ISSN 1854-6250

APEM journal

Advances in Production Engineering & Management

Volume 20 | Number 3 | September 2025

University of Maribor

Published by CPE
apem-journal.org

Advances in Production Engineering & Management

Identification Statement

APEM journal

ISSN 1854-6250 | Abbreviated key title: **Adv produc engineer manag** | Start year: 2006
ISSN 1855-6531 (on-line)

Published quarterly by **Chair of Production Engineering (CPE)**, **University of Maribor**
Smetanova ulica 17, SI – 2000 Maribor, Slovenia, European Union (EU)

Phone: 00386 2 2207522, Fax: 00386 2 2207990
Language of text: English

University of Maribor

APEM homepage: apem-journal.org
University homepage: www.um.si

APEM Editorial

OPEN ACCESS

Subsidizer: The journal is subsidized by Slovenian Research and Innovation Agency

Published by CPE, University of Maribor.

Advances in Production Engineering & Management is indexed and abstracted in the **WEB OF SCIENCE** (maintained by **Clarivate**): **Science Citation Index Expanded**, **Journal Citation Reports** – Science Edition, **Current Contents** – Engineering, Computing and Technology • **Scopus** (maintained by **Elsevier**) • **EBSCO**: Academic Search Alumni Edition, Academic Search Complete, Academic Search Elite, Academic Search Premier, Engineering Source, Sales & Marketing Source, TOC Premier • **ProQuest**: CSA Engineering Research Database – Cambridge Scientific Abstracts, Materials Business File, Materials Research Database, Mechanical & Transportation Engineering Abstracts, ProQuest SciTech Collection • **TEMA** (**DOMA**) • The journal is listed in **Ulrich's** Periodicals Directory and **Cabell's** Directory

University of Maribor
Chair of Production Engineering (CPE)

Advances in Production Engineering & Management

Volume 20 | Number 3 | September 2025 | pp 295–418

Contents

Journal homepage: apem-journal.org

ISSN 1854-6250 (print)
ISSN 1855-6531 (on-line)

Published by CPE, University of Maribor.

Scope and topics

Advances in Production Engineering & Management (*APEM journal*) is an interdisciplinary refereed international academic journal published quarterly by the *Chair of Production Engineering* at the *University of Maribor*. The main goal of the *APEM journal* is to present original, high quality, theoretical and application-oriented research developments in all areas of production engineering and production management to a broad audience of academics and practitioners. In order to bridge the gap between theory and practice, applications based on advanced theory and case studies are particularly welcome. For theoretical papers, their originality and research contributions are the main factors in the evaluation process. General approaches, formalisms, algorithms or techniques should be illustrated with significant applications that demonstrate their applicability to real-world problems. Although the *APEM journal* main goal is to publish original research papers, review articles and professional papers are occasionally published.

Fields of interest include, but are not limited to:

Additive Manufacturing Processes
Advanced Production Technologies
Artificial Intelligence in Production
Assembly Systems
Automation
Big Data in Production
Block Chain in Manufacturing
Computer-Integrated Manufacturing
Cutting and Forming Processes
Decision Support Systems
Deep Learning in Manufacturing
Discrete Systems and Methodology
e-Manufacturing
Evolutionary Computation in Production
Fuzzy Systems
Human Factor Engineering, Ergonomics
Industrial Engineering
Industrial Processes
Industrial Robotics
Intelligent Manufacturing Systems
Joining Processes
Knowledge Management
Logistics in Production
Machine Learning in Production
Machine-to-Machine Economy
Machine Tools
Machining Systems
Manufacturing Systems
Materials Science, Multidisciplinary
Mechanical Engineering
Mechatronics
Metrology in Production
Modelling and Simulation
Numerical Techniques
Operations Research
Operations Planning, Scheduling and Control
Optimisation Techniques
Project Management
Quality Management
Risk and Uncertainty
Self-Organizing Systems
Smart Manufacturing
Statistical Methods
Supply Chain Management
Virtual Reality in Production

APEM journal
Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 299–308
https://doi.org/10.14743/apem2025.3.541
ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Improving AGV path planning efficiency using Genetic Algorithms with Hamming distance-based initialization

Breznikar, Ž.[a], Gotlih, J.[a], Artič, Ž.[b], Brezocnik, M.[a]

[a]University of Maribor, Faculty of Mechanical Engineering, Maribor, Slovenia
[b]University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

ABSTRACT

This paper presents a Genetic Algorithm (GA) framework for warehouse navigation as a Travelling Salesman Problem (TSP) variant for Automated Guided Vehicles (AGVs). The warehouse layout is represented as a graph, where pick-up locations serve as terminal nodes. A distance matrix, computed via Breadth-First Search (BFS) enables efficient route evaluation. To promote diversity in the initial population, a Hamming distance-based vectorized initialization strategy is employed, ensuring that the chromosomes are maximally distinct. The GA balances exploration and exploitation by dynamically adjusting the fitness function. Early generations emphasize diversity, while later ones focus on solution refinement, improving convergence and avoiding premature stagnation. Our key contribution demonstrates that the Hamming distance-based approach achieves comparable or better results with significantly fewer chromosomes. This reduces computational cost and runtime, making the method well-suited for real-time AGV routing in warehouses. The framework is adaptable to structured environments and shows strong potential for integration into real-world logistics and robotics applications. Future work will focus on optimizing the algorithm and integrating it into the ROS 2 environment. The simplified version of the algorithm can be accessed at: *https://github.com/IntoTheVoid-61/Warehouse-Pathfinder*.

ARTICLE INFO

Keywords:
Automated guided vehicles (AGV);
Warehouse routing;
Genetic algorithms (GA);
Combinatorial optimization;
Hamming distance initialization;
Robot operating system 2 (ROS 2)

**Corresponding author:*
ziga.breznikar@student.um.si
(Breznikar, Ž.)

Article history:
Received 6 July 2025
Revised 24 October 2025
Accepted 27 October 2025

1. Introduction

The rapid expansion of automated manufacturing has brought increasing demand for intelligent logistics and autonomous systems in warehouses. Among these, Automated Guided Vehicles (AGVs) play a crucial role in transporting goods efficiently within complex environments. A considerable body of research has investigated the application of AGV systems across a wide range of environments and operational contexts, supported by diverse methodological approaches. In recent years, studies have increasingly adopted metaheuristic techniques to enhance system performance and optimization [1-12]. As warehouse layouts become more intricate and dynamic, determining optimal routes for AGVs becomes a critical challenge, essential for reducing delivery times and operational costs.

Genetic Algorithms (GAs), inspired by the principles of natural selection and evolution, are well suited for solving complex optimization problems such as AGV routing. By evolving population of candidate solutions through operations, inspired by biological evolution, GAs can efficiently explore large solution spaces and efficiently convergence towards near-optimal solutions even in the presence of multiple constraints and non-linearities. Their flexibility and robustness

have made them a popular choice for various combinatorial problems, including the well-known Travelling Salesman Problem (TSP) [13, 14], which closely parallels AGV routing in warehouses.

Traditional or uninformed search strategies, such as brute-force or blind random search, quickly become impractical as the scale of the problem increases [15]. In high-dimensional warehouse environments, the combinatorial explosion of possible routes leads to significant computational costs and sub-optimal outcomes. This underscores the need for heuristic or metaheuristic approaches that can guide the search process intelligently.

In this study, we propose a GA-based approach for warehouse routing, tailored for a top-down two-dimensional warehouse representation. Our algorithm encodes AGV routes as sequence of terminal locations and evolves these sequences to minimize travel distance. A central focus of this work is the effect of initial population diversity on GA performance. Specifically, we compare two initialization strategies: the conventional random initialization and a Hamming distance-based approach [16] explicitly designed to maximize population diversity, thereby enhancing exploratory capabilities in the early stages of the GA.

To evaluate the efficiency of the proposed method, extensive experiments were conducted using both initialization strategies under controlled settings. Results demonstrate that Hamming distance-based initialization performs comparably or better than standard approach, even when using significantly fewer individuals per generation. This indicates that strategic population design can reduce computational costs while maintaining solution quality-a critical insight for real-time AGV routing applications in operational warehouse systems.

The remainder of this paper presents the implementation of our GA-based routing algorithm, the design of the Hamming-based initialization, and detailed statistical analysis of the results obtained through multiple experimental runs.

2. Related work

The Travelling Salesman Problem (TSP) is a classic benchmark in combinatorial optimization and has been extensively addressed using GAs due to their ability to explore large, complex search spaces and avoid local optima through evolutionary operators. Standard GA implementations typically start with randomly generated populations, which help avoid early convergence but still frequently lead to premature stagnation in complex or highly constrained problems.

In the context of warehouse logistics, the TSP is frequently adapted to model route optimization for AGVs, where efficient sequencing of pick-up and delivery tasks is critical. Several works have proposed heuristic and metaheuristic-based solutions, including Ant Colony Optimization (ACO) and GA-based frameworks [13]. However, many approaches assume idealized conditions or rely on brute-force exploration, which becomes computationally expensive as the number of tasks and constraints grow. As a general approach, the authors in [17] proposed an enhanced GA for the TSP, in which the population is initialized using the Iterative Approximate Method, significantly improving solution efficiency and convergence speed.

Recent advancements have focused on enhancing diversity within GA population to improve exploration and convergence stability. For example, some authors have introduced adaptive mutations rates or hybridized GAs with local search techniques to maintain population diversity. Hamming distance, a measure of dissimilarity between binary strings, has been proposed for initializing populations that are maximally distinct [18]. While it has been applied in other domains, its use in structured, graph-based environments like warehouse routing remains relatively underexplored.

To our knowledge, few studies have empirically compared Hamming distance-based initialization with traditional random initialization in the context of AGV routing on warehouse graphs, employing statistically rigorous analysis to evaluate performance and convergence behaviour. This paper contributes to the existing literature by integrating a Hamming distance-based population initialization within a GA specifically designed for AGV navigation. By fostering high initial diversity, the algorithm reduces reliance on large population sizes, thus enhancing computational efficiency without sacrificing solution quality.

3. Environmental modelling and preprocessing

3.1 Matrix-based environment encoding

The warehouse environment is defined by three structural parameters:

- number of aisles,
- number of storage locations per aisle,
- number of storage blocks.

These parameters provide a compact and flexible description of the warehouse layout, which is essential for scalable simulations and graph-based modelling.

To facilitate algorithmic processing, the physical warehouse is first abstracted into a two-dimensional matrix. Each cell in the matrix corresponds to a discrete warehouse location and is assigned an integer label representing its functional role. Fig. 1 illustrates the top-down view of the warehouse layout and its matrix representation.

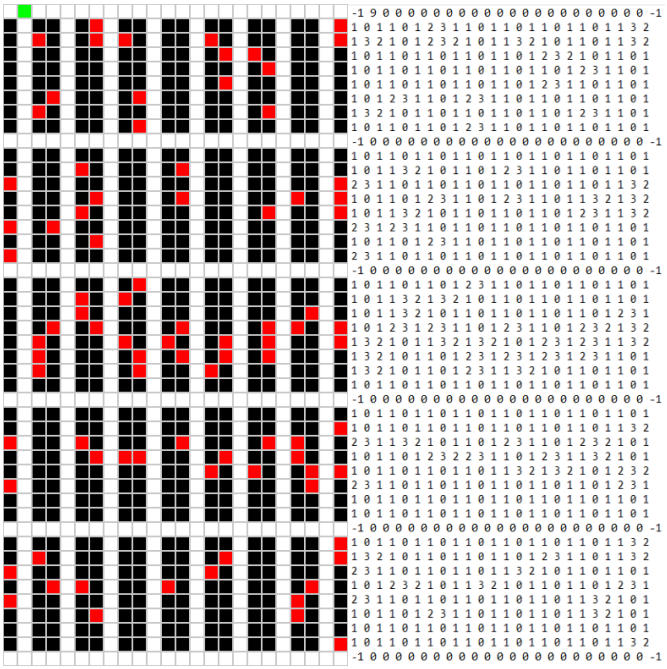


Fig. 1 Top-down view of the warehouse layout and its matrix representation

Table 1 Classification of colours and numerical values from Fig. 1

Value	Colour code	Description
0	White	Free space: Traversable
1	Black	Empty storage location: Untraversable
2	Red	Storage location: Untraversable
3	White	Pickup location: Traversable
9	Green	Start and end point: Traversable

Table 1 summarises the classification scheme used to assign numerical values to different elements of the warehouse.

Only the cells labelled 0, 3 and 9 are traversable. Among these, all nodes labelled as 3 represent mandatory pickup locations that the AGV must visit at least once. This matrix formulation allows a direct transformation into a graph, where each traversable cell becomes a node and edges represent valid moves between adjacent cells.

3.2 Graph-based preprocessing

To enable efficient path planning and distance evaluation, the matrix representation of the warehouse is transformed into a weighted graph [19]. In this graph, each traversable matrix cell (labelled as 0, 3 or 9) becomes a node. An undirected edge is created between every pair of adja-

cent traversable nodes. By default, edges are assigned a weight of 1, representing uniform movement cost. However, in real-world applications, factors such as bottlenecks, blocked or narrow passages, and other environmental constraints may increase the traversal difficulty. Such conditions are modelled by assigning higher weights to the affected edges.

Once the graph is constructed, all terminal nodes, marked with the label 3 in the matrix, are identified. These represent the pickup locations that the AGV must visit at least once. The set of terminals serves as the basis for solving a TSP-like optimization task.

To quantify distances between terminals, a distance matrix is computed using a BFS algorithm [20]. For each terminal node, BFS calculates the shortest path (in terms of total edge weight) to every other terminal. The resulting distance matrix is a symmetric square matrix where each element $D_{i,j}$ represents the shortest traversable distance between terminals i and j.

The distance matrix serves as a critical input to the optimization algorithm. It allows for rapid evaluation of the total route length of any candidate solution, without requiring real-time path-finding through the graph [21]. This preprocessing step thus transforms the original navigation problem into a purely combinatorial optimization task, significantly reducing computational overhead during the evolutionary search.

4. Genetic algorithm optimization

4.1 Chromosome representation and problem complexity

To apply a GA to the problem of AGV routing within a warehouse, we first define how a potential solution (chromosome) is represented. Each chromosome encodes a specific sequence in which the autonomous vehicle should visit all required terminal nodes, that is, the pickup locations identified during preprocessing.

A chromosome consists of a permutation of all terminal nodes, where each gene represents a single terminal node, and the order of genes determines the traversal path of the AGV. This is represented in Fig. 2, which illustrates 5 different potential solutions to the routing problem, given 8 pickup locations.

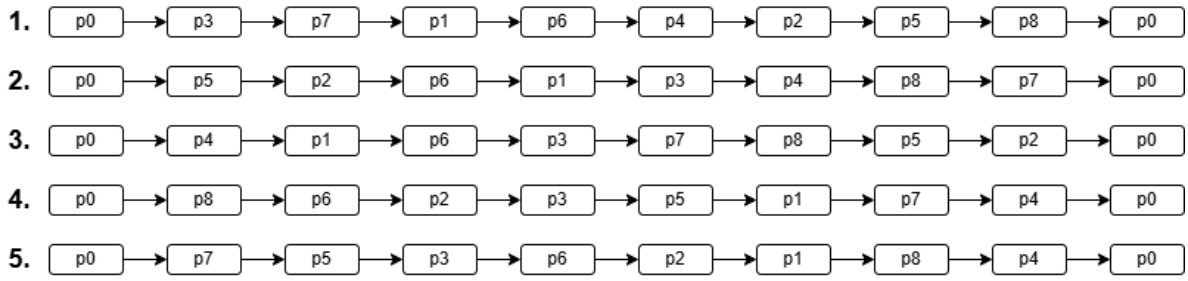


Fig. 2 Example of 5 different solutions to the routing problem with 8 terminal nodes.

This problem closely resembles the mentioned TSP, a well-known combinatorial optimization problem. For n terminal nodes, there are $n!$ possible permutations, making exhaustive search methods computationally infeasible, even for relatively small values of n. As the problem scales, brute-force methods become impractical due to factorial growth in complexity.

Given a set of terminals $T = \{p_1, p_2, \dots, p_n\}$, the objective is to find a permutation π, as shown in Eq. 1, that minimizes the total travel distance.

$$\pi = \sum_{i=1}^{n} D_{\pi(i),\pi(i+1)} \tag{1}$$

Since brute-force methods become computationally infeasible for large n, a GA is employed to address this challenge. GAs are well-suited for permutation-based combinatorial problems and enable efficient exploration of large solution spaces by evolving a population of chromosomes through genetic operations.

4.2 Hamming distance-based population initialization

A critical step in ensuring the effectiveness of a GA is the initialization of a diverse population. Diversity promotes broad exploration of the solution space in early generations and helps avoid premature convergence to local optima. To systematically promote diversity, we employed a strategy based on the Hamming distance.

For two sequences of equal length, the Hamming distance is defined as the number of positions at which the corresponding elements differ [22]. In our context, each chromosome is a permutation of terminal nodes, and the Hamming distance between two chromosomes indicates the number of differing terminal positions in the visitation sequence.

The population is initialized by iteratively generating random permutations and comparing them against the already selected chromosomes. A candidate chromosome is accepted into the population only if a minimum Hamming distance from the set of existing chromosomes exceeds a decreasing threshold, starting from the maximum value n (the number of terminals). This ensures that the initial population is highly diverse, therefore promoting broad exploration of the solution space. If, after a predefined number of attempts, no chromosome meets the current Hamming threshold, the threshold is reduced by one and the process repeats. This adaptive mechanism balances diversity with feasibility. A high-level implementation of the described method is shown in Fig. 3.

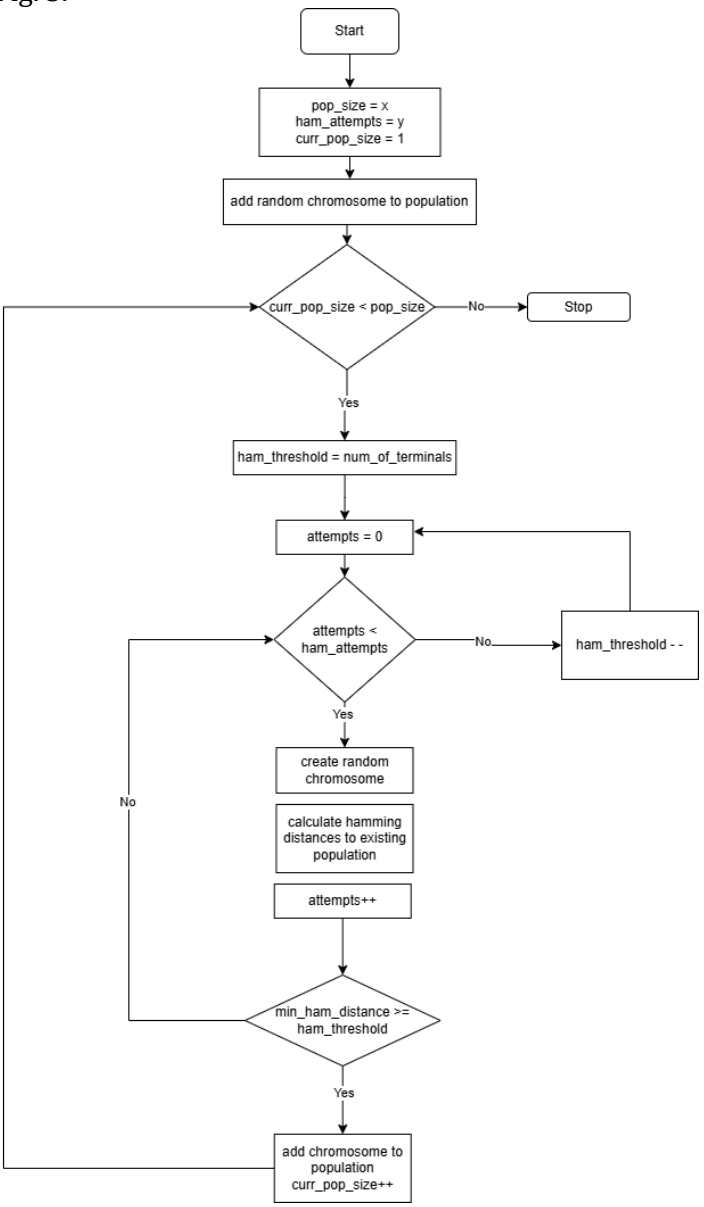


Fig. 3 Flowchart of a high-level hamming distance-based algorithm to ensure initial population diversity.

This Hamming distance-based algorithm ensures diversity in the starting population, enabling a broader and more efficient exploration of the solution space. The computational complexity of the initialization process increases approximately as $O(P^2 A\, n)$, where P denotes the population size, n the number of pickup locations, and A the number of random Hamming distance attempts per chromosome.

4.3 Evolutionary dynamics and evaluation

This section describes the key components of the GA, including selection mechanisms, elitism strategy, mutation operators, and the dynamic fitness evaluation. These mechanisms were designed to balance exploration and exploitation throughout the optimization process.

A dynamic tournament selection method is used to select chromosomes for reproduction or crossover. In each tournament, a subset of k chromosomes from the current population is randomly selected, and the best-performing individual is chosen. To balance exploration and exploitation over time, the tournament size is dynamically adjusted, as shown in Eq. 2.

$$k = round((k_{max} - k_{min}) \cdot progress + k_{min}) \tag{2}$$

where $progress$ is defined in Eq. 3.

$$progress = \frac{current\_generation}{max\_generation} \tag{3}$$

This allows smaller tournament sizes early on (encouraging diversity) and larger tournaments later (favouring selection pressure). Following the selection, each chromosome is either reproduced directly into the next generation with probability p, or undergoes crossover with a different chromosome with probability $1 - p$.

To preserve high quality solutions, elitism is applied by copying the top i chromosomes unaltered into the next generation. The number of elite chromosomes increases quadratically with the progress of generations as shown in Eq. 4.

$$i = round((i_{max} - i_{min}) \cdot progress^2 + i_{min}) \tag{4}$$

This strategy ensures that more optimal solutions are retained as the algorithm converges.

Two types of mutation operators are employed to maintain genetic diversity:

- Single-gene swap: Two genes (positions) in the chromosome are randomly swapped [23].
- 2-opt-swap: A sub-sequence of genes is reversed, a common local optimization technique in TSP-like problems [24].

The mutation probability is dynamic and is sampled from a uniform distribution *U(a,b)*, where bounds *a* and *b* evolve as generations progress, as shown in Eq. 5 and Eq. 6.

$$a = (a_{end} - a_{start}) \cdot progress + a_{start}, \text{ where } a_{start} > a_{end} \tag{5}$$

$$b = (b_{end} - b_{start}) \cdot progress + b_{start}, \text{ where } b_{start} > b_{end} \tag{6}$$

This allows higher variability early on (favouring exploration), which gradually decreases as the algorithm approaches convergence.

The fitness function transitions from the exploration phase to the exploitation phase to balance global search and solution refinement. During the exploration phase, the fitness incorporates both solution quality and population diversity, as shown in Eq. 7.

$$f = \alpha \cdot normalized\_path\_length - (1 - \alpha) \cdot normalized\_hamming\_distance \tag{7}$$

Normalization is necessary to account for differing numerical scales. Path lengths are normalized relative to population statistics, while Hamming distances are normalized relative to the maximum possible value, corresponding to the chromosome length. The coefficient α balances the two terms and is defined by a sigmoid function [25] as shown in Eq. 8.

$$\alpha = \frac{1}{1 + e^{-x}} \tag{8}$$

where x is defined as shown in Eq. 9.

$$x = 20 \cdot \frac{generation\_counter}{exploration\_end} - 10 \tag{9}$$

In Eq. 9, *generation_counter* is an integer value representing the current generation number, while *exploration_end* is the predefined generation at which exploitation begins. During the exploitation phase, fitness is based solely on the provided path length of a chromosome. This phase focuses entirely on improving the solution quality.

A high-level schematic of the proposed GA framework is illustrated in Fig. 4.

Fig. 4 Algorithmic structure of the GA-based navigation framework

5. Results and discussion

This section presents a comprehensive evaluation of the impact of Hamming distance-based population initialization (H) compared to random initialization (NH) in a GA designed for optimizing warehouse routing. The analysis spans four key performance metrics:

- runtime duration,
- initial population diversity,
- convergence speed,
- final solution quality (best tour length).

The experiments were performed over 20 independent runs per setting, using identical seeds and warehouse environments to ensure consistency.

5.1 Equal population size comparison

In the first series of experiments, both methods were executed under identical setting with 30 individuals per generation. Each configuration was repeated 20 times using the same seeds and warehouse environment. To validate the use of parametric tests, Shapiro-Wilk and D'Agostino-Pearson normality tests were conducted for all metrics [26]. Results indicated normal distribution across all groups. Table 2 summarizes the results.

