPT maintains high performance regardless of the initial solution quality. The statistical validation confirms that the proposed hybrid framework delivers reliable results and can operate effectively under both standard and challenging initialization settings. When considering both fairness and consistency, it provides a strong and practical solution for real-world logistics.

5. Conclusion

This research was motivated by the need to improve routing efficiency in real-world logistics, particularly for bottled water distribution systems. Traditional metaheuristic approaches often struggle with premature convergence, limited diversity in initial solutions, and insufficient local refinement strategies, which can reduce their effectiveness in solving complex vehicle routing problems. To overcome these challenges, this study introduced HORL_2OPT, a hybrid optimization framework that integrates the Hippopotamus Optimization Algorithm (HOA), *Q*-learning-based rein-

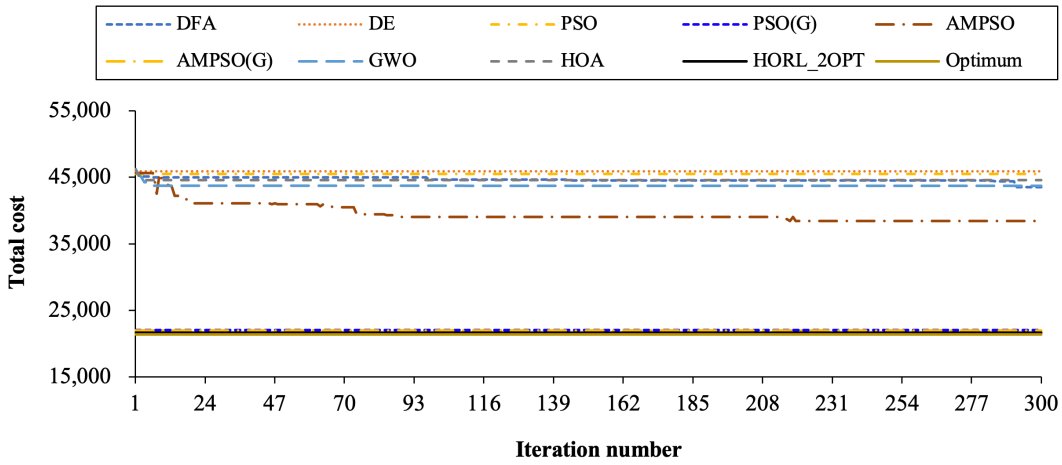

Fig. 7. Convergence plot of HORL_2OPT against other benchmarking algorithms for the si175.

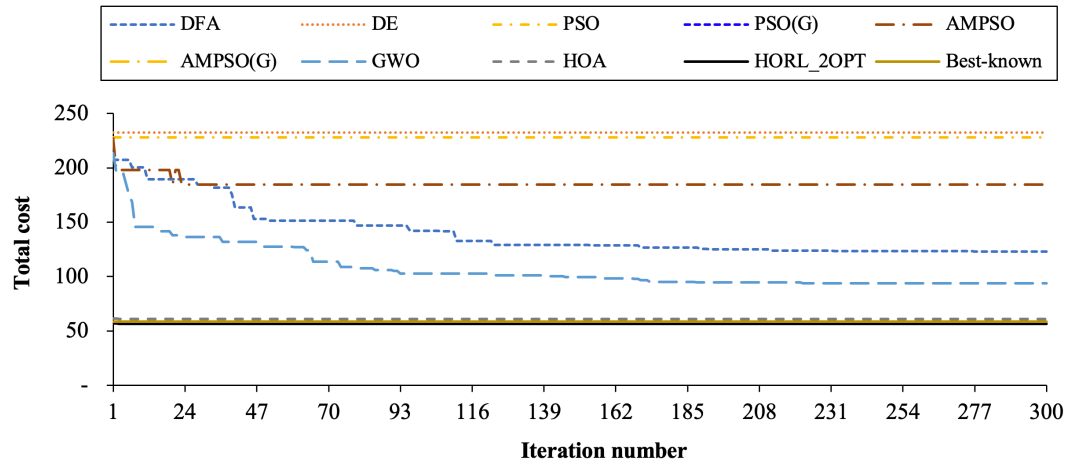

Fig. 8. Convergence plot of HORL_2OPT against other benchmarking algorithms for the D10W2T1.

forcement learning, and the 2-opt local search heuristic. The proposed model aims to improve both global exploration and local exploitation by combining the strengths of these three components into a unified algorithm.

The method was evaluated through extensive experiments on both synthetic benchmark datasets from TSPLIB and real-world delivery data collected from a bottled water company in Southern Thailand. The algorithm's performance was benchmarked against well-known methods, including DE, PSO, GWO, and AMPSO under both random and heuristic initialization scenarios. Results consistently showed that HORL_2OPT either matched or outperformed these methods across a variety of problem sizes and configurations. For instance, on the berlin52 problem, it achieved a total route distance of 8,034.2 compared to 8,186.0 by AMPSO and 25,215.2 by DE. Similarly, in real-world scenarios, HORL_2OPT produced shorter routes than both LINGO and heuristic-initialized competitors. Statistical analysis confirmed the

significance of these improvements, reinforcing the robustness of the proposed hybrid model.