Table 2 Statistical analysis of GA metrics: Hamming vs. Non-Hamming

Metric	Hamming (Mean ± SD)	Non-Hamming (Mean ± SD)	*p*-value	Significance
Runtime (s)	45.0283 ± 5.3	40.2011 ± 5.79	0.0057	significant
Initial population Hamming distance	80.3567 ± 0.03	78.2347 ± 0.05	< 0.0001	significant
Generations	6122.05 ± 692.09	5907.45 ± 739.28	0.2904	not significant
Best tour length	400.8000 ±11.19	403.6000 ± 9.46	0.3915	not significant

Key takeaways:

- Population diversity: The Hamming-initialized population achieved significantly higher diversity, confirming the effectiveness.
- Runtime: Hamming initialization introduced a slight computational overhead due to pairwise distance calculations.
- Final Quality & Convergence: No statistically significant improvement was observed.

5.2 Exploring population size effects

Additional experiments were conducted to investigate whether Hamming distance-based initialization can effectively compensate for a reduced population size in GA applications. This evaluation was carried out on a structurally distinct warehouse layout to ensure generalizability of the findings. In this setting, we compared the performance of a GA configured with only 30 individuals initialized using the Hamming distance strategy (H30) against a GA employing 150 individuals initialized randomly (NH150). Despite the fivefold disparity in population size, the H30 configuration consistently demonstrated comparable solution quality to the NH150 setup in both final tour length and convergence behaviour.

Statistical analysis of the tour length distributions supported this finding, yielding a significant *p*-value [27] ($p < 0.05$) indicating that the performance difference favoured the H30 configuration. These results underscore the effectiveness of diversity-promoting strategies in evolutionary algorithms. Instead of compensating for insufficient diversity by brute-force scaling of the population size, initializing the population with maximally dissimilar chromosomes enables more efficient exploration of the solution space. This provides a strong argument for adopting informed initialization techniques, especially in resource-constrained environments where computational efficiency is critical, such as real-time AGV routing in dynamic warehouse settings.

6. Conclusion

This work explored the application of a GA to solve warehouse routing problem, a task critical to the efficiency of AGV system in logistics and manufacturing environments. Our results confirm that a GA-based approach is not only viable but effective for generating high quality routing solutions within reasonable computational budget. The simplified version of the algorithm can be accessed at: *https://github.com/IntoTheVoid-61/Warehouse-Pathfinder*. Future work of the algorithm will extend to its optimization and integration into the ROS2 environment.

In a targeted comparison, a GA with just 30 chromosomes initialized using Hamming-based method achieve statistically comparable results in tour quality relative to that of a GA with 150 randomly initialized individuals. This outcome highlights the central contribution of our study:

intelligent population seeding can significantly reduce the required population size without compromising performance.

This finding has tangible implications for real-world deployment. Reducing population size lowers computational time and memory usage, making approach more suited for embedded or real-time systems commonly used in AGV applications. Accordingly, this study provides a practical and scalable GA-based framework for warehouse routing, with added benefit of an initialization method that enhances performance under constrained resources.

Future work should focus on deploying the proposed system on real-world AGV platforms and evaluating its performance through experiments and field-testing. While the algorithm demonstrates strong results in two-dimensional navigation (based on top-down view of the warehouse), it does not yet account for three-dimensional considerations. Furthermore, as this study was conducted on an abstracted AGV model, specific characteristics of actual vehicle were not incorporated. Notably, the model assumed that AGV could transport an unlimited mass of cargo and did not require a return to the starting point for unloading. Future implementations on embedded system should therefore account for such practical constraints, including limited payload capacity and the need for cargo drop-off behaviour.

References

[1] Löffler, M., Boysen, N., Schneider, M. (2021). Picker routing in AGV-assisted order picking systems*, INFORMS Journal on Computing*, Vol. 34, No. 1, 440-462, doi: 10.1287/ijoc.2021.1060.

[2] Agatz, N., Bouman, P., Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone*, Transportation Science,* Vol. 52, No. 4, 965-981, doi: 10.1287/trsc.2017.0791.

[3] Han, Z., Wang, D., Liu, F., Zhao, Z. (2017). Multi-AGV path planning with double-path constraints by using an improved genetic algorithm*, Plos One,* Vol. 12, No. 7, Article Id. e0181747, doi: 10.1371/journal.pone.0181747.

[4] Simon, J., Trojanová, M., Hošovský, A., Sárosi, J. (2021), Neural network driven automated guided vehicle platform development for Industry 4.0 environment, *Tehnički Vjesnik – Technical Gazette,* Vol. 28, No. 6, 1936-1942, doi: 10.17559/TV-20200727095821.

[5] Bostancı, B., Karaağaç, A. (2019). Investigating the shortest survey route in a GNSS traverse network, *Tehnički Vjesnik – Technical Gazette,* Vol. 26, No. 2, 355-362, doi: 10.17559/TV-20170924174221.

[6] Xu, L., Wang, Y., Liu, L., Wang, J. (2016). Exact and heuristic algorithms for routing AGV on path with precedence constraints*, Mathematical Problems in Engineering*, Vol. 2016, No. 1, Article No. 5040513, doi: 10.1155/2016/5040513.

[7] Wang, D.L., Ding, A., Chen, G.L., Zhang, L. (2023). A combined genetic algorithm and A* search algorithm for the electric vehicle routing problem with time windows, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 403-416, doi: 10.14743/apem2023.4.481.

[8] Zhang, H.Q. (2024). Optimizing obstacle avoidance path planning for intelligent mobile robots in multi-obstacle environments, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 358-370, doi: 10.14743/apem2024.3.512.

[9] Boccia, M., Masone, A., Sterle, C., Murino, T. (2023). The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach, *European Journal of Operational Research*, Vol. 307, No. 2, 590-603, doi: 10.1016/j.ejor.2022.10.023.

[10] Micieta, B., Durica, L., Binasova, V. (2018). New solution of abstract architecture for control and coordination decentralized systems, *Tehnički Vjesnik – Technical Gazette,* Vol. 25, Supplement 1, 135-143, doi: 10.17559/TV-20160117100949.

[11] Liu, M.L., Yao, X.Z., Huang, J.Y., Zhang, C. (2022). Optimization of unmanned vehicle scheduling and order allocation, *International Journal of Simulation Modelling*, Vol. 21, No. 3, 477-488, doi: 10.2507/IJSIMM21-3-613.

[12] Wang, S.R., Huang, Q. (2022). A hybrid code genetic algorithm for VRP in public-private emergency collaborations, *International Journal of Simulation Modelling*, Vol. 21, No. 1, 124-135, doi: 10.2507/IJSIMM21-1-595.

[13] Wang, Y.D., Lu, X.C., Shen, J.R. (2021). Improved Genetic Algorithm (VNS-GA) using polar coordinate classification for workload balanced multiple Traveling Salesman Problem (mTSP), *Advances in Production Engineering & Management,* Vol. 16, No. 2, 173-184, doi: 10.14743/apem2021.2.392.

[14] Sui, J., Ding, S., Huang, X., Yu, Y., Liu, R., Xia, B., Ding, Z., Xu, L., Zhang, H., Yu, C., Bu, D. (2025). A survey on deep learning-based algorithms for the traveling salesman problem, *Frontiers of Computer Science,* Vol. 19, Article No. 196322, doi: 10.1007/s11704-024-40490-y.

[15] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2009). *Introduction to algorithms*, Third edition, MIT Press, Cambridge, Massachusetts, USA.

[16] Leung, Y., Gao, Y., Xu, Z.-B. (1997). Degree of population diversity - a perspective on premature convergence in genetic algorithms and its Markov chain analysis, *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, 1165-1176, doi: 10.1109/72.623217.

[17] Alkafaween, E., Hassanat, A., Essa, E., Elmougy, S. (2024). An efficiency boost for genetic algorithms: Initializing the GA with the iterative approximate method for optimizing the Traveling Salesman Problem – Experimental insights, *Applied Sciences,* Vol. 14, No. 8, Article No. 3151, doi: 10.3390/app14083151.
[18] Cruz Chávez, M.A., Martínez-Oropeza, A. (2016). Feasible initial population with genetic diversity for a population-based algorithm applied to the vehicle routing problem with time windows, *Mathematical Problems in Engineering,* Vol. 2016, No. 1, Article No. 3851520, doi: 10.1155/2016/3851520.
[19] Diestel, R. (2017). *Graph theory*, 5th edition, Springer, Berlin, Germany, doi: 10.1007/978-3-662-53622-3.
[20] Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2014). *Data structures & algorithms in Java*, 6th edition, Wiley, Hoboken, New Jersey, USA.
[21] Deza, M.M., Deza, E. (2013). *Encyclopedia of distances,* 2nd edition, Springer, Berlin, Germany, doi: 10.1007/978-3-642-30958-8.
[22] Black, P.E. (2006). Hamming distance, In: Black, P.E. (ed.), *Dictionary of algorithms and data structures* [online], Black, P.E. (ed.), from *https://www.nist.gov/dads/HTML/HammingDistance.html*, accessed May 13, 2025.
[23] Sivanandam, S., Deepa, S. (2008). Genetic algorithms, In: Sivanandam, S., Deepa, S. (eds.), *Introduction to genetic algorithms,* Springer, Berlin, Germany, 15-37, doi: 10.1007/978-3-540-73190-0_2.
[24] Uddin, F., Riaz, N., Manan, A., Mahmood, I., Song, O.-Y., Malik, A.J., Abbasi, A.A. (2023). An improvement to the 2-Opt heuristic algorithm for approximation of optimal TSP tour, *Applied Sciences*, Vol. 13, No. 12, Article No. 7339, doi: 10.3390/app13127339.
[25] Science Direct. Sigmoid Function, from *https://www.sciencedirect.com/topics/computer-science/sigmoid-function*, accessed April 4, 2025.
[26] Altman, D.G., Bland, J.M. (1995). Statistics notes: The normal distribution, *The BMJ*, Vol. 310, Article No. 298, doi: 10.1136/bmj.310.6975.298.
[27] Gao, J. (2020). P-values – a chronic conundrum, *BMC Medical Research Methodology*, Vol. 20, Article No. 167, doi: 10.1186/s12874-020-01051-6.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 309–324
https://doi.org/10.14743/apem2025.3.542

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Golden Drop Algorithm: A Water Wave and Golden Ratio inspired novel metaheuristic for global optimization

Karaçizmeli, İ.H.[a,*]

[a]Industrial Engineering Department, Harran University, Şanlıurfa, Türkiye

ABSTRACT

Metaheuristic methods are approximate solution techniques that are widely used to solve complex optimization problems. Drawing inspiration from nature, physics, or human behavior, researchers have developed a diverse range of metaheuristic algorithms, either population-based or single-solution-based, with varying neighborhood structures and memory usage. Due to the diversity and complexity of optimization problems in different domains, no single algorithm can be universally effective for all types of problems. This study proposes a novel metaheuristic algorithm, the Golden Drop Algorithm (GDA), which is simple, efficient, and easy to implement. GDA is a physics-inspired method that simulates the propagation of circular waves formed by raindrops falling into water, with wave expansion governed by the golden ratio. The algorithm was evaluated using the Congress on Evolutionary Computation (CEC) 2017 benchmark functions and was compared with three well-known metaheuristics: Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Particle Swarm Optimization (PSO). The results demonstrate that GDA is not only straightforward and computationally efficient, but also capable of producing competitive and feasible solutions. In 50 % of the CEC 2017 benchmark functions, GDA ranked first, while in 32 % of the functions, it secured second place. In addition, its performance was validated through three benchmark engineering design problems: welded beam design, pressure vessel design, and compression spring design, highlighting its robustness and effectiveness in solving constrained nonlinear optimization tasks.

ARTICLE INFO

Keywords:
Metaheuristic optimization;
Physics-inspired metaheuristics;
Water wave;
Golden ratio;
CEC 2017 benchmark functions;
Engineering design problems

**Corresponding author:*
hkaracizmeli@harran.edu.tr
(Karaçizmeli, İ.H.)

Article history:
Received 6 August 2025
Revised 8 September 2025
Accepted 15 September 2025

1. Introduction

Optimization is a process of solving a problem under certain constraints and with a specific objective in mind [1]. It is a major topic of investigation in applied mathematics [2]. Optimization research is carried out not only to broaden theoretical perspectives but also to solve real-life problems in diverse disciplines ranging from economics to finance, engineering to management, and statistics to computer science [3, 4].

The methods used to solve optimization problems are divided into two main categories: exact and approximate solution methods [5]. The more complex the problems, the more difficult it becomes to produce solutions by using exact solution methods. However, with approximate methods, near-optimum results can be obtained in a shorter time. Approximate methods can be divided into two groups: problem-specific heuristics and metaheuristics that can be applied to diverse problems [5]. The number of studies on metaheuristics has increased exponentially, especially with the development of computer technologies [6].

Using different perspectives, researchers have classified metaheuristics in different ways. There are classifications based on the inspiration source of the algorithm, the neighborhood

structure, the memory status, whether the search is population-based or single-point [7]. In another classification, algorithms are divided into four categories according to their behavior: evolution, swarm intelligence, physics, and human behavior-based [8, 9].

This study proposes a novel nature-inspired metaheuristic optimization algorithm. The proposed algorithm is population-based, employs a binary neighborhood structure—where each candidate solution has two neighbors, and the search continues in the direction of the better neighbor—incorporates memory, and is based on physics. This method is called the GDA. The purpose of the present study was to develop a new algorithm that is simple and can easily be implemented. This algorithm is inspired by the propagation of circular waves formed by raindrops falling into water. The solution space is searched by the propagation of the waves. When searching for neighborhoods, waves move according to the golden ratio. The algorithm developed was tested on the CEC 2017 functions, which are known to be extremely challenging [2]. The results are compared with those produced by the GWO, GA, and PSO, which are popular algorithms. The data revealed that the GDA yielded successful and satisfactory results.

To assess its practical effectiveness, the proposed algorithm was applied to three benchmark engineering design problems: welded beam design, pressure vessel design, and compression spring design. For these problems, the widely recognized Differential Evolution (DE) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithms were also incorporated into the set of comparison methods. The comparative analysis against CMA-ES, DE, GWO, GA, and PSO demonstrated that GDA consistently delivered the best performance across all test instances, with the exception of a single case.

As far as is known, the term metaheuristics was coined by Glover in 1986 [10]. While the foundations of metaheuristic algorithms were initially laid after the 1950s, major developments took place in the 1980s [11]. For example, Holland presented the GA as an optimization tool in 1975 [12]. Kirckpatrick developed the simulated annealing algorithm in 1983 [13]. Glover proposed the tabu search algorithm in 1989 [14]. Kennedy presented the PSO in 1995 [15]. The ant colony optimization method, which Dorigo founded in 1991 [16], is presented as a metaheuristic method together with Di Caro in 1999 [17]. Karaboga developed the artificial bee colony algorithm in 2005 [18]. Yang developed the firefly algorithm in 2008 [19].

In addition to these popular algorithms, many other algorithms and their hybrid versions have been developed. Since metaheuristic algorithms are not intended to solve specific problems, they can be successfully used to address various types of problem. However, their success may vary depending on the type of problem itself. Therefore, besides doing work on existing algorithms, researchers have a strong desire to develop new algorithms.

The present study was inspired by water, one of the fundamental components of nature. The expansion and propagation of circular waves formed on the surface of water was the idea behind this algorithm. Therefore, previous studies on water and waves were investigated. The water cycle algorithm, for example, modeled tributaries that merge to form rivers and rivers that merge and flow into the sea. The sea is considered as the optimum or near-optimal result [20]. The intelligent water drop algorithm was developed by relating the amount of sand carried by multiple water drops from one point to another in a river [21, 22]. In another study, the way this algorithm progressed was modified and used [23]. In water wave optimization, better solutions were sought in solution space using different wavelengths [24]. The circular water waves algorithm is designed based on the propagation principle of circular waves produced by a water drop falling into water [25]. The solution starts with a single drop and twice as many neighborhoods as the problem size are computed in each iteration. This negatively affects the computational performance of the algorithm. The vortex search algorithm also resembles waves in terms of shape. However, this algorithm is inspired by the motion of vortices and is an algorithm based on a single solution [5].

Studies using the golden ratio for optimization purposes have started to diversify, particularly in the last five years. It was first introduced in the golden section search technique, developed by the American mathematician Jack Kiefer in 1953 [26]. In another technique introduced in another study, PSO was used in combination with the golden ratio [27]. An optimization algorithm was developed by combining the sine function with the golden ratio [28]. Inspired by the

golden ratio in the growth of plants and animals, Nematollahi *et al.* introduced the golden ratio optimization method (GROM) [29]. Various researchers have used the GROM or its hybrid versions in combination with other methods to solve some problems [30-35]. For example, in another study, a hybrid version of the teaching–learning-based optimization algorithm with the golden ratio was proposed [36].

An examination of the algorithms developed in recent years brings forth various algorithms, including the *water strider algorithm*, which takes into account the life cycle of the water strider (a type of insect) [37]; the *colony-based search algorithm*, which mimics the hunting strategies of flocks living in colonies [2]; and the *modified Archimedes optimization* algorithm, inspired by Archimedes' principle [4]. There are also other studies that we cannot mention here due to space constraints. Apparently, such studies have always put an effort to create a novel algorithm that is more efficient and more successful. Similarly, GDA was inspired by nature and developed as a relatively efficient, easy-to-understand, and easy-to-implement method.

2. Methodology

The GDA was inspired by circular waves created by raindrops falling on water. It is a population-based algorithm, and the movement of waves is in line with the golden ratio.

2.1 The golden ratio

The golden ratio is an irrational number defined as $(1+\sqrt{5})/2$ [38]. It is symbolized by the Greek numeral phi (Φ), and its value is approximately 1.618034. The golden ratio can also be explained on the Fibonacci spiral, as seen in Fig. 1a. The sum of two consecutive numbers in the Fibonacci sequence gives the next number. The ratios of the larger numbers to the smaller numbers throughout the sequence are very close to each other. From the 17th number onward, it is approximately equal to the value of Φ. The golden ratio has been used in many fields, from mathematics to physics and architecture to art. As far as known, the golden ratio was first mentioned in the book *De Divina Proportione* by the mathematician Luca Pacioli, illustrated by Leonardo da Vinci, and published in 1509 [38]. The golden ratio, also called the divine proportion by Pacioli, is thought to provide the best harmony among the parts of a whole and is found in nature (Fig. 1b) [39]. It is known that this ratio was used in ancient Sumerian tablets, pyramids built by the Egyptians, and the Parthenon design of the Greeks [40, 41].

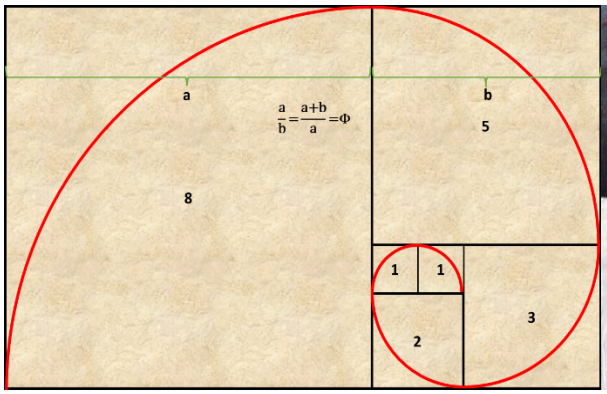


Fig. 1a Fibonacci spiral [39]

Fig. 1b the divine proportions from nature [39]

2.2 Drops and waves

The GDA starts with a population (X) that is composed of n raindrops, which are randomly generated at a given interval. Each drop represents a feasible solution and has d dimensions. In nature, drops falling into water form circular waves surrounding themselves (Fig. 2).

These waves are formed sequentially from the center outward. The distance between the waves is called wavelength (λ), which refers to the distance from the peak of the first wave to the drop in the center. This distance depends on the velocity of the wave (v). The λ and v do not change unless the frequency (f) changes. The v is equal to the ratio between the λ and the frequency of the wave (Eq. 1).

Fig. 2 Drops falling into water [42]

$$v = \frac{\lambda}{f} \tag{1}$$

In nature, waves are reflected when they hit reflective materials and damped when they hit non-reflective materials. In the GDA, the property of reflection in waves is disregarded, but the property of damping is used.

2.3 Motions of waves

As illustrated in the pseudocode (Algorithm 1), the algorithm begins by initializing the parameters d, h_k, h_{max}, n and the maximum number of iterations $Iter_{max}$. Here, h_{max} represents the maximum number of waves, while h_k denotes the number of waves designated for a precise search. These parameters, which influence the search velocity, should be determined according to the type of problem. Then the objective function values are computed for the randomly generated population X, and the resulting objective function value ($d_{bestval}$) and corresponding position vector ($d_{bestpos}$) are stored for each drop. The global best values are recorded as $g_{bestval}$ and $g_{bestpos}$. Subsequently, wave motions are initiated. The GDA algorithm incorporates four distinct types of wave motion, one of which is implemented as an optional mechanism.

The first wave type corresponds to the motion of standard circular waves propagating around the drops (Fig. 3a). If the frequency is assumed to be 1 in Eq. 1, then λ and v become equal. In other words, once v is calculated, λ is effectively determined as well. As shown in Eq. 2, the value of v is first computed by dividing the position value of the center drop by the golden ratio. Then the value of v is both added to and subtracted from the position value of the center drop (Eq. 3). Thus, the points w^+ in the positive direction and w^- in the negative direction. In this way, the first circular wave is formed. The distance traveled by the wave, that is, λ, changes by an amount proportional to the golden ratio at the wave's current position.

The objective function values are evaluated at these two points, and the better of the two is compared with $d_{bestval}$. If an improved value is found, it is treated as a new drop, and the second type of wave motion is initiated. Otherwise, successive waves are generated using the same v value. The same procedure is repeated for each new wave.

$$v_i = \frac{x_i}{\Phi} \tag{2}$$

$$w_i^-, w_i^+ = x_i \pm v_i \tag{3}$$

Algorithm 1 The pseudocode of GDA

```
Set input parameters d, h_k, h_max, n, Iter_max
Create the initial population
Compute the objective function
d_bestval = The best objective function value of the x_i
d_bestpos= The best position vector of the x_i
g_bestval = The best objective function value of the X
g_bestpos= The best position vector of the X
Compute v using Eq. 2
While (Iter < Iter_max)
    For i = 1:n
        If 1 < h < h_k
            Calculate v using Eq. 7
        Else if h_k ≤ h ≤ h_max
            v_i^iter = v_i^iter-1
        End if
        Update w_i+ and w_i- positions using Eq. 3
        Compute objective function values for w_i+ and w_i-
        Choose the better one as new objective function value Obj_new
        If h > h_max then
            Update the x_i using Eq. 5 and Eq. 6
            Compute objective function value Obj_new
            Compute v using Eq. 2
            h = 1
        End if
        If Obj_new < d_bestval then
            Update the d_bestval and d_bestpos vector
            Compute the v using Eq. 4
        Else
            h = h + 1
        End if
        If d_bestval < g_bestval
            Update the g_bestval and g_bestpos vector
        End if
    End for
Iter = Iter + 1
End while
```

The second type of wave motion treats the newly identified drop as the new center point (Fig. 3b). A circular wave is then generated around this point; however, the corresponding v is smaller than that of the previous wave, as the golden ratio is applied to the distance between the previous drop and the current one. The value of v is computed using Eq. 4, which enables the GDA to perform more refined searches, thereby enhancing its local search capabilities. A new wave is generated by substituting the computed v value into Eq. 3.

$$v_i^{iter} = \frac{v_i^{iter-1}}{\Phi} \tag{4}$$

The third type of wave motion is triggered following a damping condition (Fig. 3c). If the objective function does not improve after the generation of h_{max} consecutive waves, the current wave is considered damped, and a new drop falls into the water. The golden ratio of the distance between $g_{bestpos}$ and $d_{bestpos}$ is computed, as defined in Eq. 5, and this value is assigned as the new value of v. A new drop is then randomly generated within the interval $[g^-_{bestpos}, g^+_{bestpos}]$, using this newly computed v value Eq. 6. The objective function value for the new drop is subsequently evaluated. Thereafter, the value v for the new drop is computed using Eq. 2, and the wave generation procedure restarts.

$$v_i = \frac{g_{bestpos} - d_{bestpos}}{\Phi} \tag{5}$$

$$g^-_{bestpos}, g^+_{bestpos} = g_{bestpos} \pm v_i \tag{6}$$

The fourth wave type is an optional component of the algorithm, structurally similar to the first wave type (see Fig. 3a), but operating at lower velocities compared to the first and second wave types. This wave is particularly effective for conducting a more refined and precise search, complementing the algorithm's fast exploration capability by enhancing detailed local exploitation. The parameter p governs the velocity of this wave and is selected within the range $(0, \Phi - 1)$. Smaller values of p yield lower computed v values, resulting in the subsequent wave being generated closer to the current one. After computing v using Eq. 7, a new wave is produced via Eq. 3, and potential solution points are generated along this wave in both positive and negative directions.

$$v_i = x_i \times p \tag{7}$$

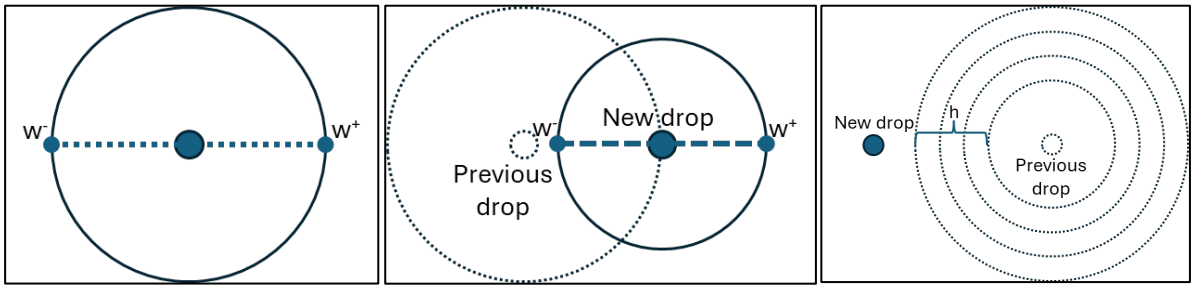


Fig. 3a Motion type 1 **Fig. 3b** Motion type 2 **Fig. 3c** Motion type 3

3. Experiments on CEC 2017 benchmark problems

The GDA was tested using the CEC 2017 which is currently composed of 28 functions [43]. Two of these functions are unimodal; seven are multimodal; ten are hybrid, and nine are composition. More information about these functions is provided in Awad *et al.* [44]. The 10-dimensional (10D), 30-dimensional (30D), and 50-dimensional (50D) versions of these functions were solved. As in CEC 2017, the maximum number of function evaluations (*MaxFES*) was accepted as the stopping criterion of the algorithms and its value was used as 10,000 X D. The search range was [-100, 100]. Each function was executed independently 20 times for the 10D case, and 50 times for both the 30D and 50D cases. The results obtained were compared with proven and popular algorithms, including GA, GWO and PSO [45].

The mean, standard deviation, and best values of the results are presented in Table 1 for 10D, Table 2 for 30D, and Table 3 for 50D, based on the functions and algorithms. As shown in Table 1, for the 10D case, the best results were obtained using GDA in 15 functions, GWO in 6 functions, and PSO and GA in 4 functions each. Among these, GDA was ranked first overall. Table 2 shows a similar trend. For 30D, the best results were achieved using GDA in 15 functions, GWO in 7 functions, PSO in 5 functions, and GA in 1 function. Once again, GDA ranked first overall, and for the functions where it did not achieve the best result, the difference from the best-performing algorithms was generally small. As presented in Table 3, for 50D, the best results were obtained with GDA in 12 functions, GWO in 10 functions, and PSO in 6 functions. In the 9 functions where GDA did not rank first, it secured second place.

Table 1 Results of 10D problems

	GDA			GA			GWO			PSO		
Fn	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std
1	2.52E+03	**1.08E+02**	2.51E+03	9.89E+06	5.12E+06	3.90E+06	7.36E+07	2.32E+04	1.33E+08	2.64E+03	3.05E+02	3.56E+03
3	3.40E+02	**3.00E+02**	6.43E+01	1.24E+04	3.68E+03	8.51E+03	2.61E+03	1.33E+03	1.72E+03	3.00E+02	**3.00E+02**	4.77E-14
4	4.22E+02	**4.00E+02**	3.09E+01	4.12E+02	4.10E+02	2.54E+00	4.10E+02	4.05E+02	9.51E+00	4.17E+02	4.00E+02	2.89E+01
5	5.38E+02	5.08E+02	1.55E+01	5.24E+02	5.20E+02	4.06E+00	5.11E+02	**5.07E+02**	4.39E+00	5.24E+02	5.10E+02	1.62E+01
6	6.12E+02	**6.00E+02**	7.26E+00	6.09E+02	6.06E+02	2.93E+00	6.01E+02	6.01E+02	4.04E-01	6.06E+02	6.01E+02	7.06E+00
7	7.44E+02	7.22E+02	1.78E+01	7.33E+02	7.24E+02	6.02E+00	7.24E+02	**7.17E+02**	1.04E+01	7.29E+02	7.23E+02	7.38E+00
8	8.22E+02	**8.07E+02**	9.26E+00	8.14E+02	8.11E+02	4.57E+00	8.14E+02	8.09E+02	3.76E+00	8.20E+02	8.13E+02	5.92E+00
9	1.04E+03	9.02E+02	1.27E+02	9.04E+02	9.02E+02	2.47E+00	9.10E+02	**9.00E+02**	1.33E+01	9.04E+02	9.01E+02	4.75E+00
10	1.80E+03	**1.14E+03**	2.83E+02	1.60E+03	1.35E+03	2.57E+02	1.61E+03	1.45E+03	1.65E+02	1.79E+03	1.47E+03	2.79E+02
11	1.17E+03	1.12E+03	4.95E+01	2.18E+03	**1.11E+03**	2.07E+03	1.15E+03	1.12E+03	4.29E+01	1.14E+03	1.13E+03	1.27E+01
12	7.00E+03	3.08E+03	4.21E+03	5.09E+05	1.48E+05	3.17E+05	1.19E+06	3.84E+04	8.26E+05	7.02E+03	**2.33E+03**	4.49E+03
13	8.94E+03	**1.46E+03**	8.84E+03	4.60E+04	7.39E+03	7.23E+04	8.86E+03	5.68E+03	2.79E+03	5.77E+03	2.21E+03	5.87E+03
14	1.51E+03	1.46E+03	3.97E+01	1.11E+04	3.36E+03	8.09E+03	2.20E+03	1.45E+03	1.51E+03	2.74E+03	**1.43E+03**	2.24E+03
15	3.08E+03	1.89E+03	1.11E+03	4.07E+03	**1.83E+03**	2.63E+03	5.13E+03	3.24E+03	1.32E+03	7.81E+03	2.65E+03	4.31E+03
16	1.83E+03	**1.60E+03**	1.23E+02	1.77E+03	1.70E+03	4.82E+01	1.71E+03	1.61E+03	7.47E+01	1.87E+03	1.79E+03	1.04E+02
17	1.76E+03	1.73E+03	2.44E+01	1.75E+03	1.74E+03	2.17E+00	1.76E+03	**1.72E+03**	5.97E+01	1.74E+03	1.74E+03	4.91E+00
18	7.55E+03	**1.97E+03**	7.32E+03	1.10E+04	3.17E+03	5.67E+03	1.70E+04	5.60E+03	1.19E+04	1.86E+04	2.71E+03	1.68E+04
19	3.29E+03	**1.92E+03**	1.82E+03	5.49E+03	2.16E+03	2.73E+03	4.82E+03	1.94E+03	4.89E+03	2.12E+04	2.51E+03	3.02E+04
20	2.12E+03	**2.02E+03**	7.09E+01	2.04E+03	2.03E+03	8.81E+00	2.14E+03	2.11E+03	4.47E+01	2.14E+03	2.12E+03	2.40E+01
21	2.24E+03	**2.20E+03**	5.60E+01	2.28E+03	2.22E+03	4.17E+01	2.30E+03	2.29E+03	3.26E+00	2.30E+03	2.29E+03	6.62E+00
22	2.30E+03	**2.22E+03**	2.28E+01	2.32E+03	2.31E+03	2.74E+00	2.31E+03	2.30E+03	8.39E+00	2.31E+03	2.30E+03	1.86E+01
23	2.64E+03	2.61E+03	1.91E+01	2.65E+03	2.63E+03	1.18E+01	2.61E+03	**2.61E+03**	5.53E+00	2.64E+03	2.63E+03	7.35E+00
24	2.76E+03	**2.50E+03**	8.31E+01	2.71E+03	2.57E+03	1.01E+02	2.73E+03	2.69E+03	2.86E+01	2.74E+03	2.72E+03	2.65E+01
25	2.93E+03	**2.90E+03**	2.25E+01	2.95E+03	2.93E+03	1.25E+01	2.94E+03	2.93E+03	9.50E+00	2.92E+03	2.90E+03	2.53E+01
26	3.10E+03	2.80E+03	3.26E+02	2.90E+03	**2.74E+03**	1.76E+02	3.21E+03	2.96E+03	3.73E+02	2.90E+03	2.82E+03	7.15E+01
27	3.11E+03	3.10E+03	1.61E+01	3.15E+03	3.11E+03	3.64E+01	3.11E+03	**3.09E+03**	3.50E+01	3.13E+03	3.10E+03	3.14E+01
28	3.38E+03	3.17E+03	1.22E+02	3.22E+03	**3.14E+03**	1.54E+02	3.31E+03	3.18E+03	8.72E+01	3.27E+03	3.16E+03	8.36E+01
29	3.27E+03	3.17E+03	6.51E+01	3.22E+03	3.18E+03	3.54E+01	3.24E+03	3.18E+03	7.81E+01	3.24E+03	**3.16E+03**	7.12E+01
Rank		**1**			**4**			**2**			**3**	

Table 2 Results of 30D problems

	GDA			GA			GWO			PSO		
Fn	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std
1	2.92E+08	4.67E+06	4.79E+08	1.49E+08	1.11E+08	4.18E+07	1.39E+09	2.29E+08	1.93E+09	1.17E+09	**3.14E+03**	2.40E+09
3	1.71E+04	**2.34E+03**	1.23E+04	1.40E+05	1.06E+05	4.30E+04	3.48E+04	3.04E+04	3.56E+03	2.66E+04	1.90E+04	7.10E+03
4	6.11E+02	5.09E+02	5.66E+01	7.53E+02	7.10E+02	3.05E+01	5.53E+02	5.07E+02	3.28E+01	5.97E+02	**5.06E+02**	1.17E+02
5	6.63E+02	5.99E+02	3.77E+01	6.70E+02	6.28E+02	4.00E+01	6.02E+02	**5.68E+02**	2.95E+01	6.11E+02	5.91E+02	2.90E+01
6	6.44E+02	6.24E+02	9.79E+00	6.25E+02	6.19E+02	4.34E+00	6.10E+02	**6.04E+02**	5.04E+00	6.39E+02	6.29E+02	8.68E+00
7	1.01E+03	8.47E+02	7.33E+01	9.29E+02	8.96E+02	3.40E+01	8.61E+02	**8.02E+02**	4.15E+01	8.55E+02	8.37E+02	1.80E+01
8	9.38E+02	**8.54E+02**	3.32E+01	9.56E+02	9.43E+02	1.58E+01	8.78E+02	8.72E+02	5.47E+00	9.04E+02	8.94E+02	1.04E+01
9	3.82E+03	1.99E+03	1.04E+03	1.23E+03	**1.05E+03**	1.70E+02	1.88E+03	1.43E+03	5.51E+02	3.11E+03	1.92E+03	1.20E+03
10	3.67E+03	**1.59E+03**	8.68E+02	5.51E+03	5.13E+03	3.87E+02	4.38E+03	3.97E+03	2.91E+02	4.59E+03	4.40E+03	2.97E+02
11	1.29E+03	**1.18E+03**	7.85E+01	7.85E+03	2.99E+03	5.10E+03	2.02E+03	1.34E+03	1.23E+03	1.23E+03	1.20E+03	2.39E+01
12	4.90E+06	**2.41E+04**	5.75E+06	6.43E+06	4.45E+06	1.85E+06	4.40E+07	4.27E+06	3.94E+07	2.15E+06	2.32E+05	1.79E+06
13	3.76E+04	**6.17E+03**	2.69E+04	9.97E+06	2.70E+06	8.25E+06	6.32E+05	7.65E+04	9.21E+05	2.73E+04	1.14E+04	2.36E+04
14	5.23E+04	**2.48E+03**	4.52E+04	1.68E+06	2.78E+05	1.21E+06	4.46E+05	2.89E+04	4.83E+05	1.59E+04	2.89E+03	1.17E+04
15	1.36E+04	2.76E+03	1.10E+04	7.98E+05	1.47E+05	7.71E+05	1.32E+07	8.23E+04	2.48E+07	4.24E+03	**1.85E+03**	2.65E+03
16	2.95E+03	**2.01E+03**	3.64E+02	2.74E+03	2.44E+03	2.89E+02	2.40E+03	2.27E+03	1.19E+02	2.72E+03	2.53E+03	2.26E+02
17	2.38E+03	1.87E+03	2.86E+02	2.09E+03	2.05E+03	4.65E+01	1.90E+03	**1.79E+03**	1.15E+02	2.24E+03	2.04E+03	2.44E+02
18	3.78E+05	**1.09E+04**	6.10E+05	3.04E+06	2.38E+06	9.82E+05	3.51E+05	6.58E+04	4.03E+05	1.15E+05	8.18E+04	2.44E+04
19	1.10E+04	**2.04E+03**	2.12E+04	1.35E+06	5.35E+05	8.10E+05	3.37E+06	5.94E+04	5.16E+06	3.75E+04	7.63E+03	5.09E+04
20	2.52E+03	**2.17E+03**	1.70E+02	2.41E+03	2.36E+03	3.72E+01	2.33E+03	2.18E+03	1.22E+02	2.47E+03	2.42E+03	7.31E+01
21	2.45E+03	2.35E+03	5.04E+01	2.46E+03	2.44E+03	3.13E+01	2.38E+03	**2.35E+03**	2.05E+01	2.41E+03	2.40E+03	1.41E+01
22	3.32E+03	**2.34E+03**	1.70E+03	2.64E+03	2.58E+03	5.69E+01	2.64E+03	2.53E+03	1.56E+02	4.48E+03	2.43E+03	1.90E+03
23	2.97E+03	2.78E+03	1.00E+02	2.93E+03	2.90E+03	3.72E+01	2.74E+03	**2.72E+03**	1.54E+01	2.86E+03	2.83E+03	3.39E+01
24	3.13E+03	2.94E+03	8.45E+01	3.17E+03	3.09E+03	7.02E+01	2.91E+03	**2.90E+03**	7.69E+00	3.09E+03	3.02E+03	7.10E+01
25	2.98E+03	2.90E+03	4.34E+01	3.07E+03	3.06E+03	1.11E+01	2.98E+03	2.94E+03	4.39E+01	2.89E+03	**2.89E+03**	1.00E+01
26	5.56E+03	**2.93E+03**	1.13E+03	4.27E+03	3.94E+03	3.12E+02	4.41E+03	4.08E+03	3.75E+02	4.65E+03	3.55E+03	1.16E+03
27	3.32E+03	**3.23E+03**	5.73E+01	3.42E+03	3.36E+03	4.64E+01	3.24E+03	3.23E+03	9.51E+00	3.27E+03	3.23E+03	3.11E+01
28	3.38E+03	3.25E+03	4.99E+01	3.51E+03	3.47E+03	4.51E+01	3.52E+03	3.35E+03	2.85E+02	3.29E+03	**3.18E+03**	1.38E+02
29	4.20E+03	**3.60E+03**	3.18E+02	4.15E+03	3.82E+03	2.91E+02	3.72E+03	3.65E+03	7.67E+01	4.04E+03	3.88E+03	1.60E+02
Rank		**1**			**4**			**2**			**3**	

Table 3 Results of 50D problems

	GDA			GA			GWO			PSO		
Fn	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std	Mean	Best	Std
1	4.91E+09	**3.15E+07**	4.04E+09	8.65E+09	8.24E+09	5.21E+08	7.78E+09	5.61E+09	2.16E+09	2.11E+09	8.65E+08	8.78E+08
3	3.63E+04	**1.33E+04**	1.37E+04	2.19E+05	1.83E+05	2.91E+04	1.08E+05	9.50E+04	1.11E+04	1.20E+05	9.05E+04	2.96E+04
4	7.51E+02	**5.44E+02**	1.63E+02	1.35E+03	1.17E+03	1.29E+02	1.27E+03	9.61E+02	2.67E+02	9.24E+02	6.41E+02	4.54E+02
5	8.72E+02	7.46E+02	6.40E+01	8.26E+02	8.01E+02	2.09E+01	6.88E+02	**6.65E+02**	2.55E+01	7.41E+02	6.95E+02	4.17E+01
6	6.61E+02	6.43E+02	1.04E+01	6.40E+02	6.29E+02	9.09E+00	6.19E+02	**6.14E+02**	5.93E+00	6.44E+02	6.42E+02	2.25E+00
7	1.66E+03	1.33E+03	1.11E+02	1.21E+03	1.15E+03	4.86E+01	1.01E+03	9.93E+02	1.69E+01	1.12E+03	**9.91E+02**	1.44E+02
8	1.23E+03	1.04E+03	9.10E+01	1.18E+03	1.15E+03	2.41E+01	9.95E+02	**9.66E+02**	2.80E+01	1.01E+03	9.78E+02	4.37E+01
9	1.72E+04	7.32E+03	8.19E+03	1.07E+04	8.87E+03	1.90E+03	5.84E+03	**5.10E+03**	8.19E+02	8.67E+03	8.04E+03	8.27E+02
10	9.89E+03	**6.11E+03**	2.20E+03	1.05E+04	1.00E+04	4.38E+02	7.94E+03	6.18E+03	2.97E+03	7.22E+03	6.95E+03	3.03E+02
11	3.00E+03	1.37E+03	1.58E+03	2.01E+04	1.19E+04	6.41E+03	5.28E+03	3.25E+03	2.75E+03	1.58E+03	**1.35E+03**	2.18E+02
12	5.14E+07	**4.40E+06**	5.24E+07	1.70E+08	5.37E+07	8.21E+07	7.97E+08	1.25E+08	6.95E+08	1.34E+09	1.06E+07	1.84E+09
13	3.99E+05	**1.83E+04**	1.87E+06	3.46E+07	2.25E+07	1.09E+07	2.97E+08	1.35E+08	3.09E+08	3.97E+08	4.26E+04	5.03E+08
14	2.06E+06	1.19E+05	3.72E+06	9.34E+06	4.60E+06	4.98E+06	9.60E+05	**7.49E+04**	7.45E+05	4.78E+05	1.72E+05	2.50E+05
15	4.15E+04	5.70E+03	3.38E+04	7.15E+06	2.43E+06	5.94E+06	4.96E+06	3.55E+04	5.84E+06	9.30E+03	**2.57E+03**	7.06E+03
16	4.03E+03	**2.37E+03**	5.91E+02	3.61E+03	3.04E+03	4.62E+02	3.11E+03	2.80E+03	4.04E+02	3.13E+03	2.81E+03	3.79E+02
17	3.58E+03	**2.51E+03**	4.25E+02	3.32E+03	3.11E+03	1.93E+02	2.81E+03	2.70E+03	1.71E+02	3.13E+03	2.95E+03	2.33E+02
18	9.50E+05	**1.96E+05**	6.48E+05	7.90E+06	2.83E+06	7.67E+06	4.77E+06	9.10E+05	4.62E+06	6.88E+05	2.93E+05	3.93E+05
19	2.81E+04	3.67E+03	1.54E+04	8.27E+05	6.47E+05	2.29E+05	9.70E+05	4.75E+05	4.36E+05	3.29E+05	**2.74E+03**	4.00E+05
20	3.34E+03	2.47E+03	4.15E+02	3.00E+03	2.94E+03	7.55E+01	2.55E+03	**2.37E+03**	1.96E+02	3.10E+03	2.75E+03	2.40E+02
21	2.62E+03	2.49E+03	6.66E+01	2.68E+03	2.66E+03	1.87E+01	2.49E+03	**2.44E+03**	3.55E+01	2.60E+03	2.54E+03	6.94E+01
22	1.01E+04	**2.38E+03**	2.05E+03	7.89E+03	3.75E+03	4.66E+03	7.93E+03	6.92E+03	7.53E+02	8.67E+03	7.86E+03	6.56E+02
23	3.82E+03	3.31E+03	2.28E+02	3.26E+03	3.15E+03	9.68E+01	2.98E+03	**2.91E+03**	8.44E+01	3.28E+03	3.22E+03	5.85E+01
24	3.59E+03	3.23E+03	1.80E+02	3.55E+03	3.52E+03	5.78E+01	3.16E+03	**3.07E+03**	8.22E+01	3.38E+03	3.24E+03	1.23E+02
25	3.21E+03	**3.06E+03**	6.32E+01	3.88E+03	3.79E+03	8.24E+01	3.87E+03	3.70E+03	1.80E+02	3.10E+03	3.06E+03	4.11E+01
26	1.26E+04	7.33E+03	1.65E+03	8.18E+03	6.59E+03	1.78E+03	6.09E+03	**5.76E+03**	3.43E+02	7.45E+03	6.96E+03	4.44E+02
27	4.01E+03	3.50E+03	2.96E+02	4.26E+03	4.16E+03	9.48E+01	3.58E+03	3.54E+03	4.03E+01	3.58E+03	**3.36E+03**	2.04E+02
28	3.71E+03	**3.39E+03**	1.84E+02	4.76E+03	4.55E+03	1.86E+02	4.22E+03	4.00E+03	2.48E+02	3.78E+03	3.51E+03	3.13E+02
29	6.13E+03	4.46E+03	6.49E+02	5.70E+03	5.44E+03	3.72E+02	4.63E+03	4.45E+03	2.04E+02	4.60E+03	**4.40E+03**	1.35E+02
Rank		**1**			**4**			**2**			**3**	

The differences between the best values yielded by the algorithms and the optimum values of the functions (i.e., the error values) were computed. The sums of the error values for 10D, 30D and 50D for all types of functions are presented in Table 4. As shown in Fig. 4a, an analysis of the error sums indicates that GDA exhibits the lowest error sum in unimodal, hybrid, and composite functions, while in multimodal functions, it achieves the lowest error sum following GWO. The algorithm's run time was also analyzed. As illustrated in Fig. 4b, the average run time was 0.214 seconds for 10D functions, 0.818 seconds for 30D functions, and 6.580 seconds for 50D functions. Furthermore, rapid convergence was observed across all function types for 10D, 30D, and 50D cases (Figs. 5a and 5b, 6a and 6b, 7a and 7b).