The main finding confirms that integrating reinforcement learning (*Q*-learning) with metaheuristics, particularly HOA, and local search (2-opt) can significantly enhance solution quality and convergence by providing superior initial solutions and refined local searches. This integration not only improves the algorithm's ability to escape local optima but also reduces the number of iterations needed to reach near-optimal solutions. The method's reliability across both controlled and real-world datasets demonstrates its flexibility and scalability for a wide range of routing applications.

From a practical perspective, the HORL_2OPT framework shows promise for deployment in other logistics systems where rapid and high-quality routing is essential. Its structure can be generalized to support tasks such as emergency medical delivery, inventory restocking, and last-mile distribution. Moreover, the ability to accommodate both algorithmic efficiency and field-based variability enhances its applicability in decision-making for industries that demand high service levels and cost efficiency.

Future research will focus on enhancing the model's responsiveness to real-time factors. In particular, integrating dynamic traffic data into the optimization process will allow the system to adapt routes based on current road conditions, improving its real-world performance. Additionally, extending the model to include environmental objectives, such as minimizing fuel usage and carbon emissions, will support broader goals in sustainable transportation and smart logistics. These directions will strengthen the practical value of HORL_2OPT and contribute to the development of more intelligent, eco-conscious routing systems.

Acknowledgements

This research was supported by National Science, Research and Innovation Fund (NSRF) and Prince of Songkla University (Ref. No. ENG6701183S). The authors would like to thank the case study company for providing the operational data necessary for this research.

References

[1] Oonchokdee N, Pichitlamken J, Wangwatcharakul W. Cost reduction by fleet planning for parcel delivery service. Sci Technol Asia. 2024;29(2):74–84.

[2] Rajwar K, Deep K, Das S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artif Intell Rev. 2023;56(11):13187–257.

[3] Toaza B, Esztergár-Kiss D. A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. Appl Soft Comput. 2023;148:110908.

[4] Abualigah L, Elaziz MA, Khasawneh AM, Alshinwan M, Ibrahim RA, Al-Qaness MA, et al. Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. Neural Comput Appl. 2022;34(6):4081–110.

[5] Saidi AKAA, Ayadi N. Enhancing logistics efficiency: a comprehensive study of Royal Solidex company. IJRDO [Internet]. 2025 [cited 2025 Jun 3]. Available from: https://doi.org/10.53555/bm.v11i1.6217

[6] Tagaro JC, Valda DTS, Villa SE III, Yasuda MD. Logistics Optimization: A Literature Review of Techniques for Streamlining Land Transportation in Supply Chain Operations [Internet]. 2025 [cited 2025 Jun 3]. Available from: https://www.researchgate.net/publication/384246072

[7] Hamrouni C, Alutaybi A, Ouerfelli G. Multi-agent mapping and tracking-based electrical vehicles with unknown environment exploration. World Electr Veh J. 2025;16(3):162.

[8] Malathy V, Al-Jawahry HM, Madhura G K, Suganya G, Rashmi P. A reinforcement learning method in cooperative multi-agent system for production control system. In: Proceedings of the 2024 International Conference on Data Science and Network Security (ICDSNS); 2024. p. 1–4.

[9] Lin C, Han G, Zhang T, Shah SBH, Peng Y. Smart underwater pollution detection based on graph-based multi-agent reinforcement learning towards AUV-based network ITS. IEEE Trans Intell Transp Syst. 2023;24(7):7494–505.

[10] Van der Zwan M. Multi-agent task allocation and path planning for autonomous ground support equipment [master's thesis]. Delft: TU Delft; 2023.

[11] Cano JA, Gómez-Montoya RA, Salazar F, Cortés P. Disruptive and conventional technologies for the support of logistics processes: a literature review. Int J Technol. 2021;12(3):448–60.

[12] Feng B, Ye Q. Operations management of smart logistics: a literature review and future research. Front Eng Manag. 2021;8(3):344–355.

[13] Al Zadajali A, Ullah A. The effectiveness of logistics services on firms' performances – a literature review. AJEBI. 2024;3(1):125–32.

[14] Rymarczyk P, Bogacki S, Figura C, Rutkowski M, Staliński P. Optimizing order picking processes in warehouses: strategies for efficient routing and clustering. J Mod Sci. 2024;57(3):467–484.

[15] Xiaoshan Y, Weiwei G. Research on logistics distribution route optimization based on deep learning model and block chain technology. 3C Empresa. 2023;12(1):68–85.

[16] Toro OEM, Escobar ZAH, Granada EM. Literature review on the vehicle routing problem in the green transportation context. Luna Azul. 2016;42:362–87.

[17] Singh H. Logistics optimization for circular economy: a vehicle route planning based analysis [master's thesis]. Vaasa: University of Vaasa; 2024.

[18] Liu L, Lee LS, Seow HV, Chen CY. Logistics center location-inventory-routing problem optimization: a systematic review using PRISMA method. Sustainability. 2022;14(23):15853.