Table 4 Total errors by function type

Algorithm	Unimodal	Multimodal	Hybrid	Composition
GDA	**3.61E+07**	1.51E+04	**4.78E+06**	**1.67E+04**
GA	8.36E+09	2.44E+04	9.74E+07	2.36E+04
GWO	5.84E+09	**1.49E+04**	2.66E+08	2.23E+04
PSO	8.65E+08	1.94E+04	1.14E+07	2.32E+04

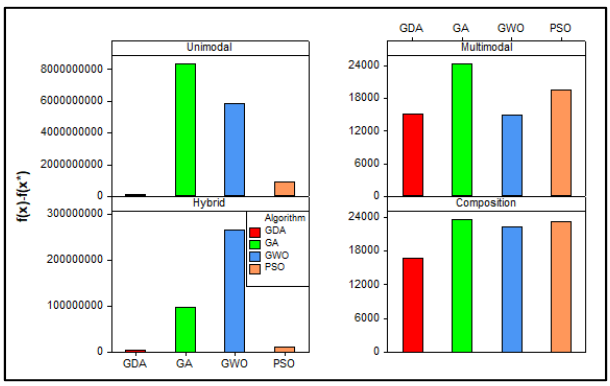


Fig. 4a Total errors by function type

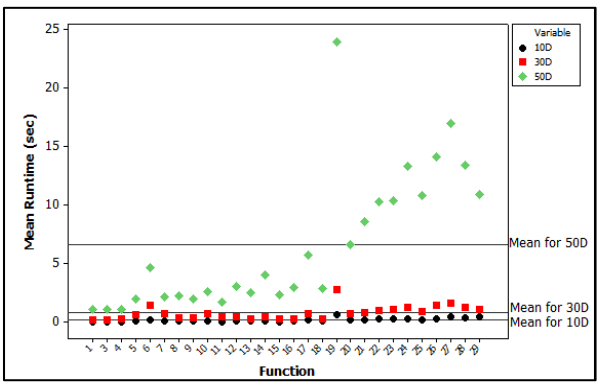


Fig. 4b Run times by function dimension

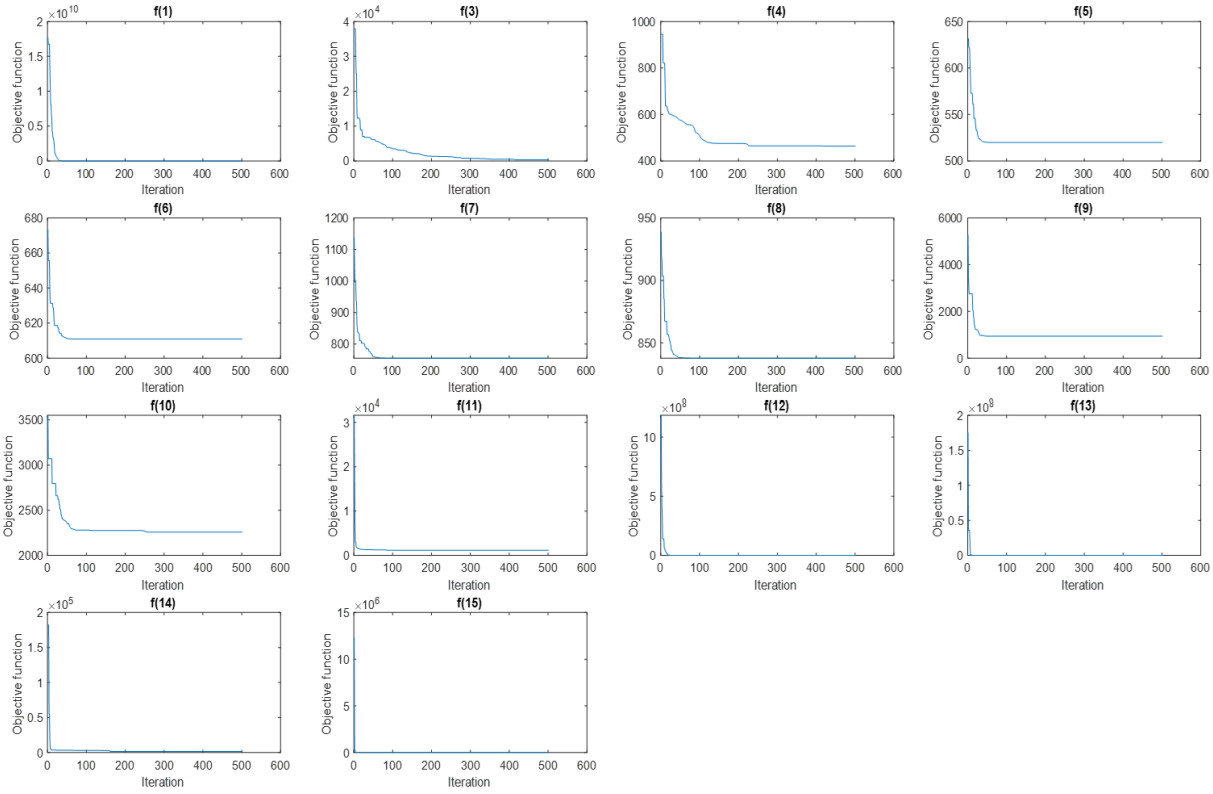


Fig. 5a 10D convergence graphs for f(1)-f(15)

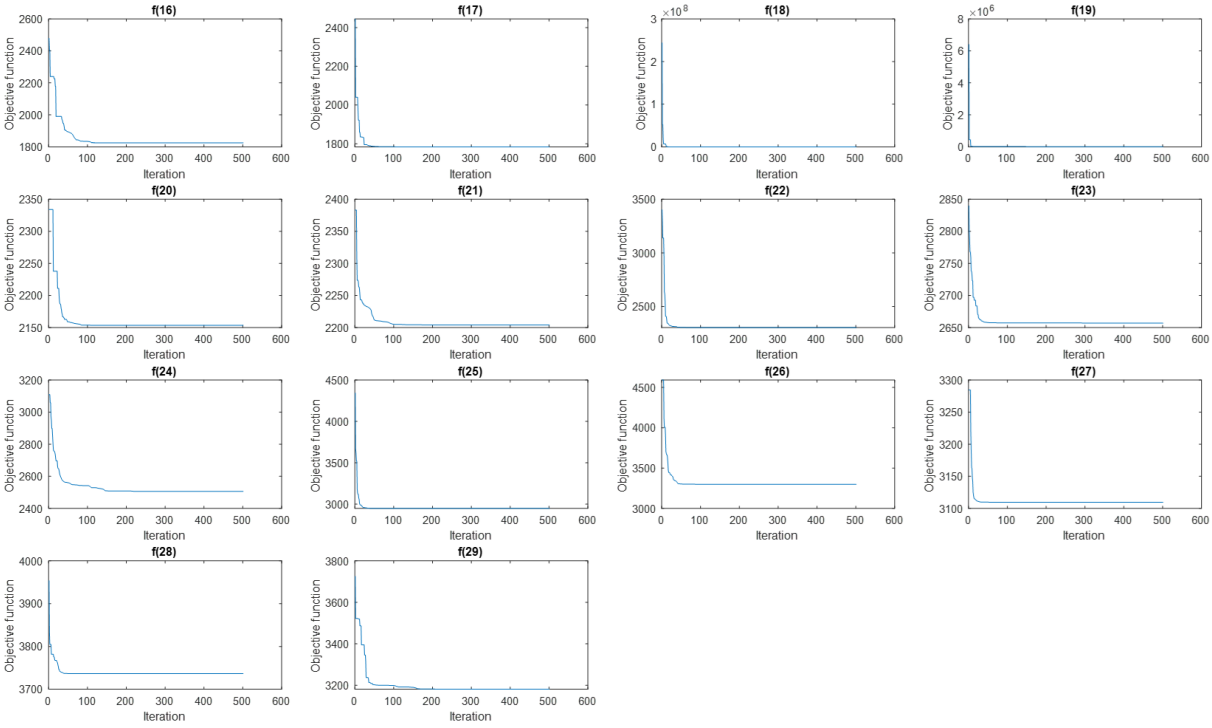


Fig. 5b 10D convergence graphs for f(16)-f(29)

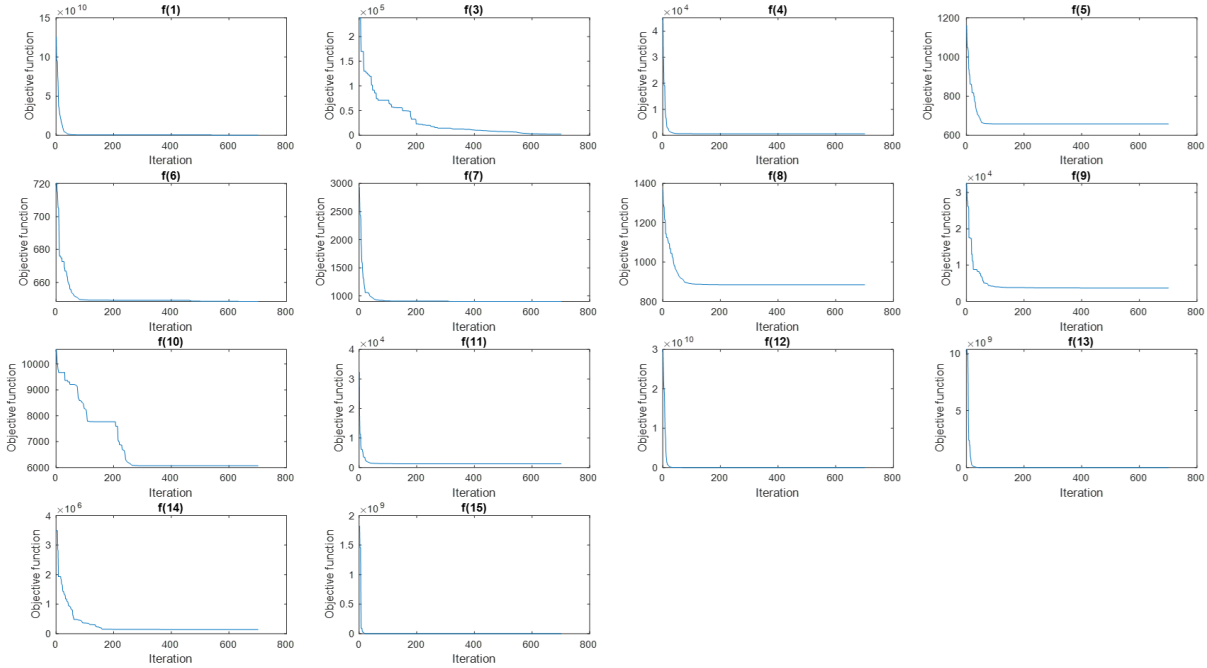


Fig. 6a 30D convergence graphs for f(1)-f(15)

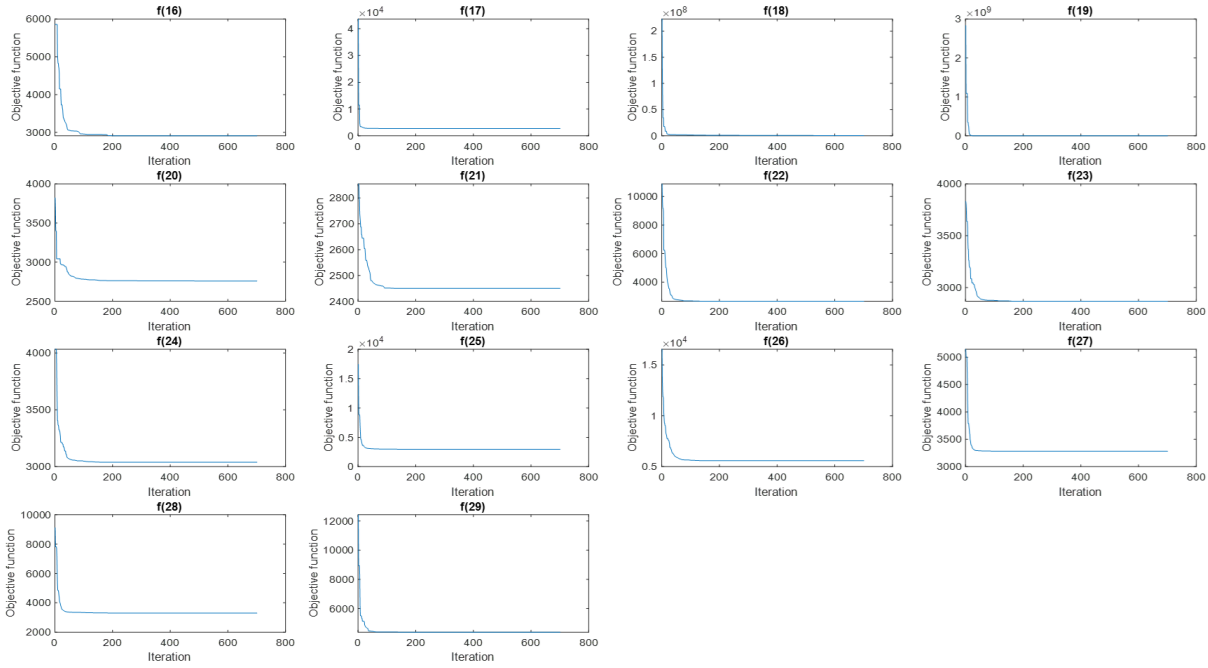


Fig. 6b 30D convergence graphs for f(16)-f(29)

The error values were first analyzed using the Friedman test and then using the Wilcoxon signed rank test to investigate whether the GDA had a statistically significant difference. Across all 10D, 30D, and 50D groups, the results of the Friedman test indicated that GDA had the lowest rank value. Similarly, according to the Wilcoxon signed-rank test, GDA was significantly different from the other algorithms and exhibited a lower error value (Table 5).

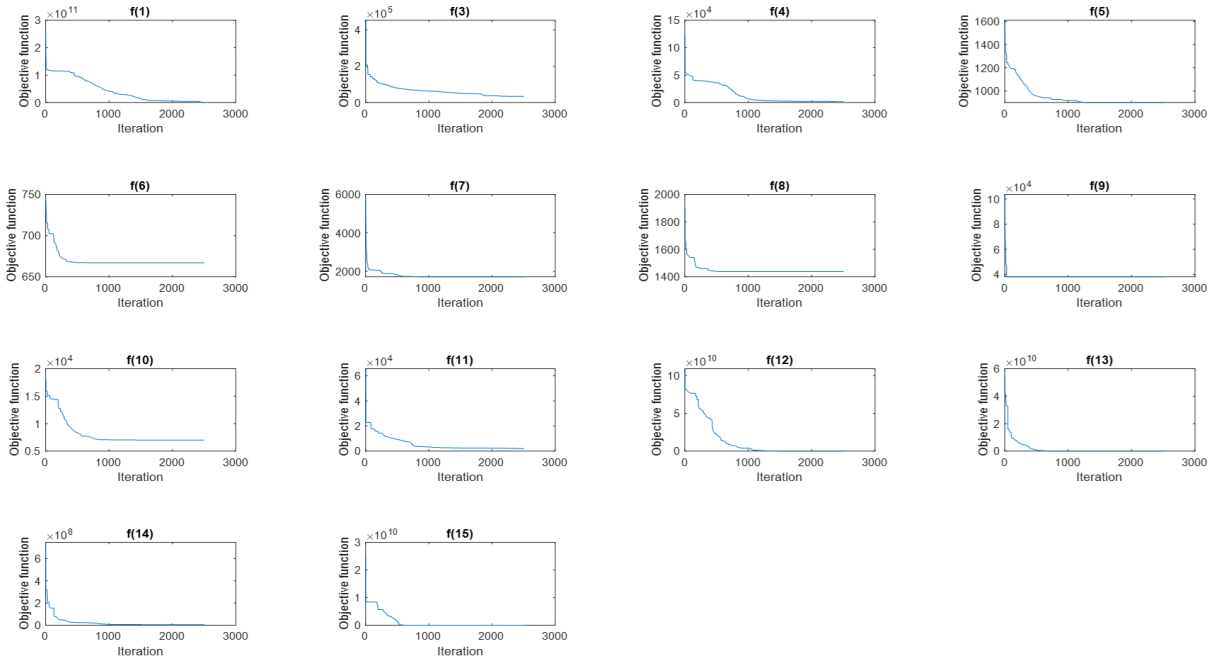


Fig. 7a 50D convergence graphs for f(1)-f(15)

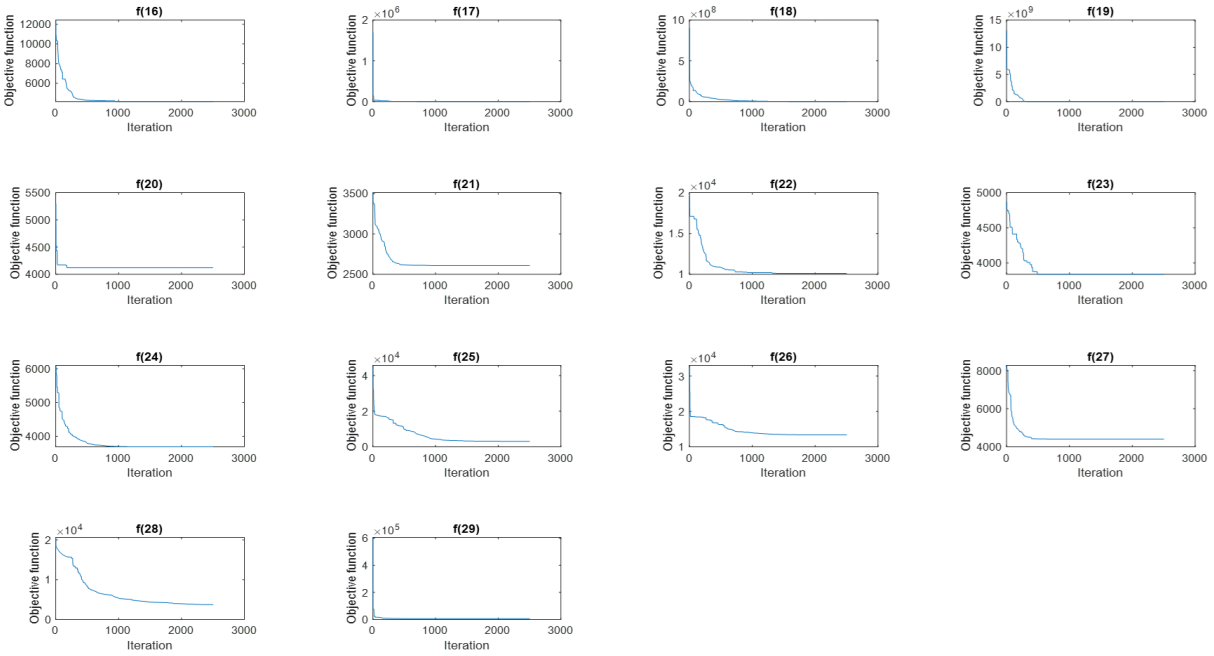


Fig. 7b 50D convergence graphs for f(16)-f(29)

Table 5 Results of statistical analysis

Algorithm	10 D		30 D		50 D	
	Friedman test	Wilcoxon test (*p*-values)	Friedman test	Wilcoxon test (*p*-values)	Friedman test	Wilcoxon test (*p*-values)
GDA	1.61	-	1.68	-	1.96	
GWO	2.57	0.000	2.29	0.005	2.18	0.027
PSO	2.75	0.002	2.39	0.010	2.25	0.026
GA	3.07	0.001	3.64	0.000	3.61	0.000

4. Experiments on engineering design problems

To assess whether the GDA is an effective optimization approach, it was applied to three well-known engineering design problems: welded beam design, pressure vessel design, and compression spring design. The performance of GDA was benchmarked against that of GA, GWO, and PSO, using the results reported by Dehghani *et al.* for comparison [46]. It was also compared with CMA-ES and DE, using the results reported by Amiri *et al.* for CMA-ES and by Debnath *et al.* for DE [47, 48].

The welded beam design problem is a structural optimization benchmark that focuses on minimizing the total cost of a welded beam subject to mechanical constraints such as maximum stress, deflection, buckling, and weld strength (Fig. 8). It includes four decision variables: weld size (h), weld length (l), and cross-sectional dimensions of the beam (b, t). Deb provided one of the most cited and complete formulations of this problem, which remains widely used [49]. More recently, modern metaheuristic studies have revisited this problem to test multi-objective and constraint-handling capabilities in challenging search spaces [50].

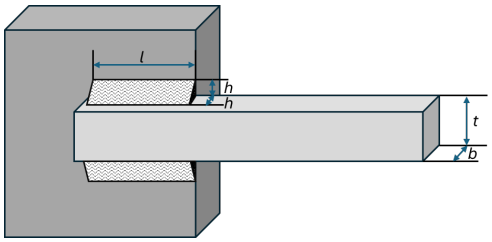


Fig. 8 Technical illustration of the welded beam design problem

As shown in Table 6, the results of the welded beam design problem solved using the GDA are compared with those of other optimization algorithms. GDA outperformed the competing algorithms by achieving a lower objective function value. The optimal solution found by GDA corresponds to a cost of 1.72486, attained at the parameter vector (0.20573, 3.47045, 9.03659, 0.20573).

The pressure vessel design problem is a classical benchmark in engineering optimization, extensively used to evaluate the performance of metaheuristic algorithms. The objective is to minimize the total cost of constructing a cylindrical vessel subject to various constraints, including material thickness, internal pressure, volume requirements, and manufacturability (Fig. 9). The design variables typically include shell thickness (T_s), head thickness (T_h), inner radius (R), and vessel length (L). While this problem was first formally structured and popularized in the early 2000s [51], recent studies continue to employ it to benchmark hybrid and nature-inspired optimization algorithms due to its nonlinear, mixed-integer characteristics [52].

Table 6 Comparative results for the welded beam design problem

Algorithms	Best	Mean	Worst	Std. Dev.	Rank	*h*	*l*	*t*	*b*
GDA	**1.72486**	1.75233	1.82173	0.02673	**1**	0.20573	3.47045	9.03659	0.20573
GA	1.83625	1.86353	2.03525	0.13949	5	0.20649	3.63587	10	0.20325
GWO	1.72547	1.72968	1.74165	0.00487	4	0.20561	3.4721	9.04093	0.20571
PSO	1.87397	2.11924	2.32013	0.03482	6	0.16417	4.03254	10	0.22365
CMA-ES	1.7249	1.7934	2.3189	0.13205	3	N/A	N/A	N/A	N/A
DE	1.7249	N/A	N/A	0.00000	2	0.20573	3.47049	9.03366	0.20573

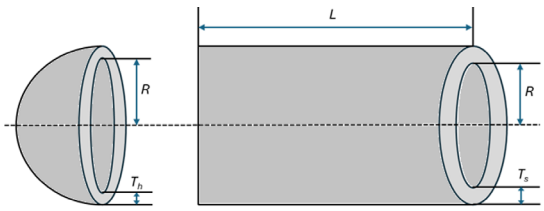


Fig. 9 Technical illustration of the pressure vessel design problem

Table 7 Comparative results for the pressure vessel design problem

Algorithms	Best	Mean	Worst	Std. Dev.	Rank	T_s	T_h	R	L
GDA	**5,885.33**	6,388.5	6,836.4	271.22	**1**	0.7782	0.3846	40.3196	200
GA	6,550.02	6,643.99	8,005.44	657.52	6	1.0995	0.9066	44.4564	179.6589
GWO	6,011.51	6,477.31	7,250.92	327.01	3	0.8457	0.4186	43.8163	156.3816
PSO	5,890.33	6,264.01	7,005.75	496.13	2	0.7524	0.3995	40.4525	198.0027
CMA-ES	6,059.7	6,534.1	7,368.1	524.75	5	N/A	N/A	N/A	N/A
DE	6,059.7	N/A	N/A	31,698	4	0.8125	0.4375	42.0985	176.636

As shown in Table 7, the GDA outperformed other optimization algorithms in solving the pressure vessel design problem by yielding a lower total cost. The optimal solution was obtained at the parameter vector (0.7782, 0.3846, 40.3196, 200), resulting in an objective function value of 5,885.33.

The spring design problem aims to minimize the weight (or volume) of a compression spring while satisfying stress, deflection, and geometric constraints (Fig. 10). The design involves selecting the wire diameter (d), mean coil diameter (D), and number of active coils (n). It has been extensively studied as a constrained nonlinear optimization problem in foundational texts [53]. In recent years, it has been revisited using chaos-enhanced and hybrid optimization algorithms due to its complex constraint interactions and multiple local optima [54]. The simplicity of the objective combined with complex feasible regions makes it a preferred test case in algorithm development.

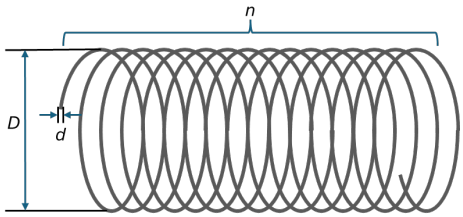


Fig. 10 Technical illustration of the compression spring design problem

Table 8 presents a comparison of the results obtained from solving the compression spring design problem using the GDA and those obtained by other algorithms. CMA-ES achieved a slightly lower weight value compared to the other algorithms and GDA; however, the result obtained by CMA-ES is very close to that of GDA. Specifically, using GDA, a weight of 0.01268 was obtained at the solution point with parameter values (0.05142, 0.35002, 11.69855).

Based on the results of these three engineering design problems, it can be concluded that the GDA is effective in obtaining optimal or near-optimal solutions. Furthermore, it demonstrates competitive performance not only in terms of the best solution but also with respect to the average, worst-case values, and standard deviation.

Table 8 Comparative results for the compression spring design problem

Algorithms	Best	Mean	Worst	Std. Dev.	Rank	d	D	n
GDA	0.01268	0.01374	0.01678	0.00093	2	0.05142	0.35002	11.69855
GA	0.01278	0.01307	0.01521	0.00038	3	0.05025	0.31635	15.2396
GWO	0.01282	0.01446	0.01784	0.00162	4	0.05	0.31596	14.22623
PSO	0.01304	0.01404	0.01625	0.00207	5	0.0501	0.31011	14
CMA-ES	**0.01267**	0.01349	0.01777	0.00124	**1**	N/A	N/A	N/A
DE	0.0318	N/A	N/A	0.00000	6	1.4194	0.4809	14.7014

5. Conclusion

This study proposes an algorithm called GDA, which is inspired by the propagation of circular waves formed by raindrops falling into water. The motion of the waves is associated with the ancient golden ratio, which is also considered mysterious. This is an algorithm that can achieve success quickly. The GDA is a practical algorithm that is easy to understand and easy to implement. The computational steps are short, and there are few parameters to be used.

The proposed algorithm was tested using the 10D, 30D, and 50D CEC 2017 functions, which comprise 28 single-objective, real-parameter numerical optimization problems. It was found that the algorithm converges quickly and achieves successful results across all 10D, 30D, and 50D functions. The algorithm's performance is compared with popular and practical algorithms such as the GA, GWO and PSO. In unimodal, hybrid, and composite functions, the lowest error sum was achieved using the GDA. In multimodal functions, it was the second algorithm after the GWO. In 50 % of the 10D, 30D, and 50D functions, GDA ranked first, while in 32 % of the functions, it secured second place.

The performance of optimization algorithms may vary depending on the type of test functions. Success on unimodal functions typically reflects convergence capability, whereas success on multimodal functions indicates the algorithm's ability to avoid getting trapped in local optima. In contrast, strong performance on hybrid and composition functions demonstrates both the ability to escape local optima and the algorithm's balance between exploration and exploitation strategies [55, 56].

From this perspective, the GDA yielded successful results across all four function categories. Therefore, it can be concluded that the algorithm exhibits satisfactory convergence behavior, strong local optima avoidance, and a well-balanced search strategy.

Furthermore, the GDA produced promising results in three well-established engineering design problems: welded beam design, pressure vessel design, and compression spring design, thereby validating its practical applicability to real-world optimization tasks. The results obtained using GDA for these three engineering problems were compared not only with GA, GWO, and PSO, but also with the well-established CMA-ES and DE algorithms. For the spring design problem, the result obtained by GDA was slightly behind CMA-ES by a very small margin, whereas for the other problems, GDA achieved the best solutions.

Globally considered, it can be said that the GDA is a competitive algorithm. In prospective studies, the algorithm can be applied to multi-objective optimization problems, discrete optimization problems, and special-purpose studies. In this sense, GDA can be a good method that can potentially enrich this field of research.

References

[1] Kaya, S., Gümüşçü, A., Aydilek, İ.B., Karaçizmeli, İ.H., Tenekeci, M.E. (2021). Solution for flow shop scheduling problems using chaotic hybrid firefly and particle swarm optimization algorithm with improved local search, *Soft Computing,* Vol. 25, 7143-7154, doi: 10.1007/s00500-021-05673-w.

[2] Civicioglu, P., Besdok, E. (2024). Colony-based search algorithm for numerical optimization, *Applied Soft Computing*, Vol. 151, Article No. 111162, doi: 10.1016/j.asoc.2023.111162.

[3] Ruszczynski, A. (2006). *Nonlinear Optimization*, Princeton University Press, New Jersey, USA, doi: 10.1515/9781400841059.

[4] Nurmuhammed, M., Akdağ, O., Karadağ, T. (2024). Modified Archimedes optimization algorithm for global optimization problems: A comparative study, *Neural Computing and Applications,* Vol. 36, 8007-8038, doi: 10.1007/s00521-024-09497-1.

[5] Doğan, B., Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex search algorithm, *Information Sciences*, Vol. 293, 125-145, doi: 10.1016/j.ins.2014.08.053.

[6] Gümüşçü, A., Kaya, S., Tenekeci, M.E., Karaçizmeli, İ.H., Aydilek, İ.B. (2022). The impact of local search strategies on chaotic hybrid firefly particle swarm optimization algorithm in flow-shop scheduling, *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 8, Part B, 6432-6440, doi: 10.1016/j.jksuci.2021.07.017.

[7] Blum, C., Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys,* Vol. 35, No. 3, 268-308, doi: 10.1145/937503.937505.

[8] Mohamed, A.W., Hadi, A.A., Mohamed, A.K. (2020). Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm, *International Journal of Machine Learning and Cybernetics,* Vol. 11, 1501-1529, doi: 10.1007/s13042-019-01053-x.

[9] Agrawal, P., Abutarboush, H.F., Ganesh, T., Mohamed, A.W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019), *IEEE Access*, Vol. 9, 26766-26791, doi: 10.1109/ACCESS.2021.3056407.

[10] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, Vol. 13, No. 5, 533-549, doi: 10.1016/0305-0548(86)90048-1.

[11] Sörensen, K., Glover, F.W. (2013). Metaheuristics, In: Gass, S.I., Fu, M.C. (eds.), *Encyclopedia of operations research and management science*, Vol. 62, 960-970, doi: 10.1007/978-1-4419-1153-7_1167.

[12] Holland, J.H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, The MIT Press, Cambridge, Massachusetts, USA, doi: 10.7551/mitpress/1090.001.0001.
[13] Kirkpatrick, S., Gelatt Jr, C.D., Vecchi, M.P. (1983). Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, 671-680, doi: 10.1126/science.220.4598.671.
[14] Glover, F. (1989). Tabu search – part I, *INFORMS Journal on Computing,* Vol. 1, No. 3, 190-206, doi: 10.1287/ijoc.1.3.190.
[15] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization, In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Australia, Vol. 4, 1942-1948, doi: 10.1109/ICNN.1995.488968.
[16] Dorigo, M., Maniezzo, V., Colorni, A. (1991). *Positive feedback as a search strategy*, Technical report, No. 91-016, Politecnico di Milano, Milan, Italy.
[17] Dorigo, M., Di Caro, G. (1999). *The ant colony optimization meta-heuristic in new ideas in optimization*, McGraw Hill, London, United Kingdom.
[18] Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*, Technical report-tr06, 200, 1-10, Department of Computer Engineering, Engineering Faculty, Erciyes University, Kayseri, Turkey.
[19] Yang, X.-S. (2008). *Nature-inspired metaheuristic algorithms*, Luniver Press, Cambridge, United Kingdom.
[20] Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, Vol. 110-111, 151-166, doi: 10.1016/j.compstruc.2012.07.010.
[21] Shah-Hosseini, H. (2007). Problem solving by intelligent water drops, In: *Proceedings of 2007 IEEE Congress on Evolutionary Computation,* Singapore, 3226-3231, doi: 10.1109/CEC.2007.4424885.
[22] Shah-Hosseini, H. (2012). An approach to continuous optimization by the intelligent water drops algorithm, *Procedia-Social and Behavioral Sciences*, Vol. 32, 224-229, doi: 10.1016/j.sbspro.2012.01.033.
[23] Alijla, B.O., Wong, L.-P., Lim, C.P., Khader, A.T., Al-Betar, M.A. (2014). A modified intelligent water drops algorithm and its application to optimization problems, *Expert Systems with Applications*, Vol. 41, No. 15, 6555-6569, doi: 10.1016/j.eswa.2014.05.010.
[24] Zheng, Y.-J. (2015). Water wave optimization: A new nature-inspired metaheuristic, *Computers & Operations Research*, Vol. 55, 1-11, doi: 10.1016/j.cor.2014.10.008.
[25] Colak, M.E., Varol, A. (2015). A novel intelligent optimization algorithm inspired from circular water waves, *Elektronika ir Elektrotechnika*, Vol. 21, No. 5, 3-6. doi: 10.5755/j01.eee.21.5.13316.
[26] Kiefer, J. (1953). Sequential minimax search for a maximum, *Proceedings of the American Mathematical Society*, Vol. 4, 502-506, doi: 10.2307/2032161.
[27] Manikantan, K., Arun, B.V., Yaradoni, D.K.S. (2012). Optimal multilevel thresholds based on Tsallis entropy method using golden ratio particle swarm optimization for improved image segmentation, *Procedia Engineering*, Vol. 30, 364-371, doi: 10.1016/j.proeng.2012.01.873.
[28] Tanyıldızı, E., Demir, G. (2017). Golden sine algorithm: A novel math-inspired algorithm, *Advances in Electrical and Computer Engineering*, Vol. 17, No. 2, 71-78, doi: 10.4316/AECE.2017.02010.
[29] Nematollahi, A.F., Rahiminejad, A., Vahidi, B. (2020). A novel meta-heuristic optimization method based on golden ratio in nature, *Soft Computing,* Vol. 24, 1117-1151, doi: 10.1007/s00500-019-03949-w.
[30] Dey, A., Chattopadhyay, S., Singh, P.K., Ahmadian, A., Ferrara, M., Sarkar, R. (2020). A hybrid meta-heuristic feature selection method using golden ratio and equilibrium optimization algorithms for speech emotion recognition, *IEEE Access*, Vol. 8, 200953-200970, doi: 10.1109/ACCESS.2020.3035531.
[31] Nusair, K., Alasali, F. (2020). Optimal power flow management system for a power network with stochastic renewable energy resources using golden ratio optimization method, *Energies*, Vol. 13, No. 14, Article No. 3671, doi: 10.3390/en13143671.
[32] Chattopadhyay, S., Dey, A., Singh, P.K., Geem, Z.W., Sarkar, R. (2021). COVID-19 detection by optimizing deep residual features with improved clustering-based golden ratio optimizer, *Diagnostics*, Vol. 11, No. 2, Article No. 315, doi: 10.3390/diagnostics11020315.
[33] Koçer, H.G., Uymaz, S.A. (2021). A novel local search method for LSGO with golden ratio and dynamic search step, *Soft Computing,* Vol. 25, 2115-2130, doi: 10.1007/s00500-020-05284-x.
[34] Alasali, F., Tawalbeh, R., Ghanem, Z., Mohammad, F., Alghazzawi, M. (2021). A sustainable early warning system using rolling forecasts based on ANN and golden ratio optimization methods to accurately predict real-time water levels and flash flood, *Sensors*, Vol. 21, No. 13, Article No. 4598, doi: 10.3390/s21134598.
[35] Venunath, M., Sujatha, P., Koti, P. (2024). Identification of influential users in social media network using golden ratio optimization method, *Soft Computing,* Vol. 28, 2207-2222, doi: 10.1007/s00500-023-09218-1.
[36] Eirgash, M.A., Toğan, V. (2023). A novel oppositional teaching learning strategy based on the golden ratio to solve the time-cost-environmental impact trade-off optimization problems, *Expert Systems with Applications*, Vol. 224, Article No. 119995, doi: 10.1016/j.eswa.2023.119995.
[37] Kaveh, A., Dadras Eslamlou, A. (2020). Water strider algorithm: A new metaheuristic and applications, *Structures,* Vol. 25, 520-541, doi: 10.1016/j.istruc.2020.03.033.
[38] Dunlap, R.A. (1997). *The golden ratio and Fibonacci numbers*, World Scientific, Singapore, doi: 10.1142/9789812386304.
[39] Canva. Fibonacci spiral and the divine proportions from nature, from *https://www.canva.com/learn/what-is-the-golden-ratio/*, accessed May 6, 2025.
[40] İmre, E. (2022). *The beauty of the golden ratio effect on perception*, (in Turkish), PhD thesis, Trakya University, Edirne, Turkey.

[41] Ünver, G. (2017). *In terms of artistic anatomy effect to aesthetic perception of face morphometry* (in Turkish), MSc thesis, Trakya University, Edirne, Turkey.
[42] Canva. Drops falling into water, from *https://www.canva.com/photos/MAEFj44Io64-raindrops-in-a-puddle/*, accessed May 6, 2025.
[43] Aydilek, I.B. (2018). A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, *Applied Soft Computing*, Vol. 66, 232-249, doi: 10.1016/j.asoc.2018.02.025.
[44] Awad, N.H., Ali, M.Z., Suganthan, P.N., Liang, J.J., Qu, B.Y. (2016). *Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single-objective bound constrained real-parameter numerical optimization*, Technical Report, Nanyang Technological University, Singapore.
[45] Al-Baik, O., Alomari, S., Alssayed, O., Gochhait, S., Leonova, I., Dutta, U., Malik, O.P., Montazeri, Z., Dehghani, M. (2024). Pufferfish optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, *Biomimetics*, Vol. 9, No. 2, Article No. 65, doi: 10.3390/biomimetics9020065.
[46] Dehghani, M., Montazeri, Z., Dhiman, G., Malik, O.P., Morales-Menendez, R., Ramirez-Mendoza, R.A., Dehghani, A., Guerrero, J.M., Parra-Arroyo, L. (2020). A spring search algorithm applied to engineering optimization problems, *Applied Sciences*, Vol. 10, No. 18, Article No. 6173, doi: 10.3390/app10186173.
[47] Amiri, M.H., Mehrabi Hashjin, N., Montazeri, M., Mirjalili, S., Khodadadi, N. (2024). Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm, *Scientific Reports,* Vol. 14, Article No. 5032, doi: 10.1038/s41598-024-54910-3.
[48] Debnath, S., Baishya, S., Sen, D., Arif, W. (2021). A hybrid memory-based dragonfly algorithm with differential evolution for engineering application, *Engineering with Computers,* Vol. 37, 2775-2802, doi: 10.1007/s00366-020-00958-4.
[49] Deb, K. (2000). An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2-4, 311-338, doi: 10.1016/S0045-7825(99)00389-8.
[50] Chauhan, S., Vashishtha, G., Kumar, A. (2022). A symbiosis of arithmetic optimizer with slime mould algorithm for improving global optimization and conventional design problem, *The Journal of Supercomputing*, Vol. 78, No. 5, 6234-6274, doi: 10.1007/s11227-021-04105-8.
[51] Coello, C.A.C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems, *Computers in Industry*, Vol. 41, No. 2, 113-127, doi: 10.1016/S0166-3615(99)00046-9.
[52] El-Shorbagy, M.A., El-Refaey, A.M. (2022). A hybrid genetic–firefly algorithm for engineering design problems, *Journal of Computational Design and Engineering*, Vol. 9, No. 2, 706-730, doi: 10.1093/jcde/qwac013.
[53] Arora, J.S. (2004). *Introduction to optimum design,* Second edition, Elsevier Academic Press, San Diego, California, USA, doi: 10.1016/B978-0-12-064155-0/50028-8.
[54] Talatahari, S., Azizi, M. (2020). Optimization of constrained mathematical and engineering design problems using chaos game optimization, *Computers & Industrial Engineering*, Vol. 145, Article No. 106560, doi: 10.1016/j.cie.2020.106560
[55] Varol Altay, E., Altay, O. (2021). Comparison of current metaheuristic optimization algorithms with CEC2020 test functions, *Dicle University Journal of Engineering,* Vol. 12, No. 5, 729-741, doi: 10.24012/dumf.1051338.
[56] Altay, O. (2022). Chaotic slime mould optimization algorithm for global optimization, *Artificial Intelligence Review,* Vol. 55, 3979-4040, doi: 10.1007/s10462-021-10100-5.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 325–339
https://doi.org/10.14743/apem2025.3.543

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Optimizing abrasive water jet milling of alumina ceramics with RBF neural networks

Feng, Y.T.[a,*], **Shi, Z.R.**[a], **Yang, X.**[a], **Huang, W.**[a], **Luo, X.**[a], **Li, Y.H.**[a], **Yu, L.**[a]

[a]Geely University of China, Chengdu, P.R. China

ABSTRACT

Abrasive water jet technology is an advanced machining method that combines high-pressure water jet with solid abrasives. Owing to its unique cold-processing characteristics, high flexibility, and environmental bebefits, it has been widely applied in aerospace, medical devices, microelectronics, defense and other fields. Focusing on alumina ceramic plates, this study systematically investigates abrasive water jet (AWJ) milling through an integrated experimental and modeling approach. The research framework consists of three main phases: the development of an experimental design for abrasive water jet milling of alumina ceramics; systematic parameter optimization using single-factor and orthogonal array experiments, with material removal rate and milling depth as key performance indicators; and the application of a radial basis function (RBF) neural network model for milling depth prediction. The experimental results demonstrate that optimal parameter combinations improve machining efficiency by 38 % compared to baseline conditions. The developed RBF model achieves exceptional predictive accuracy, with maximum absolute and relative errors of 0.30 mm and 18.8 %, respectively, and a mean absolute error of 12.01 % across validation trials. This work provides a theoretical foundation for precision machining of advanced ceramics while demonstrating a viable pathway toward intelligent process optimization in AWJ technology.

ARTICLE INFO

Keywords:
Abrasive water jet (AWJ);
Milling;
Alumina ceramic;
Precision machining;
Material removal rate;
Single-factor experiment;
Orthogonal array;
RBF neural network

**Corresponding author:*
fengyetao917@163.com
(Feng, Y.T.)

Article history:
Received 16 June 2025
Revised 30 August 2025
Accepted 3 September 2025

1. Introduction

Abrasive Water jet (AWJ) technology is an advanced machining method that integrates high-pressure water jet with solid abrasives. Owing to its unique cold-processing characteristics, high flexibility, and environmental benefits, it has been widely adopted in aerospace, medical devices, microelectronics, and defense industries. As industrial materials evolve toward high hardness, high-temperature resistance, and brittleness, the limitations of conventional pure water jet technology have become increasingly apparent—such as the reliance on ultra-high pressure for machining hard materials (leading to prohibitive equipment costs) and low processing efficiency. By introducing abrasive particles into high-velocity water jets, AWJ significantly enhances jet penetration capability and material removal efficiency while avoiding thermal damage and surface deformation, making it ideal for processing complex materials (e.g., composites, titanium alloys, ceramics). However, challenges persist, including severe nozzle wear, high energy consumption, and substantial capital investment, particularly in high-precision machining of brittle materials like ceramics, where surface roughness control, material removal modeling, and process parameter optimization remain critical yet unresolved issues.

While AWJ technology has achieved notable progress in metal and composite machining, key bottlenecks remain in the efficient precision milling of high-hardness brittle ceramics such as alumina. First, the strong nonlinear coupling between brittle material removal mechanisms and process parameters cannot be adequately captured by traditional single-factor experiments. Second, surface integrity control and depth prediction in ceramic machining lack reliable models, as existing studies predominantly rely on empirical formulas with limited generalizability. Third, systematic process databases for AWJ milling of alumina ceramics remain underdeveloped, hindering industrial adoption. To address these challenges, this study focuses on alumina ceramic plates and employs a hybrid experimental-machine learning approach to overcome these limitations.

This study systematically investigates the AWJ milling process for alumina ceramics, with the following key innovations:

- By integrating single-factor and orthogonal array experiments, the dynamic influence hierarchy of five critical parameters (water jet pressure, abrasive flow rate, nozzle traverse speed, standoff distance, and lateral feed) on material removal rate (MRR) and milling depth is elucidated. For instance, the parameter hierarchy for MRR is identified as water jet pressure > abrasive flow > nozzle speed > standoff distance > lateral feed, transcending the constraints of conventional single-parameter optimization.
- An optimal parameter combination (water jet pressure: 260 MPa, abrasive flow: 240 g/min, standoff distance: 20 mm, etc.) is proposed for brittle alumina ceramics, demonstrating the feasibility of AWJ in precision ceramic machining and filling a critical gap in applied research.
- A nonlinear prediction model for milling depth is innovatively developed using a Radial Basis Function (RBF) neural network. Compared to traditional Backpropagation (BP) networks, the RBF model exhibits superior generalization capabilities on small datasets, achieving a maximum relative error of 17 % and a mean absolute error of 12.01 %, thereby advancing intelligent process optimization tools.
- This work establishes an integrated research framework of "experimental data acquisition → parameter optimization → RBF modeling → process validation" for AWJ milling of alumina ceramics. By synergizing experimental insights with artificial intelligence, it explores the influence mechanisms of multi-parameter interactions on machining performance and delivers a neural network-based depth prediction model. The outcomes provide theoretical foundations and technical solutions for efficient precision machining of ceramics, accelerating the evolution of AWJ technology toward intelligent and high-precision applications.

2. Related work

Abrasive Water Jet Machining has been extensively studied for its capability in precision processing diverse materials. In foundational machinability research, Alberdi *et al.* redefined the separation velocity criterion by evaluating the machinability index, revealing significantly higher Nm values for carbon fiber composites than metals with notable variations between composites [1]. Concurrently, Uthayakumar *et al.* identified water jet pressure as the most critical factor affecting surface morphology in nickel-based superalloys [2]. Regarding key process parameters, Khan and Haque demonstrated that abrasive hardness dictates kerf width and taper during glass machining [3], while Azmir and Ahsan's Taguchi analysis showed water pressure as the dominant factor for surface roughness and abrasive type for taper ratio in glass/epoxy composites [4]. Supriya and Srinivas emphasized that high water pressure and abrasive flow rate enhance cut depth and surface finish in stainless steel [5], and Begic-Hajdarevic *et al.* optimized parameter sets for aluminum balancing productivity and surface quality [6].

For quality prediction and improvement, Hlaváč *et al.* established a theoretical model linking striation deflection angle (θ) to predict maximum cutting speed, with head tilting improving wall quality [7]. Chen and Siores attributed striations to wavelike kinetic energy distribution of abrasives and developed an oscillation technique to increase smooth-zone depth by 30-40 % [8]. Chen *et al.* further applied forward head oscillation in ceramic cutting, noting 70-75 ° impact

angles eliminated trailing striations [9]. Yuvaraj and Kumar corroborated that a 70 ° jet angle better preserved surface integrity in AISI D2 steel versus conventional 90 ° [10].

Studies on specialty materials include research in which Perec identified olivine and crushed glass as cost-effective alternatives to garnet for titanium cutting [11], Niranjan *et al*. validated the applicability of abrasive waterjet machining for AZ91 magnesium alloy [12], Kong *et al*. demonstrated deep milling of NiTi shape-memory alloys [13], Hocheng and Chang derived a 2.5-power relationship between ceramic material removal rate and abrasive velocity, showing that fine abrasives combined with high pressure yield grinding-like surface finishes [14], and Zhu *et al*. achieved ductile-mode polishing of hard-brittle materials to nanoscale surface roughness under low pressure and shallow erosion angles [15].

Cost control strategies were advanced by Kantha Babu and Krishnaiah Chetty, who optimized garnet recycling for aluminum by removing sub-90 μm particles and replenishing at 60 % [16]. Emerging technologies encompass eco-friendly ice jet machining proposed by Gupta *et al.* [17], the optimization of gelatin-bound abrasive water suspension jet (AWSJM) machining by Patel and Tandon [18], Akkurt's geometric characterization of brass cutting fronts [19], Selvan and Raju's analysis of cast iron cutting under varying traverse speeds and standoff distances [20], and solutions for precision cutting of stainless and hardened steels with thicknesses exceeding 10 mm proposed by Hlaváč *et al.* [21].

Radial basis function (RBF) neural networks continue to evolve and demonstrate broad applicability across diverse domains. Notable advances include the Generalized Growing and Pruning RBF (GGAP-RBF) proposed by Huang *et al*., which introduces neuron significance to enable efficient structural adaptation and improved function approximation over existing sequential learning methods [22], and the multi-label RBF (ML-RBF) developed by Zhang, which employs per-class clustering and global weight optimization to capture label correlations, achieving higher accuracy and training speeds exceeding those of BP-MLL by more than two orders of magnitude [23].

Recent advancements in intelligent algorithms have demonstrated potent capabilities in modeling and optimizing complex engineering problems, extending far beyond the realm of AWJ technology. In the fields of localization and transportation, Li *et al.* proposed a UWB-INS fusion positioning approach based on an extended Kalman filter and a two-level error compensation model to improve localization accuracy in GNSS-denied environments. By combining inertial navigation data with UWB measurements and applying wavelet-based prefiltering, their method significantly enhanced positioning accuracy and stability, particularly during changes in vehicle motion states [24]. Leon-Medina *et al.* introduced a bio-inspired topology optimization approach based on bacterial chemotaxis to minimize structural compliance in non-derivative search spaces. The method demonstrated mesh-independent solutions and performance comparable to established evolutionary optimization techniques [25]. Yang *et al.* integrated the Sparrow Search Algorithm (SSA) with Backpropagation Neural Networks (BP), leveraging SSA's global search capability to optimize the BP network. This SSA-BP approach was successfully applied to cloud-based vehicle driving behavior recognition, achieving low identification error, a driving-behavior-based fuel consumption evaluation model with 89.216 % prior accuracy, and approximately 98 % regression fit, offering novel insights for intelligent driving and traffic safety management [26]. Shweta *et al.* developed a low-cost IoT-based monitoring framework for solar photovoltaic power plants using Sigfox LPWAN communication and sensor data analytics, combined with feature selection and deep learning techniques to detect faults and estimate potential energy and energy losses [27]. Lv *et al.* employed Reinforcement Learning to address the Flexible Job-shop Scheduling Problem with Parallel Operations, which captures the synchronization of multiple preceding operations in real manufacturing systems. Their approach is based on an Attention Restart mechanism integrated with Heterogeneous Graph Attention Networks to model complex relationships between operations and machines [28]. Song and Huo investigated manufacturing quality control using ultrasonic nondestructive testing data by integrating machine learning techniques, specifically a wavelet neural network optimized with genetic algorithms. Their approach improved defect prediction accuracy and quality stability evaluation compared to traditional wavelet neural network models, particularly in noisy and complex production environments [29].