[19] Tan K, Liu W, Xu F, Li C. Optimization model and algorithm of logistics vehicle routing problem under major emergency. Mathematics. 2023;11(5):1274.

[20] Indrianti N, Leuveano RAC, Abdul-Rashid SH, Ridho MI. Green vehicle routing problem optimization for LPG distribution: genetic algorithms for complex constraints and emission reduction. Sustainability. 2025;17(3):1144.

[21] Okulewicz M, Mańdziuk J. A metaheuristic approach to solve dynamic vehicle routing problem in continuous search space. Swarm Evol Comput. 2019;48:44–61.

[22] Abualigah L. Particle swarm optimization: advances, applications, and experimental insights. Comput Mater Contin. 2025;82(2):1539–92.

[23] Gao M, Fu X, Dong G, Li H. An adaptive mutation multi-particle swarm optimization for traveling salesman problem. In: Proceedings of the 3rd International Conference on Material, Mechanical and Manufacturing Engineering; 2015. p. 1003–1007.

[24] Iliopoulou C, Kepaptsoglou K, Vlahogianni E. Metaheuristics for the transit route network design problem: a review and comparative analysis. Public Transp. 2019;11(3):487–521.

[25] Rahman MA, Sokkalingam R, Othman M, Biswas K, Abdullah L, Kadir EA. Nature-inspired metaheuristic techniques for combinatorial optimization problems: overview and recent advances. Mathematics. 2021;9(20):2633.

[26] Pan W, Liu SQ. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. Appl Intell. 2022;53(1):405–22.

[27] Wang C, Cao Z, Wu Y, Teng L, Wu G. Deep reinforcement learning for solving vehicle routing problems with backhauls. IEEE Trans Neural Netw Learn Syst. 2025;36(3):4779–93.

[28] Raza SM, Sajid M, Singh J. Vehicle routing problem using reinforcement learning: recent advancements. In: Gupta D, Sambyo K, Prasad M, Agarwal S, editors. Lect Notes Electr Eng. 2022. p. 269–80.

[29] Chen YR, Rezapour A, Tzeng WG, Tsai SC. RL-routing: an SDN routing algorithm based on deep reinforcement learning. IEEE Trans Netw Sci Eng. 2020;7(4):3185–99.

[30] Zhao J, Mao M, Zhao X, Zou J. A hybrid of deep reinforcement learning and local search for the vehicle routing problems. IEEE Trans Intell Transp Syst. 2021;22(11):7208–18.

[31] Amiri MH, Hashjin NM, Montazeri M, Mirjalili S, Khodadadi N. Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. Sci Rep. 2024;14:5032.

[32] Mehta P, Sait S, Yildiz B, Yildiz A. Enhanced hippopotamus optimization algorithm and artificial neural network for mechanical component design. Mater Test. 2025;67(4):655–62.

[33] Han T, Wang H, Li T, Liu Q, Huang Y. MHO: a modified hippopotamus optimization algorithm for global optimization and engineering design problems. Biomimetics. 2025;10(2):90.

[34] Maurya P, Tiwari P, Pratap A. Application of the hippopotamus optimization algorithm for distribution network reconfiguration with distributed generation considering different load models for enhancement of power system performance. Electr Eng. 2025;107:3909–46.

[35] Kongkaew W, Koohathongsumrit N, Pongsathornwiwat A, Suwatcharachaitiwong S, Buakum D, Limna T, Khumprom P. Application of hippopotamus optimization algorithm for the routing problem in drinking water factory. In: Proceedings of the 24th Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS); 2024. p. 332–37.

[36] Mamatha MC, Sateesh Kumar HC. Channel estimation using deep learning techniques with hippopotamus optimization algorithm for millimeter wave MIMO-OFDM system. Int J Intell Eng Syst. 2024;17(6):1314–24.

[37] Campuzano G, Obreque C, Aguayo MM. Accelerating the Miller–Tucker–Zemlin model for the asymmetric traveling salesman problem. Expert Syst Appl. 2020;148:113229.

[38] Reinelt G. TSPLIB—a traveling salesman problem library. ORSA J Comput. 1991;3(4):376–84.

[39] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11:341–59.

[40] Ali IM, Essam D, Kasmarik K. A novel design of differential evolution for solving discrete traveling salesman problems. Swarm Evol Comput. 2019;52:100607.

[41] Yang XS. Firefly algorithms for multimodal optimization. In: Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications; 2009. p. 169–78.

[42] Zhou L, Ding L, Qiang X. A multi-population discrete firefly algorithm to solve TSP. In: Pan L, Păun G, Pérez-Jiménez MJ, Song T, editors. Bio-inspired Computing – Theories and Applications. Berlin, Heidelberg: Springer; 2014. p. 648–53.

[43] Pitakaso R, Sethanan K, Jamrus T. Hybrid PSO and ALNS algorithm for software and mobile application for transportation in ice manufacturing industry 3.5. Comput Ind Eng. 2020;144:106461.

[44] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61.