In algorithm enhancement and cross-domain application, Chang proposed a revised Girvan-Newman clustering algorithm based on Item Response Theory (IRT) for forming cooperative groups in programming learning. By calculating learner ability and interpersonal relationship metrics, this method significantly improved learning outcomes in programming courses [30]. Zou *et al.* focused on architectural image processing, introducing a novel model integrating Restricted Boltzmann Machines (RBMs) and Convolutional Neural Networks (CNNs). This model achieved high-accuracy (97 %) and efficient 2D-to-3D conversion of architectural images, substantially improving measurement precision and reducing execution time [31]. These interdisciplinary studies underscore the significant potential of intelligent algorithms (e.g., NN, SVM, GA, PSO variants, DRL, GP) in modeling complex systems, optimizing parameters, and enhancing performance. Their successful methodologies and findings offer valuable insights and synergistic possibilities for the intelligent advancement of AWJ technology, particularly in areas like parameter optimization, quality prediction, and process control.

3. Research method

3.1 Experimental setup and materials

The experiment employed a five-axis ultra-high-pressure abrasive water jet system, which primarily consists of a high-pressure generator, water softening system, abrasive feeding system, control system, and nozzle assembly. The schematic diagram of this cutting experimental setup is shown in Fig. 1.

The milling workpiece material was YHL02 alumina ceramic plate, as shown in Fig. 2. The performance parameters of the alumina ceramic material are listed in Table 1.

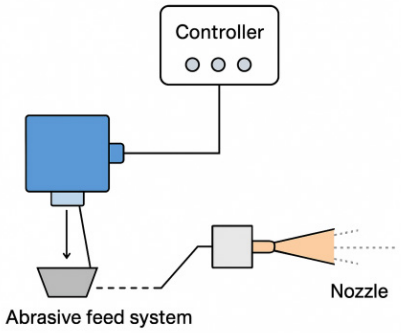


Fig. 1 High pressure generating system

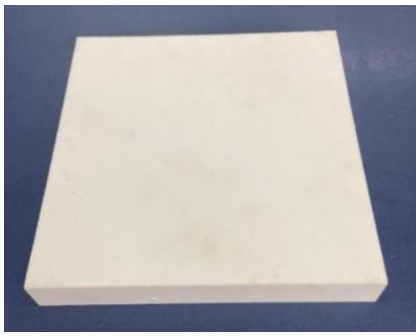


Fig. 2 Alumina ceramic plate sample

Table 1 Mechanical properties of workpiece

Experimental material	Content (%)	Vickers hardness (GPa)	Compressive strength (MPa)	Fracture toughness (MPa·m$^{1/2}$)	Density (g/cm^3)	Modulus of elasticity (GPa)
Al2O3	92	10.4	850	4.8	3.2	276

3.2 Experimental design for milling

The experimental procedure involved milling a groove (30 × 10 × 2 mm^3) in the alumina ceramic workpiece. Fig. 3 shows the programmed nozzle trajectory, using conventional 60-mesh garnet abrasive.

Considering experimental constraints, five dominant and controllable process parameters were investigated: jet pressure (MPa), nozzle traverse speed (mm/min), standoff distance (mm), stepover (mm), and abrasive mass flow rate (g/min). The parameter matrix for alumina ceramic milling is presented in Table 2.

The experimental setup includes (Fig. 4):

- workpiece fixturing arrangement,
- in-process milling demonstration.

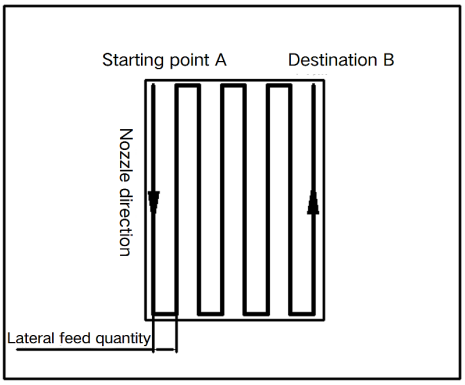


Fig. 3 The milling path

Table 2 Process parameters for milling alumina ceramics

Process parameter	Parameter level								
Jet pressure P (MPa)	220	230	240	250	260	270	280	290	300
Target range S (mm)	5	7	10	13	15	17	20	23	25
Nozzle velocity u (mm/min)	50	75	100	125	150	175	200	225	250
Lateral feed L (mm)	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
Abrasive flow m_a (g/min)	60	90	120	150	180	210	240	270	300

(a) (b)

Fig. 4 The milling experiment

The machining performance of abrasive water jet milling is evaluated by two key indicators: material volume removal rate and single-pass cutting depth. The average material removal rate (V) is calculated by dividing the total removed material volume by the milling time. The material removal volume, representing the volumetric amount removed in a single milling pass, can be determined by dividing the removed material mass by its density. The milling time is obtained through empirical formulas Eq. 1 [32].

$$T = \frac{bl}{\delta u} N_a \tag{1}$$

where T is milling time (s), l is length of milling plane (mm), δ is single-pass milling width (mm), u is nozzle traverse speed (mm/s), b is width of milling plane (mm), and N_a is number of milling passes. The depth of milling (h) was determined by averaging three caliper measurements.

3.3 The establishment of RBF

Radial basis function neural networks (RBFNNs) are well suited for predicting abrasive water jet milling depth due to their strong nonlinear mapping capability and efficient learning characteristics, particularly for small data sets. Its applicability is primarily demonstrated in abrasive water jet machining, where complex nonlinear coupling relationships exist between milling depth and process parameters such as jet pressure (P), nozzle moving speed (u), and others. The RBFNN locally transforms the input space using Gaussian functions in the hidden layer, enabling effective capture of nonlinear interactions between parameters and overcoming the limitations of traditional linear methods, such as polynomial regression. Moreover, given the limited amo-

unt of experimental data (45 data sets) and the dynamic neuron expansion mechanism implemented in the hidden layer of the RBFNN via the MATLAB *newrb* function, overfitting can be avoided by gradually increasing the number of neurons according to a predefined target error. Compared with the BP neural network, the RBFNN exhibits stronger generalization ability when applied to small data sets.

Experimental data source

The experimental system comprises 45 data sets obtained from single-factor abrasive water jet milling tests, covering the following parameter ranges: jet pressure P (220-300 MPa), nozzle moving speed u (50-250 mm/min), target distance S (5-25 mm), lateral feed L (0.6-1.0 mm), and abrasive flow rate m_a 60-300 g/min.

Data preprocessing

Min–Max normalization was applied using Eq. 2:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

To avoid data leakage, the normalization parameters derived from the training set were also applied to the test set. After normalization, all variables were scaled to the interval [0,1], which effectively improves the convergence speed of the network.

Validation data set

To verify the predictive capability of the model, nine validation experiments were conducted using different combinations of five process parameters. The corresponding measured milling depths are summarized in Table 3.

Table 3 Validation experimental data set for milling depth prediction

Argument / Serial number	P (MP)	S (mm)	u (mm/min)	L (mm)	m_a (g/min)	Actual milling depth
1	220	6	61.82	0.6	80	0.88
2	230	8	71.66	0.7	100	0.65
3	250	11	89.46	0.8	130	0.53
4	270	12	100.50	0.9	150	0.71
5	290	17	127.17	0.65	200	1.25
6	295	22	174.58	0.75	250	1.64
7	200	7	98.10	0.4	70	1.59
8	205	13	108.67	0.5	90	1.01
9	225	5	200	1.0	110	1.42

A Radial Basis Function Neural Network (RBFNN) adopts a three-layer feed forward structure, and its topological design strictly follows theoretical framework proposed by Broomhead and Lowe [33]. According to the five-dimensional input characteristics of abrasive water jet processing (jet pressure P, nozzle moving speed u, target distance S, lateral feed rate L, and abrasive flow rate m_a), the network is constructed as follows:

- Input layer: Five neurons corresponding to the standardized process parameter vector $X = [P, u, S, L, m_a]^{\mathrm{T}}$
- Hidden layer: The number of neurons is determined adaptively and generated iteratively using the MATLAB *newrb* function.
- Output layer: One linear neuron that outputs the predicted milling depth.

Among them, the activation function of the j-th neuron in the hidden layer is defined as a Gaussian radial basis function, as shown in Eq. (3):

$$\phi_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2\sigma_j^2}\right) \tag{3}$$

where c_j is the center of the j-th basis function, σ_j is the width parameter, and $||\cdot||$ denotes the Euclidean norm.

Key parameter optimization

The optimal parameter combination determined by grid search is shown in Table 4.

Table 4 Optimal parameter combination

Spread	1.0
Goal	0.001
Maximum number of neurons	45

For the performance evaluation of the model, the root mean square error (RMSE) and the coefficient of determination R^2 are employed. The root mean square error is defined in Eq. (4):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

The coefficient of determination is defined in Eq. (5):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5}$$

4. Results

4.1 Results of single milling experiment

The influence of injection pressure

During the experiment, the nozzle moving speed was 50 mm/min, the target distance was 10 mm, the lateral feed rate was 0.6 mm, and the abrasive flow rate was 120 g/min. The jet pressure varied from 220 to 300 MPa in increments of 10 MPa. Fig. 5 shows the effect of jet pressure P on the material removal rate and the average milling depth of alumina ceramics.

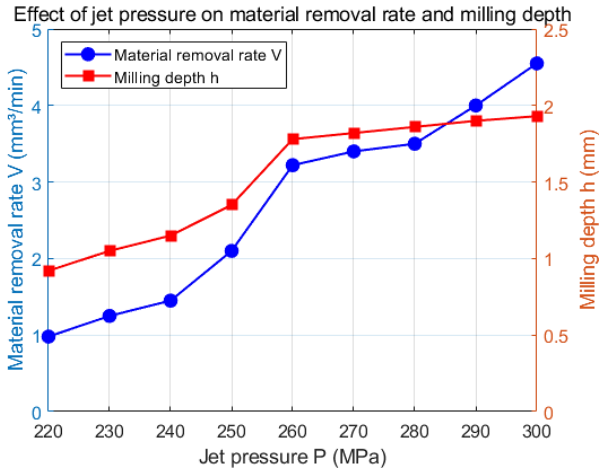


Fig. 5 Effect of the jet pressure P on the volume removal rate V and milling depth h of alumina ceramics

As can be seen from the figure, with the increase of jet pressure P, the material removal rate V presents nearly linear increase trend, and the jet pressure P is positively correlated with the material removal rate V, because the jet energy increases with the increase of jet pressure P, while the material removal rate V depends on the energy exerted on the workpiece surface per unit time. When removing materials, it is easier to reach the stress of material shedding, so the material removal rate is positively correlated with it.

The influence of the nozzle's moving speed

During the experiment, the jet pressure is 260 MPa, the target distance was 10 mm, the lateral feed rate was 0.6 mm, and the abrasive flow rate was 120 g/min. The nozzle moving speed was

varied from 50 to 250 mm/min in increments of 25 mm/min. Fig. 6 shows the effect of the nozzle moving speed u on the material removal rate and the average milling depth of alumina ceramics.

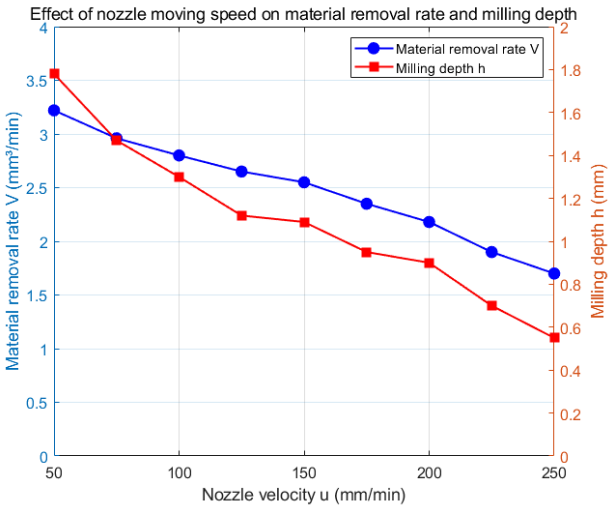


Fig. 6 Influence of the nozzle moving speed u on the volume removal rate V and milling depth h of alumina ceramics

As can be seen from the figure, as the nozzle moving speed u increases, the material removal rate V gradually decreases. This is because, at lower nozzle moving speeds, the interaction time between the jet and the workpiece material is longer, which facilitates material removal but reduces processing efficiency. Conversely, as the milling speed increases, the average impact energy of the jet acting on the workpiece decreases, resulting in a lower material removal rate.

The influence of the target distance

When the jet pressure was 260 MPa, the nozzle moving speed was 50 mm/min, the lateral feed rate was 0.6 mm, and the abrasive flow rate was 120 g/min, the target distance was varied from 5 to 25 mm (5, 7, 10, 13, 15, 17, 20, 23, and 25 mm). Fig. 7 shows the effect of the target distance S on the average material removal rate and milling depth of alumina ceramics.

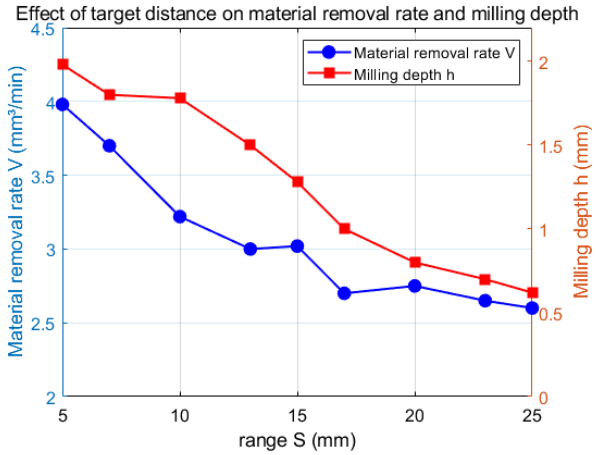


Fig. 7 Influence of the target distance S on the volume removal rate V and milling depth h of alumina ceramics

As can be seen from the figure, with the increase of target distance S, the material removal rate V decreases. This is because, with increasing target distance, the jet is increasingly affected by diffusion and air resistance, causing it to reach the workpiece surface in a more divergent form. As a result, the effective milling area increases, while the number of abrasive particles impacting the workpiece per unit area decreases, leading to a reduction in the kinetic energy transferred during impact. Consequently, the material stripping capability is weakened, resulting in a lower material removal rate.

The influence of lateral feed rate

The milling process usually requires multiple cuts to achieve the desired groove dimensions, and the lateral feed indicates the distance between the centers of adjacent abrasive water jets. According to the experimental data in Table 2, the jet pressure was 260 MPa, the nozzle moving speed was 50 mm/min, the target distance was 10 mm, and the abrasive flow rate was 120 g/min. The transverse feed rate L was varied from 0.6 to 1.0 mm (0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, and 1.0 mm). Fig. 8 shows the effect of the transverse feed rate L on the material removal rate and the average milling depth of alumina ceramics.

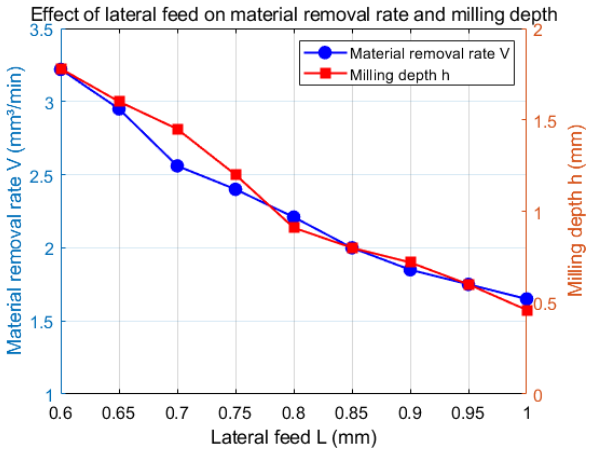


Fig. 8 Effect of the lateral feed L on the volume removal rate V and milling depth h of alumina ceramics

As can be seen from the figure, the material removal rate of alumina ceramics decreases with increasing lateral feed rate. This is because, as the transverse feed rate increases, the overlap area between adjacent cuts decreases, resulting in portions of material remaining unprocessed between neighboring tracks and, consequently, a lower material removal rate. However, when the lateral feed rate is too small, the material stripping efficiency is reduced and the milling depth becomes excessively large due to repeated milling. Therefore, an appropriate lateral feed rate should be selected.

The influence of abrasive flow rate

When the jet pressure was 260 MPa, the nozzle moving speed was 50 mm/min, the lateral feed rate was 0.6 mm, and the target distance was 10 mm, the abrasive flow rate was varied from 60 to 300 g/min (60, 90, 120, 150, 180, 210, 240, 270, and 300 g/min). Fig. 9 shows the effect of the abrasive flow rate on the volume removal rate and the average milling depth of alumina ceramics.

As can be seen from the figure, the material removal rate increases with increasing abrasive flow rate. This is because, as the abrasive flow rate increases, the number of abrasive particles per unit volume of the jet also increases, leading to a higher number of particles participating in the erosion process. Consequently, the frequency of abrasive impacts on the workpiece surface increases. Since material removal is primarily governed by abrasive particle erosion, a higher abrasive flow rate results in an increased material removal rate.

Based on the above single-factor experimental results, an optimal parameter combination was identified. Under the optimal conditions (jet pressure of 260 MPa, abrasive flow rate of 240 g/min, and nozzle moving speed of 50 mm/min), a surface roughness Ra of 3.2 μm was achieved, representing a reduction of 28 % compared with the baseline condition. This improvement also led to an estimated reduction of post-processing time by approximately 20 %.

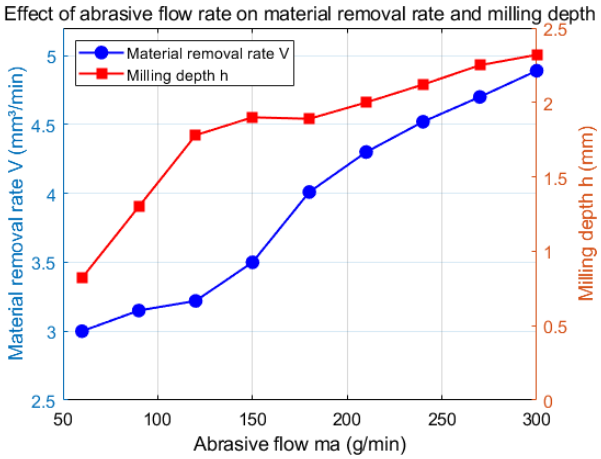


Fig. 9 Effect of the abrasive flow rate m_a on the volume removal rate V and milling depth h of alumina ceramics

4.2 Results of orthogonal experiment

Several easily controllable parameters, such as jet pressure, nozzle moving speed, and abrasive flow rate, have a significant influence on machining performance when considered individually. However, in actual production and processing, these process parameters must be analyzed comprehensively. Therefore, an orthogonal experimental method is employed to further optimize the five process parameters and identify the optimal combination of machining conditions.

In this experiment, jet pressure P, nozzle moving speed u, lateral feed rate L, target distance S, and abrasive flow rate m_a were selected as the process parameters. Four levels were defined for each factor. Since all factors had the same number of levels, an L16 (4^5) orthogonal array with five factors and four levels was employed, resulting in a total of 16 experiments. The experimental factors and their corresponding levels are listed in Table 5, while the orthogonal experimental design and results are presented in Table 6.

Table 5 Test factors and levels

Factor level	Jet pressure P(MPa)	Abrasive flow m_a (g/min)	Target Range S(mm)	Lateral feed L (mm)	Nozzle moving speed u (mm/min)
1	240	60	10	0.6	50
2	260	120	15	0.7	100
3	280	180	20	0.8	150
4	300	240	25	0.9	200

Table 6 Orthogonal experiment table and experiment results

Number	Experimental parameter					Experimental result	
	A	B	C	D	E		
	P (MPa)	m_a (g/min)	S (mm)	L (mm)	u (mm/min)	Volume removal rate V (mm³/s)	Milling depth h(mm)
1	240	60	10	0.6	50	1.12	1.25
2	240	120	15	0.7	100	0.78	1.18
3	240	180	20	0.8	150	0.85	1.20
4	240	240	25	0.9	200	0.64	1.14
5	260	60	15	0.8	200	2.05	1.62
6	260	120	10	0.9	150	2.25	1.73
7	260	180	25	0.6	100	3.58	2.15
8	260	240	20	0.7	50	3.89	2.52
9	280	60	20	0.9	100	1.85	1.47
10	280	120	25	0.8	50	2.15	1.58
11	280	180	10	0.7	200	1.96	1.51
12	280	240	15	0.6	150	2.43	1.92
13	300	60	25	0.7	150	1.52	1.50
14	300	120	20	0.6	200	2.78	1.72
15	300	180	15	0.9	50	3.54	1.93
16	300	240	10	0.8	100	4.35	2.18

The optimized parameter ranges identified in this study (e.g., jet pressure of 260-300 MPa and abrasive flow rate of up to 240 g/min) are compatible with commercially available five-axis abrasive water jet systems. Furthermore, an industrial case study conducted at a ceramics manufacturing facility in Sichuan, China, demonstrated that applying the optimized parameters resulted in an approximately 12 % improvement in processing efficiency, indicating good industrial scalability of the proposed optimization strategy.

4.3 RBF prediction results

By running the RBF prediction code in MATLAB, as shown in Fig. 10, nine independent test data sets were obtained for model validation, yielding an RMSE of 0.1569 and an R^2 value of 0.8404. The results indicate that the RBF neural network is capable of accurately fitting the abrasive water jet milling depth, with an average absolute error of 12.01 %. The maximum absolute error occurred in validation sample No. 7 and reached 0.3. The maximum relative error occurred in validation sample No. 2, with a value of 17 %.

For comparison, support vector machine (SVM) and random forest regression (RFR) models were also evaluated using the same data set. The RBF model achieved an RMSE of 0.1569 and an R^2 value of 0.8404, compared with RMSE values of 0.1925 and 0.2013 and R^2 values of 0.7812 and 0.7545 for the SVM and RFR models, respectively, indicating the superior predictive performance of the RBF model.

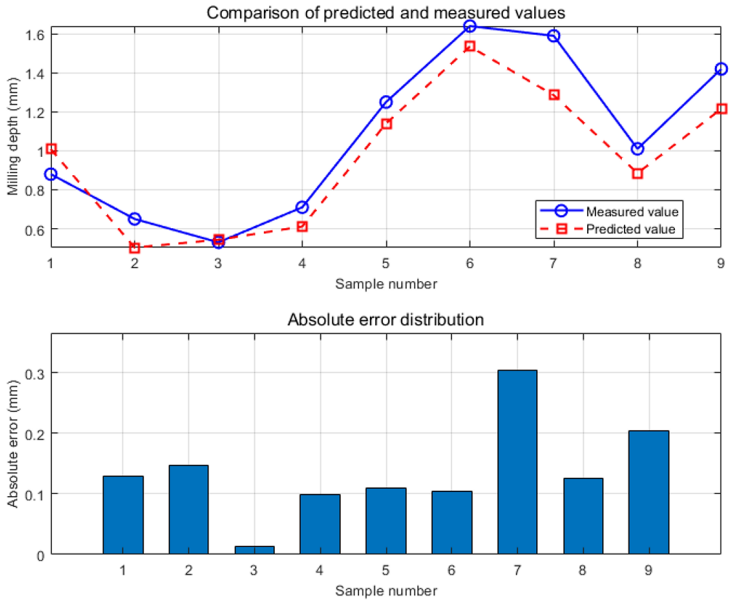


Fig. 10 RBF prediction results and absolute errors

5. Discussion

5.1 Discussion on a single milling experiment

The experimental results reveal systematic relationships between key process parameters and milling depth *h* during abrasive water jet (AWJ) milling of alumina ceramics. These dependencies are governed by fundamental energy transfer and material removal mechanisms.

Jet pressure P

The milling depth *h* exhibits a strong positive correlation with the jet pressure *P*. Increasing *P* elevates the kinetic energy of the jet, enabling more effective material erosion at greater depths within the workpiece. Higher energy density facilitates overcoming the material's fracture toughness at subsurface layers. However, excessively high *P* imposes greater demands on the ultra-high-pressure intensifier system and significantly increases energy consumption. Optimal pressure selection is therefore critical to balance machining efficiency with equipment longevity and operational costs.

Nozzle traverse speed u

An inverse relationship exists between traverse speed u and the milling depth h. Higher traverse speeds reduce the dwell time of the jet over any given point on the workpiece, thereby decreasing the total impact energy delivered per unit area per unit time. Consequently, material removal primarily occurs near the surface, with insufficient energy reaching deeper regions to cause effective fracture. The reduced interaction time ultimately limits penetration depth.

Standoff distance S

An increase in the standoff distance S leads to a reduction in the milling depth h. As the jet travels a longer distance through air, aerodynamic drag causes jet dispersion and attenuation of kinetic energy. The expanding jet profile reduces particle density and impact velocity upon reaching the workpiece surface, thereby diminishing its ability to erode material effectively at greater depths. Notably, beyond a critical standoff distance (approximately 20 mm in this study), jet energy dissipation reaches a level at which further increases in S result in negligible changes in both the material removal rate V and the milling depth h. This behavior indicates that the jet enters a regime where its residual energy falls below the threshold required for significant material removal in the hard alumina substrate.

Lateral feed L

The milling depth h is inversely related to the lateral feed L. A smaller L increases the overlap between successive jet passes, subjecting the material within the overlap zone to repeated impacts and thereby increasing the cumulative erosion depth in these regions. In contrast, a larger L reduces the overlap between adjacent passes, leaving valleys of unprocessed or minimally processed material between milled tracks, which ultimately leads to a reduction in the overall milling depth across the machined area.

Abrasive mass flow rate m_a

The milling depth h increases with increasing abrasive mass flow rate m_a. A higher abrasive flow rate introduces a greater number of abrasive particles into the jet stream per unit time, which increases the frequency of particle impacts on the workpiece surface and subsurface per unit area. The cumulative kinetic energy transferred by the increased particle flux generates higher normal and shear stresses within the material, enabling the fracture threshold to be exceeded over a larger volume. As a result, deeper penetration and enhanced material removal are achieved.

Overall, the observed dependencies indicate that the milling depth h is predominantly governed by the effective kinetic energy density delivered to the workpiece subsurface. Parameters such as jet pressure P and abrasive mass flow rate m_a directly increase this energy density, thereby promoting deeper erosion. In contrast, nozzle traverse speed u, standoff distance S, and lateral feed L act to reduce or redistribute the delivered energy density—u by limiting exposure time, S by dissipating energy prior to impact, and L by spatially distributing the energy input across the machined area. The identified saturation behavior with respect to S highlights the importance of maintaining an optimal standoff distance to maximize jet efficiency. These findings provide a clear mechanistic basis for the parameter optimization achieved through orthogonal experiments and support the physical validity of the predictive relationships captured by the RBF neural network model.

From an industrial perspective, high-energy process parameters increase operational costs. Operating at 300 MPa consumes approximately 0.22 kWh per minute, adding about 0.03-0.04 USD per minute and reducing nozzle service life by 18-22 % compared with 250 MPa. Abrasive consumption dominates the total cost (65-75 %), followed by energy (15-20 %) and nozzle wear (8-10 %), indicating the importance of balancing performance and cost efficiency.

Nozzle wear under high-pressure conditions can be mitigated through appropriate material selection. Tungsten carbide nozzles extend service life by about 40 %, diamond-lined nozzles reduce wear by up to 60 %, and coarser abrasives (e.g., 80 mesh) lower impact stress by approximately 20 %, improving nozzle durability.

5.2 Discussion on orthogonal experiment results

For the analysis of orthogonal experimental data, the range analysis method is employed to evaluate the relative influence of each process parameter, as defined in Eq. 6:

$$R = \max(T_1, T_2, T_3, T_4) - \min(T_1, T_2, T_3, T_4) \quad (6)$$

Based on the results presented in Table 7, the influence of the experimental factors on the material removal rate can be ranked as follows: jet pressure P > abrasive flow rate m_a > nozzle traverse speed u > standoff distance S > lateral feed L. The optimal parameter combination for maximizing the material removal rate is identified as $A_4B_4C_1D_1E_1$, corresponding to a jet pressure of 300 MPa, an abrasive flow rate of 240 g/min, a standoff distance of 10 mm, a lateral feed rate of 0.6 mm, and a nozzle traverse speed of 50 mm/min.

Similarly, according to the range analysis results in Table 8, the influence of the experimental factors on milling depth can be ordered as follows: jet pressure P > abrasive flow rate m_a > nozzle traverse speed u > lateral feed L > standoff distance S. The optimal parameter combination for maximizing the milling depth is $A_2B_4C_3D_1E_1$, corresponding to a jet pressure of 260 MPa, an abrasive flow rate of 240 g/min, a standoff distance of 20 mm, a lateral feed rate of 0.6 mm, and a nozzle traverse speed of 50 mm/min.

From a practical manufacturing standpoint, operating at high jet pressures increases process costs. Operating at a pressure of 300 MPa requires a power input of approximately 13.2 kW, adds about 0.03-0.04 USD per minute, and reduces nozzle life by 18-22 % compared with 250 MPa. Abrasives account for 65-75 % of total costs, followed by energy (15-20 %) and nozzle wear (8-10 %), emphasizing the need to balance performance and cost efficiency.

Table 7 Range analysis of experimental results for the material removal rate V in abrasive water jet single-pass milling

Factor / Index	A	B	C	D	E
	P (MPa)	m_a (g/min)	S (mm)	L (mm)	u (mm/min)
T1	3.39	6.54	9.68	9.91	10.70
T2	11.77	7.96	8.80	8.15	10.56
T3	8.39	9.93	9.37	9.40	7.05
T4	12.19	11.31	7.89	8.28	7.43

Table 8 Range analysis of experimental results for single-pass milling depth h in abrasive water jet machining

Factor / Index	A	B	C	D	E
	P (MPa)	m_a (g/min)	S (mm)	L (mm)	u (mm/min)
T1	0.77	1.84	2.63	3.04	3.28
T2	4.02	2.21	2.65	2.71	2.98
T3	2.48	2.79	2.91	2.58	2.35
T4	3.33	3.76	2.37	2.27	1.99

5.3 Discussion on RBF prediction

With jet pressure, nozzle moving speed, target distance, lateral feed rate, and abrasive particle size serving as the inputs of the RBF neural network, and milling depth as the output, the optimal parameter combination was determined using a grid search algorithm. Based on the experimental data, an RBF neural network model for predicting milling depth was established, yielding predicted milling depth values. The maximum absolute error compared with the measured milling depth is 0.3 mm, while the maximum relative error is 17 %. Furthermore, the milling depth can be effectively controlled by adjusting the jet pressure, nozzle moving speed, target distance, lateral feed rate, and abrasive particle size.

In industrial applications, accurate prediction of milling depth is particularly important under high-pressure operating conditions. Operating at a jet pressure of 300 MPa requires a power consumption of approximately 13.2 kW and reduces nozzle service life by 18-22 % compared with 250 MPa. Therefore, reliable RBF-based prediction can assist in selecting appropriate process parameters to reduce unnecessary energy consumption and tool wear while maintaining machining performance.

6. Conclusion

Through an experimental study of abrasive water jet milling of alumina ceramic sheets, the following conclusions are drawn:

- Systematic single-factor and orthogonal experiments revealed the influence hierarchy of five key parameters (jet pressure P, nozzle speed u, standoff distance S, transverse feed L, and abrasive flow rate m_a) on machining performance. For the material removal rate V, the dominant parameters followed the order $P > m_a > u > S > L$, with optimal settings of P = 300 MPa, m_a = 240 g/min, S = 10 mm, L = 0.6 mm, and u = 50 mm/min. For the milling depth h, the hierarchy shifted to $P > m_a > u > L > S$, with optimal parameters of P = 260 MPa, m_a = 240 g/min, S = 20 mm, L = 0.6 mm, and u = 50 mm/min.
- A radial basis function neural network (RBFNN) model implemented in MATLAB demonstrated reliable predictive capability for milling depth under multi-parameter conditions. Validation using nine independent experimental cases yielded a maximum relative error of 17 % (corresponding to a maximum absolute error of 0.3 mm) and a mean absolute error of 12.01 %, confirming the feasibility and effectiveness of neural-network-based modeling for nonlinear abrasive water jet machining processes.

References

[1] Alberdi, A., Suárez, A., Artaza, T., Escobar-Palafox, G.A., Ridgway, K. (2013). Composite cutting with abrasive water jet, *Procedia Engineering*, Vol. 63, 421-429, doi: 10.1016/j.proeng.2013.08.217.

[2] Uthayakumar, M., Khan, M.A., Kumaran, S.T., Slota, A., Zajac, J. (2016). Machinability of nickel-based superalloy by abrasive water jet machining, *Materials and Manufacturing Processes*, Vol. 31, No. 13, 1733-1739, doi: 10.1080/10426914.2015.1103859.

[3] Khan, A.A., Haque, M.M. (2007). Performance of different abrasive materials during abrasive water jet machining of glass, *Journal of Materials Processing Technology*, Vol. 191, No. 1-3, 404-407, doi: 10.1016/j.jmatprotec.2007.03.071.

[4] Azmir, M.A., Ahsan, A.K. (2009). A study of abrasive water jet machining process on glass/epoxy composite laminate, *Journal of Materials Processing Technology*, Vol. 209, No. 20, 6168-6173, doi: 10.1016/j.jmatprotec.2009.08.011.

[5] Supriya, S.B., Srinivas, S. (2018). Machinability studies on stainless steel by abrasive water jet – Review, *Materials Today: Proceedings*, Vol. 5, No. 1, Part 3, 2871-2876, doi: 10.1016/j.matpr.2018.01.079.

[6] Begic-Hajdarevic, D., Cekic, A., Mehmedovic, M., Djelmic, A. (2015). Experimental study on surface roughness in abrasive water jet cutting, *Procedia Engineering*, Vol. 100, 394-399, doi: 10.1016/j.proeng.2015.01.383.

[7] Hlaváč, L.M., Hlaváčová, I.M., Gembalová, L., Kaličinský, J., Fabian, S., Měšťánek, J., Kmec, J., Mádr, V. (2009). Experimental method for the investigation of the abrasive water jet cutting quality, *Journal of Materials Processing Technology*, Vol. 209, No. 20, 6190-6195, doi: 10.1016/j.jmatprotec.2009.04.011.

[8] Chen, F.L., Siores, E. (2001). The effect of cutting jet variation on striation formation in abrasive water jet cutting, *International Journal of Machine Tools and Manufacture*, Vol. 41, No. 10, 1479-1486, doi: 10.1016/S0890-6955(01)00013-X.

[9] Chen, L., Siores, E., Wong, W.C.K. (1998). Optimising abrasive waterjet cutting of ceramic materials, *Journal of Materials Processing Technology*, Vol. 74, No. 1-3, 251-254, doi: 10.1016/S0924-0136(97)00278-1.

[10] Yuvaraj, N., Kumar, M.P. (2016). Surface integrity studies on abrasive water jet cutting of AISI D2 steel, *Materials and Manufacturing Processes*, Vol. 32, No. 2, 162-170, doi: 10.1080/10426914.2016.1221093.

[11] Perec, A. (2018). Experimental research into alternative abrasive material for the abrasive water-jet cutting of titanium, *The International Journal of Advanced Manufacturing Technology*, Vol. 97, 1529-1540, doi: 10.1007/s00170-018-1957-2.

[12] Niranjan, C.A., Srinivas, S., Ramachandra, M. (2018). Effect of process parameters on depth of penetration and topography of AZ91 magnesium alloy in abrasive water jet cutting, *Journal of Magnesium and Alloys*, Vol. 6, No. 4, 366-374, doi: 10.1016/j.jma.2018.07.001.

[13] Kong, M.C., Axinte, D., Voice, W. (2011). Challenges in using waterjet machining of NiTi shape memory alloys: an analysis of controlled-depth milling, *Journal of Materials Processing Technology*, Vol. 211, No. 6, 959-971, doi: 10.1016/j.jmatprotec.2010.12.015.

[14] Hocheng, H., Chang, K.R. (1994). Material removal analysis in abrasive waterjet cutting of ceramic plates, *Journal of Materials Processing Technology*, Vol. 40, No. 3-4, 287-304, doi: 10.1016/0924-0136(94)90456-1.

[15] Zhu, H.T., Huang, C.Z., Wang, J., Li, Q.L., Che, C.L. (2009). Experimental study on abrasive waterjet polishing for hard-brittle materials, *International Journal of Machine Tools and Manufacture*, Vol. 49, No. 7-8, 569-578, doi: 10.1016/j.ijmachtools.2009.02.005.

[16] Kantha Babu, M., Krishnaiah Chetty, O.V. (2002). Studies on recharging of abrasives in abrasive water jet machining, *The International Journal of Advanced Manufacturing Technology*, Vol. 19, 697-703, doi: 10.1007/s001700200115.
[17] Gupta, K., Avvari, M., Mashamba, A., Mallaiah, M. (2017). Ice jet machining: A sustainable variant of abrasive water jet machining, In: Davim, J. (eds.), *Sustainable Machining*, Springer, Cham, Switzerland, 67-78, doi: 10.1007/978-3-319-51961-6_4.
[18] Patel, D., Tandon, P. (2017). Experimental investigations of gelatin-enabled abrasive water slurry jet machining, *The International Journal of Advanced Manufacturing Technology*, Vol. 89, 1193-1208, doi: 10.1007/s00170-016-9154-7.
[19] Akkurt, A. (2010). Cut front geometry characterization in cutting applications of brass with abrasive water jet, *Journal of Materials Engineering and Performance*, Vol. 19, 599-606, doi: 10.1007/s11665-009-9513-8.
[20] Selvan, M.C.P., Raju, N.M.S. (2012). Analysis of surface roughness in abrasive waterjet cutting of cast iron, *International Journal of Science, Environment and Technology*, Vol. 1, No. 3, 174-182.
[21] Hlaváč, L.M., Hlaváčová, I.M., Geryk, V., Plančár, Š. (2015). Investigation of the taper of kerfs cut in steels by AWJ, *The International Journal of Advanced Manufacturing Technology*, Vol. 77, 1811-1818, doi: 10.1007/s00170-014-6578-9.
[22] Huang, G.-B., Saratchandran, P., Sundararajan, N. (2005). A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, *IEEE Transactions on Neural Networks*, Vol. 16, No. 1, 57-67, doi: 10.1109/TNN.2004.836241.
[23] Zhang, M.-L. (2009). ML-RBF: RBF neural networks for multi-label learning, *Neural Processing Letters*, Vol. 29, 61-74, doi: 10.1007/s11063-009-9095-3.
[24] Li, Z., Zhang, Y., Shi, Y., Yuan, S., Zhu, S. (2023). Performance enhancement of INS and UWB fusion positioning method based on two-level error model, *Sensors,* Vol. 23, No. 2, Articel No. 557, doi: 10.3390/s23020557.
[25] Leon-Medina, J.X., Giraldo, J.F., Guzmán, M.A., Villalba-Morales, J.D. (2025). A bacterial chemostatic-based bio-inspired algorithm for the structural optimization of planar structures, *Arabian Journal for Science and Engineering*, 1-21, doi: 10.1007/s13369-025-10749-y.
[26] Yang, X., Xiang, K., Yuan, S., Huang, J. (2024). Vehicle driving behavior recognition and optimization strategies based on cloud computing and SSA-BP algorithm, *Studies in Informatics and Control*, Vol. 33, No. 3, 17-28, doi: 10.24846/v33i3y202402.
[27] Shweta, R., Sivagnanam, S., Kumar, K.A. (2023). IoT-based Deep Learning Neural Network (DLNN) algorithm for voltage stability control and monitoring of solar power generation, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 447-461, doi: 10.14743/apem2023.4.484.
[28] Lv, Q.H., Chen, J., Chen, P., Xun, Q.F., Gao, L. (2024). Flexible Job-shop Scheduling Problem with parallel operations using Reinforcement Learning: An approach based on Heterogeneous Graph Attention Networks, *Advances in Production Engineering & Management*, Vol. 19, No. 2, 157-181, doi: 10.14743/apem2024.2.499.
[29] Song, W.T., Huo, L. (2024). Machine learning for enhancing manufacturing quality control in ultrasonic nondestructive testing: A Wavelet Neural Network and Genetic Algorithm approach, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 347-357, doi: 10.14743/apem2024.3.511.
[30] Chang, W.-C. (2024). A revised Girvan-Newman clustering algorithm for cooperative groups detection in programming learning, *Computer Science and Information Systems*, Vol. 21, No. 2, 491-505, doi: 10.2298/CSIS220830069C.
[31] Zou, Q., Liu, F., Liao, Y. (2024). Enhancing architectural image processing: A novel 2D to 3D algorithm using improved convolutional neural networks, *Computer Science and Information Systems*, Vol. 21, No. 4, 1457-1481, doi: 10.2298/CSIS230725043Z.
[32] Feng, Y. (2007). *Research on machining technology of abrasive water jet milling ceramic materials* (in Chinese), PhD thesis, Shandong University, China.
[33] Broomhead, D.S., Lowe, D. (1988). Multivariable functional interpolation and adaptive networks, *Complex Systems*, Vol. 2, 321-355.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 340–350
https://doi.org/10.14743/apem2025.3.544

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Optimizing cooperation strategies for new energy vehicle manufacturers and technology suppliers: A game theory approach

Wang, Y.L.[a,b], **Chen, J.H.**[b], **Song, M.L.**[b], **Tang, F.**[b,*]

[a]Research Center for Enterprise Management, Chongqing Technology and Business University, Chongqing, P.R. China
[b]School of Business Administration, Chongqing Technology and Business University, Chongqing, P.R. China

ABSTRACT

New energy vehicle manufacturers usually choose to cooperate with technology suppliers to improve their products' market competitiveness. For this purpose, this paper constructs a single and dual sales-channel supply chain system comprising a technology supplier and a new energy vehicle manufacturer. Within this framework, the technology supplier acts as the leader in a Stackelberg game, while the new energy vehicle manufacturer acts as the follower. Based on intelligent driving cooperation cases between Huawei and Chang'an, BYD, and Seres, three cooperation modes—"technology introduction + own sales channel (Model A)", "cooperative R&D + own sales channel (Model B)", and "technology introduction + channel support (Model C)"—are examined to analyze cooperation mode selection strategies for the new energy vehicle manufacturer and the technology supplier. The study found that when the profit ratio of the technology supplier to the new energy vehicle manufacturer (referred to as the revenue sharing coefficient) meets certain conditions, the optimal cooperation mode of the new energy vehicle manufacturer and technology supplier is Model C. Additionally, the profit obtained by the technology supplier under Model A is always the smallest, while the profit size between the other two models depends on the revenue sharing coefficient. Moreover, five situations describe the profit outcomes of the new energy vehicle manufacturer across the three cooperation modes, which are affected by the revenue sharing coefficient and the proportion of the technology supplier's R&D investment cost. This study addresses a gap in research on cooperation modes within the new energy vehicle sector, and the conclusions obtained can provide valuable theoretical insights for new energy vehicle manufacturers and technology suppliers when selecting cooperation strategies.

ARTICLE INFO

Keywords:
New energy vehicle manufacturers;
Technology suppliers;
Cooperation modes;
Stackelberg game;
R&D investment;
Supply chain management;
Game theory;
Electric vehicle industry

**Corresponding author:*
tangfei@ctbu.edu.cn
(Tang, F.)

Article history:
Received 21 January 2025
Revised 3 June 2025
Accepted 17 June 2025

1. Introduction

According to the *Global Electric Vehicle Outlook 2022* report released by the International Energy Agency, global electric vehicle sales are expected to reach 6.6 million in 2022, representing a year-on-year increase of 100 % and accounting for 9 % of global passenger vehicle sales. Among them, China is the world's largest electric vehicle market, with sales reaching 3.39 million units—a year-on-year increase of 72 %, accounting for 51 % of global electric vehicle sales. At the same time, as more new energy vehicle companies (such as Tesla, Seres, BYD, Weilai, and Xiaopeng) continue to join, market competition has become increasingly fierce. In practice, to improve market competitiveness, many new energy vehicle manufacturers have chosen to conduct business cooperation with relevant technology suppliers with technological advantages, such as Chang'an, BYD, and Seres

with Huawei. At the same time, diverse cooperation model options exist for new energy vehicle manufacturers and technology suppliers. For example, Huawei provides Chang'an with autonomous driving solutions and in-vehicle communication technology cooperation mode for "technology introduction + own sales channel (Model A)"; BYD and Huawei have joined forces to promote the "cooperative R&D + own sales channel (Model B)" cooperation mode of intelligent driving technology innovation, and Huawei provides Seres with the "technology introduction + channel support (Model C)" cooperation mode to offer intelligent travel solutions and sales channels.

Model A refers to the cooperation mode in which new energy vehicle manufacturers only have their own sales channels and purchase technology suppliers' products at the wholesale price per unit. Model B refers to the cooperation mode in which new energy vehicle manufacturers only have their own sales channels and jointly conduct technology R&D with technology suppliers. Model C is a cooperation mode in which technology suppliers provide sales channels and related technical support for new energy vehicle manufacturers in exchange for revenue sharing from new energy vehicle manufacturers. Thus, which cooperation mode is optimal choice for both? What factors affect the choice of cooperation between the two? These are important practical issues that need to be explored in depth.

As an effective means to quickly make up for the lack of advanced technology, the academic and practical circles have been widely concerned with technology introduction. For example, Veugelers and Cassiman [1] took Belgian manufacturing enterprises as research objects and found that small innovative companies are more likely to adopt a single independent innovation or technology introduction strategy, while large companies are more inclined to combine the two. Cassiman and Veugelers [2] indicated that independent innovation and technology introduction are complementary, and the degree of complementarity is affected by corporate strategic environment and other factors. Guo *et al.* [3] studied the impact of analyst coverage on enterprises' innovation strategies and found that the information role of analysts leads companies to be more willing to acquire more innovative companies to introduce technology. Zhang *et al.* [4] analyzed the choice between the independent and technology introduction innovation of high energy-consuming enterprises under the goal of "dual control of energy consumption," and the results showed that technology introduction is the main way for high energy-consuming enterprises to achieve technological innovation. Wang *et al.* [5] discussed the internal mechanism of US economic sanctions on the technology introduction and independent R&D of enterprises in target countries and concluded that this impact is heterogeneous. Xie and Zhang [6] considered the influence of performance conditions of performance-based equity incentive exercise on enterprises' choice of an independent or technology introduction innovation strategy.

Collaborative R&D can shorten the R&D cycle and reduce costs and the uncertainty risk of R&D output through technical communication and knowledge sharing, thus enhancing enterprises' competitiveness [7]; this has attracted extensive attention from many scholars. For example, Cellini and Lambertini [8] studied the Cournot duopoly process innovation dynamic R&D problem and found that private and social incentives for collaborative R&D are consistent at all acceptable levels of technological spillovers that characterize innovation activities. Gupta [9] analyzed the impact of R&D spillover effect on manufacturers' choice of cooperation mode and R&D investment in process innovation. Petrakis and Tsakas [10] studied the impact of potential entrants on collaborative R&D among incumbent firms. Zheng *et al.* [11] analyzed the realization conditions and influence mechanism of the evolutionary stability strategy of cross-organizational cooperative R&D of general technology from a micro level. Zheng *et al.* [12] studied the cooperative R&D mode of general technology and its influencing factors from the perspective of technology chain. Zhou *et al.* [9] explored the motivation of R&D cooperation between leading firms and followers under the presence of technology spillover, as well as the impact of R&D cooperation on firm competitiveness, industry total R&D investment, and social welfare. Yan *et al.* [13] analyzed the evolution law of cooperative innovation behavior and the influence mechanism of heterogeneity of external financing mode under the condition of technology monopoly. Other scholars have discussed the payment structure of cross-organizational cooperative R&D in the supply chain [14].

To sum up, the studies on technology introduction and cooperative R&D mode are already rich, but none have taken new energy vehicle manufacturers as research objects. Except [7], few have

involved the selection of cooperation mode. Additionally, existing research on new energy vehicle manufacturers has mainly focused on the recycling and utilization of new energy vehicle power batteries [15-17], subsidy policies for new energy vehicles [18] and innovation of new energy vehicle manufacturers [19]. Different from the existing studies, this paper takes the new energy vehicle manufacturers Chang'an, BYD, and Seres, and the technology supplier Huawei as the research objects, considers three cooperation models (Models A, B, and C), and discusses the choice of cooperation modes between new energy vehicle manufacturers and technology suppliers.

2. Problem description and notation

Consider a supply chain system involving a technology supplier and a new energy vehicle manufacturer, which can operate through either a single sales channel or dual sales channels. In this system, the technology supplier sells the R&D product to the new energy vehicle manufacturer at the unit wholesale price w, and the new energy vehicle manufacturer then sells the manufactured car to the market, where the retail price of the technology supplier's R&D product is assigned to p (referred to as retail price p). To further improve the level of product technology R&D investment and stimulate market demand, new energy automobile enterprises can choose to conduct technology R&D cooperation with the technology supplier, and the technology supplier can provide sales channel support for the new energy vehicle manufacturer to a certain extent. Additionally, referring to the studies of Wang *et al.* [20, 21], Duan *et al.* [22], we assume that the product's demand function is $D = a - bp + \lambda e$, where arameter a is the potential market size, b is the sensitivity coefficient of consumers to market price p; λ is the sensitivity coefficient of consumers to the level of technology R&D investment e, and the cost of technology R&D investment is ke^2 [20, 21].

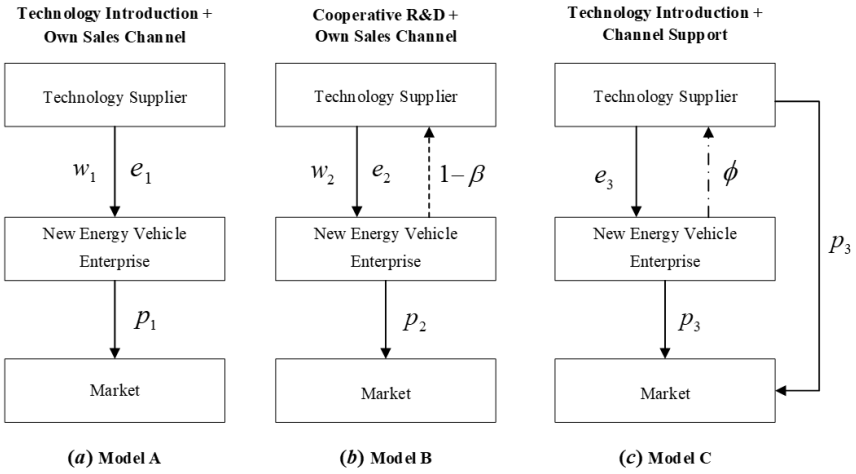


Fig. 1 Different cooperation modes between the new energy vehicle manufacturer and the technology supplier

This paper specifically considers three modes of cooperation between the new energy vehicle manufacturer and the technology supplier: (1) Cooperation Model A is the cooperation mode in which the new energy vehicle manufacturer only has its own sales channels and purchases the technology supplier's products at a unit wholesale price w_1 (see Fig. 1a), while the technology supplier independently conducts technology R&D and bears the R&D cost ke^2, as in the case of Chang'an and Huawei; (2) Cooperation Model B is the cooperation mode in which the new energy vehicle manufacturer has only its own sales channels and jointly conducts technology R&D with the technology supplier, in which the new energy vehicle manufacturer shares a proportion $(1-\beta)$ of the technology R&D cost ke^2 (see Fig. 1b), as in the case of BYD and Huawei; (3) Cooperation Model C is the cooperation mode in which, in addition to the sales channels of the new energy vehicle manufacturer, the technology supplier also provides sales channels and related technical support for the new energy vehicle manufacturer, so as to maximize the profit of the entire dual sales-channel supply chain system. Finally, the technology supplier shares the profits obtained by the new energy vehicle manufacturer according to the proportion ϕ agreed in the contract (see Fig. 1c), as in the case of Seres and Huawei. Based on this, this paper explores the selection of the optimal cooperation

mode between the new energy vehicle manufacturer and the technology supplier and analyzes the impact of different cooperation modes on the decision-making and profits of both parties.

For different cooperation modes g ($g \in \{A, B, C\}$), the game relationship between the technology suppliers and the new energy vehicle manufacturers is also different. Under cooperation Models A and B, the technology supplier and the new energy vehicle manufacturer aim to maximize their respective profits by making optimal pricing and technology R&D investment decisions. Based on Wang *et al.* [23-24], the Stackelberg game between the technology supplier and the new energy vehicle manufacturer is considered: (1) the technology supplier first determines its technology R&D investment level e and wholesale price w; (2) the new energy vehicle manufacturer then determines its retail price p according to the optimal decisions of the technology supplier. In cooperative Model C, according to the research of Jian *et al.* [25], Liu *et al.* [26], Haiju *et al.* [27], the decision-making of the technology supplier and the new energy vehicle manufacturer is considered as follows: (1) to determine the optimal technology R&D investment level e and retail price p with the goal of maximizing the profit of the entire dual sales-channel supply chain; (2) technology supplier and the new energy vehicle manufacturer will discuss their respective profit sharing ratio ϕ value according to their respective contributions, $\phi \in (0,1)$. The relevant parameter settings in this paper are shown in Table 1.

Table 1 Symbols and definitions

Symbols	Meaning
a	Market demand scale
b	Consumer sensitivity to retail prices
λ	The sensitivity coefficient of consumers to the level of investment in technology R&D
c	Unit production cost
e	Level of investment in technology R&D
w_g	Wholesale price in mode g, $g \in A, B, C\}$
p_g	The retail price assigned to the product developed by the technology supplier in mode g (referred to as retail price p_g)
π_T	Total profit of the technology supplier and the new energy vehicle manufacturer
$*$	Optimal value
c_1	Unit sales cost of new energy vehicle manufacturer
c_2	Technology supplier unit cost of sales
k	Technology R&D cost factor
β	Technology supplier's share of technology development costs
α	The market share of new energy vehicle manufacturer
ϕ	The proportion of the new energy vehicle manufacturer's profit allocated to the technology supplier (referred to as the revenue-sharing coefficient)
π_M^g	Profit for technology supplier
π_R^g	The profits of new energy vehicle manufacturer
$\{A, B, C\}$	{Modes A, B, C}

3. Model construction

This section mainly analyzes the optimal decision-making of the technology supplier and the new energy vehicle manufacturer under three cooperation modes. First, it analyzes the cooperation mode of Model A, and then examines the optimal decision-making of the technology supplier and the new energy vehicle manufacturer under the cooperation mode of Model B. Finally, it analyzes the optimal decisions of each party under Model C.

3.1 Model A

In Model A, the technology supplier independently conducts technology R&D and bears all R&D costs $k(e^A)^2$, while the new energy vehicle manufacturer purchases the technology supplier's products at the unit wholesale price w^A. For example, Huawei provides Chang'an with autonomous driving solutions and in-vehicle communication technology support. At this stage, a Stackelberg game is conducted between the new energy vehicle manufacturer and the technology supplier. Specifically, the technology supplier first determines the wholesale price w^A and the technology R&D investment level e^A, after which the new energy vehicle manufacturer determines the retail

price p^A. The optimization problems of the technology supplier and the new energy vehicle manufacturer can be expressed as follows:

$$\max_{w^A, e^A} \pi_M^A = w^A(a - bp^A + \lambda e^A) - k(e^A)^2 \tag{1}$$

$$\max_{p^A} \pi_R^A = (p^A - c - c_1 - w^A)(a - bp^A + \lambda e^A) \tag{2}$$

In Model A, Eqs. 1 and 2 are solved using backward induction to obtain the optimal decisions and profits of the technology supplier and the new energy vehicle manufacturer, as shown in Proposition 1.

Proposition 1. Under Model A, the optimal wholesale price, retail price, technology R&D investment level, the technology supplier profit, and the new energy vehicle manufacturer profit are, respectively,

$$\begin{cases} w^{A*} = \dfrac{4k[a - b(c + c_1)]}{8bk - \lambda^2} \\ p^{A*} = \dfrac{6ak + (c + c_1)(2bk - \lambda^2)}{8bk - \lambda^2} \\ e^{A*} = \dfrac{[a - b(c + c_1)]\lambda}{8bk - \lambda^2} \\ \pi_M^{A*} = \dfrac{[a - b(c + c_1)]^2 k}{8bk - \lambda^2} \\ \pi_R^{A*} = \dfrac{4bk^2[a - b(c + c_1)]^2}{(\lambda^2 - 8bk)^2} \end{cases} \tag{3}$$

3.2 Model B

When the new energy vehicle manufacturer chooses to engage in vertical technology R&D cooperation with the technology supplier, following Zhou *et al.* [7] and Zheng *et al.* [11-12], the new energy vehicle manufacturer shares a proportion $(1 - \beta)$ of the technology R&D cost $k(e^B)^2$ of the technology supplier, while the technology supplier bears the remaining cost $\beta k(e^B)^2$. In practice, BYD and Huawei have joined forces to promote technological R&D cooperation in intelligent driving technology innovation. The Stackelberg game is conducted between the new energy vehicle manufacturer and the technology supplier. Specifically, the technology supplier first determines the wholesale price w^B and investment level e^B of technology R&D, after which the new energy vehicle manufacturer determines the retail price p^B. The optimization problems of the technology supplier and the new energy vehicle manufacturer can be expressed as:

$$\max_{w^B, e^B} \pi_M^B = w^B(a - bp^B + \lambda e^B) - \beta k(e^B)^2 \tag{4}$$

$$\max_{p^B} \pi_R^B = (p^B - c - c_1 - w^B)(a - bp^B + \lambda e^B) - (1 - \beta)k(e^B)^2 \tag{5}$$

Similar to Model A, Eqs. 4 and 5 are solved using backward induction to obtain the optimal decisions and profits of the technology supplier and the new energy vehicle manufacturer, as shown in Proposition 2.

Proposition 2. Under Model B, the optimal wholesale price, retail price, investment level of technology R&D, profit of the technology supplier, and profit of the new energy vehicle manufacturer are, respectively,

$$\begin{cases} w^{B*} = \dfrac{4k\beta[a-b(c+c_1)]}{8bk\beta-\lambda^2} \\ p^{B*} = \dfrac{6ak\beta+(2bk\beta-\lambda^2)(c+c_1)}{8bk\beta-\lambda^2} \\ e^{B*} = \dfrac{[a-b(c+c_1)]\lambda}{8bk\beta-\lambda^2} \\ \pi_M^{B*} = \dfrac{[a-b(c+c_1)]^2k\beta}{8bk\beta-\lambda^2} \\ \pi_R^{B*} = \dfrac{[a-b(c+c_1)]^2[4bk\beta^2+(\beta-1)\lambda^2]k}{(\lambda^2-8bk\beta)^2} \end{cases} \tag{6}$$

3.3 Model C

Under Model C, following Jian *et al.* [25], Wang and Zhang [26], Haiju *et al.* [27], both the new energy vehicle manufacturer and the technology supplier aim to maximize the profit of the dual sales-channel supply chain system and achieve the goal by simultaneously determining the optimal technology R&D investment level e^{C*} and retail price p^{C*}. Under this model, the technology supplier alone bears the cost of technology R&D ke_3^2 and provides sales channels to help the new energy vehicle manufacturer sell products, thus forming a dual sales-channel supply chain system. Finally, the technology supplier and the new energy vehicle manufacturer share the benefits, that is, the new energy vehicle manufacturer transfers a fraction ϕ of its profit to the technology supplier, as observed in the Huawei–Seres partnership. Accordingly, the optimization problem of the dual sales-channel supply chain can be expressed as:

$$\max_{e^C,p^C} \pi_T^C = (p_3-c-c_1)\alpha(a-bp_3+\lambda e_3) + (p_3-c-c_2)(1-\alpha)(a-bp_3+\lambda e_3) - ke_3^2 \tag{7}$$

The profits of the technology supplier π_M^C and the new energy vehicle manufacturer π_R^C are, respectively,

$$\pi_M^C = (p_3-c-c_2)(1-\alpha)(a-bp_3+\lambda e_3) - ke_3^2 + \phi\alpha(p_3-c-c_1)(a-bp_3+\lambda e_3) \tag{8}$$

$$\pi_R^C = (1-\phi)\alpha(p_3-c-c_1)(a-bp_3+\lambda e_3) \tag{9}$$

In the above equation, α is the market share occupied by the new energy vehicle manufacturer, and $(1-\alpha)$ is the market share occupied by the technology supplier. Similar assumptions have also been made in studies by Rong *et al.* [28], Qin *et al.* [29]. Under Model C, the optimal technical input level, retail price, and participant profit of the dual sales-channel supply chain system can be obtained, as shown in Proposition 3.

Proposition 3. Under Model C, the optimal wholesale price, retail price, investment level of technology R&D, the profit of the technology supplier, and the profit of the new energy vehicle manufacturer are, respectively,

$$\begin{cases} p^{C*} = \dfrac{2ak+(c+c_2+c_1\alpha-c_2\alpha)(2bk-\lambda^2)}{4bk-\lambda^2} \\ e^{C*} = \dfrac{[a-b(c+c_2+c_1\alpha-c_2\alpha)]\lambda}{4bk-\lambda^2} \\ \pi_M^{C*} = \dfrac{k[a-b(c+c_2+c_1\alpha-c_2\alpha)]Z_1}{(\lambda^2-4bk)^2} \\ \pi_R^{C*} = \dfrac{2bk\alpha(\phi-1)[a-b(c+c_2+c_1\alpha-c_2\alpha)]Z_2}{(\lambda^2-4bk)^2} \end{cases} \tag{10}$$

$$Z_1 = \begin{pmatrix} -a\lambda^2+4abk\big(1+\alpha(-1+\phi)\big)+b\lambda^2\begin{pmatrix} c+c_2(\alpha-1)[2\alpha(\phi-1)-1] \\ +c_1\alpha[2\phi-1-2\alpha(\phi-1)] \end{pmatrix} \\ -4b^2k\big(c+c_2(\alpha-1)(\alpha(\phi-1)-1)+c\alpha(\phi-1)+c_1\alpha(\alpha+2\phi-\alpha\phi-1)\big) \end{pmatrix}$$

$$Z_2 = 2bk[c-c_1(\alpha-2)-2ak+c_2(\alpha-1)] + (c_1-c_2)(\alpha-1)\lambda^2$$

3.4 Model comparison

Propositions 4 and 5 can be obtained by comparing optimal pricing, the technology R&D investment level, and profits under Models A, B, and C.

Proposition 4. Comparison of the optimal pricing and R&D investment level under the three cooperation models: (1) $w^{B*} > w^{A*}$; (2) $p^{B*} > p^{A*} > p^{C*}$; (3) when $0 < \beta < \beta_1$, then $e^{B*} > e^{C*} > e^{A*}$; when $\beta > \beta_1$, then $e^{C*} > e^{B*} > e^{A*}$, $\beta_1 = \frac{4ak-4bk(c+c_1)+(c_1-c_2)(1-\alpha)\lambda^2}{8k[a-b(c+c_2+c_1\alpha-c_2\alpha)]}$.

Proposition 1 shows that: (1) the wholesale price under Model B is always greater than that under Model A. This is mainly because the cooperative R&D sales-channel mode stimulates the technology supplier's R&D investment level and increases its technology input cost, which in turn leads to an increase in the wholesale price of the product; (2) the retail price of products under models B and C is the highest and lowest, respectively. This is because Model C aims to maximize the profit and market share of the entire dual sales-channel supply chain, and the retail price under Model C's centralized decision will be the lowest. The retail price of Model B is greater than that of Model A because of the size of the wholesale price; (3) the level of technology R&D investment under Model A is always the lowest among the three models. When the technology R&D investment cost-sharing ratio $(1-\beta)$ between the new energy vehicle manufacturer and the technology supplier is small $(0 < \beta < \beta_1)$, the technology supplier's technology R&D investment level in Model B will be greater than that in Model C; otherwise, the technology supplier's technology R&D investment level in Model C will be greater than that in Model B. Without loss of generality, the parameters are set as follows: as $a = 100$, $c = 2$, $c_1 = 1$, $c_2 = 0.8$, $b = 2$,$\lambda = 1$, $k = 0.5$, $\alpha = 0.4$. Meanwhile, $\beta > 0.125$ can be obtained because the Cosset matrix in Section 3.2 needs to satisfy $8bk\beta - \lambda^2 > 0$ (see Figs. 2-4).

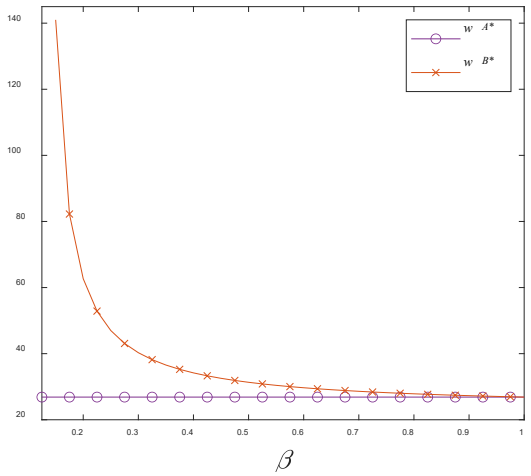


Fig. 2 The effect of β on wholesale prices

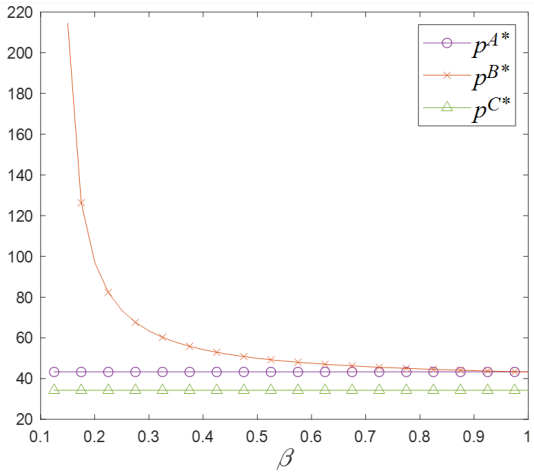


Fig. 3 The effect of β on retail prices

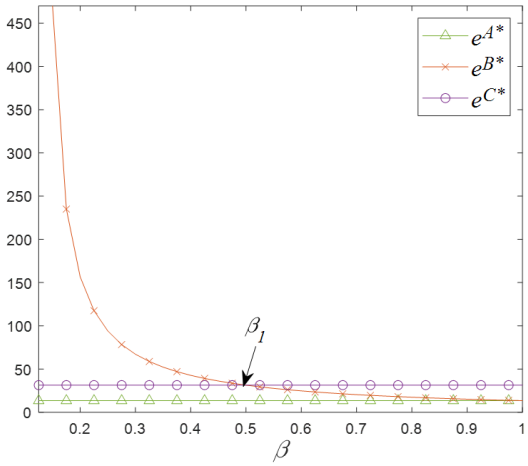


Fig. 4 The influence of β on the level of investment in technology R&D

Proposition 5. Compare the optimal profit of the technology supplier and the new energy vehicle manufacturer under the three cooperation models: (1) when $\phi < \phi_1$, then $\pi_M^{B*} > \pi_M^{C*} > \pi_M^{A*}$; when $\phi > \phi_1$, then $\pi_M^{C*} > \pi_M^{B*} > \pi_M^{A*}$; (2) when $\phi < \phi_2$ and $\beta < \beta_2$, then $\pi_R^{C*} > \pi_R^{A*} > \pi_R^{B*}$; when $\phi < \phi_2$ and $\beta > \beta_2$, then $\pi_R^{C*} > \pi_R^{B*} > \pi_R^{A*}$; when $\phi_2 < \phi < \phi_3$ and $\beta < \beta_2$, then $\pi_R^{A*} > \pi_R^{C*} > \pi_R^{B*}$; when $\phi > \phi_3$ and $\beta < \beta_2$, then $\pi_R^{A*} > \pi_R^{B*} > \pi_R^{C*}$; when $\phi > \phi_3$ and $\beta > \beta_2$, then $\pi_R^{B*} > \pi_R^{A*} > \pi_R^{C*}$. Where $\beta_2 = \frac{64b^2k^2 - 12bk\lambda^2 + \lambda^4}{64b^2k^2 - 4bk\lambda^2}$,

$$\phi_2 = 1 + \frac{k\{2[a - b(c + c_1)]^2(\lambda^2 - 4bk)^2\}}{\begin{pmatrix}\alpha[a - b(c + c_2 + c_1\alpha - c_2\alpha)](\lambda^2 - 8bk)^2 \\ \{2bk[c - c_1(\alpha - 2) + c_2(\alpha - 1)] - 2ak + \lambda^2(c_2 - c_1)(1 - \alpha)\}\end{pmatrix}}$$

$$\phi_3 = 1 + \frac{[a - b(c + c_1)]^2(\lambda^2 - 4bk)^2[4bk\beta^2 + \lambda^2(\beta - 1)]}{\begin{pmatrix}2b\alpha[a - b(c + c_2 + c_1\alpha - c_2\alpha)](\lambda^2 - 8bk\beta)^2 \\ \{2bk[c - c_1(\alpha - 2) + 2c(\alpha - 1)] - 2ak + (c_1 - c_2)(\alpha - 1)\lambda^2\}\end{pmatrix}}$$

$$\phi_1 = \frac{\begin{pmatrix}a^2(16b^2k^2(1 - 2\alpha)\beta + 4bk(\alpha - 1)\lambda^2 - (\beta - 1)\lambda^4) + 2ab(16b^2k^2 \\ \big(c_1 + 2c_2(\alpha - 1) + c(2\alpha - 1)\big)\beta - 4bk(\alpha - 1)(c + c_2 - 2c_1(1 + \alpha)\beta \\ +2c_2(1 + \alpha)\beta)\lambda^2 + \big(c_2(\alpha^2 - 1) + c(\beta - 1) + c_1(\beta - \alpha^2)\big)\lambda^4) \\ -b^2(16b^2k^2(2c\big(c_1 + 2c_2(\alpha - 1)\big) + 4c_1c_2(\alpha - 1)\alpha^2 - 2{c_2}^2(\alpha - 1)^2(1 + \alpha) \\ +c^2(2\alpha - 1) + {c_1}^2(1 + 2\alpha^2 - 2\alpha^3))\beta - 4bk(\alpha - 1)(c^2 + 2c_1c_2\alpha^2(1 + 4\beta) \\ -{c_2}^2(\alpha - 1)(1 + \alpha + 2\beta + 4\alpha\beta) - {c_1}^2(\alpha^2 + 2\beta + 2\alpha\beta + 4\alpha^2\beta) + 2c \\ (c_2 - 2c_1(1 + \alpha)\beta + 2c_2(1 + \alpha)\beta))\lambda^2 + (4c_1c_2(\alpha - 1)\alpha^2 - {c_2}^2(\alpha - 1)^2 \\ (1 + 2\alpha) + c^2(\beta - 1) + {c_1}^2(\alpha^2 - 2\alpha^3 + \beta) + 2c\big(c_2(\alpha^2 - 1) + c_1(\beta - \alpha^2)\big))\lambda^4)\end{pmatrix}}{\begin{pmatrix}2b\alpha[b(c + c_2 + c_1\alpha - c_2\alpha) - a](8bk\beta - \lambda^2) \\ \{2ak - 2bk[c - c_1(\alpha - 2) + c_2(\alpha - 1)] - (c_1 - c_2)(\alpha - 1)\lambda^2\}\end{pmatrix}}$$

Proposition 5 mainly analyzes the impact of three cooperation modes on the profits of the technology supplier and the new energy vehicle manufacturer. Specifically, (1) the profit of the technology supplier in Model A is always the smallest, while the profit size between models B and C depends on the revenue sharing ratio ϕ: if the profit proportion of the technology supplier in the new energy vehicle manufacturer is smaller ($\phi < \phi_1$), that in Model B is larger; conversely, the technology supplier will gain more profits in Model C. (2) The profit of the new energy vehicle manufacturer under the three cooperation models depends on the proportional coefficient of revenue sharing and the proportional coefficient of sharing of technology R&D costs of the new energy vehicle manufacturer to technology supplier: (a) when the technology supplier shares a small proportion of the profit of the new energy vehicle manufacturer under Model C ($\phi < \phi_2$) and the proportion of the technology R&D costs ($1 - \beta$) shared by the new energy vehicle manufacturer under B is large ($\beta < \beta_2$), the profit of the new energy vehicle manufacturer under Model C is the largest, followed by models A and B. This is because under the centralized decision-making of Model C, the new energy vehicle manufacturer has a relatively large profit share, while in Model B, it bears a higher proportion of technology R&D costs, which reduces overall profitability. (b) When the technology supplier shares a smaller proportion of the profits of the new energy vehicle manufacturer under Model C ($\phi < \phi_2$) and the proportion of technology R&D costs ($1 - \beta$) shared by the new energy vehicle manufacturer under Model B ($\beta > \beta_2$), the new energy vehicle manufacturer obtains the largest profits under Model C, followed by models B and A. (c) When the technology supplier shares a larger proportion of the profits of the new energy vehicle manufacturer under Model C ($\phi_2 < \phi < \phi_3$) and the new energy vehicle manufacturer shares a higher proportion of technology R&D costs ($1 - \beta$) under Model B ($\beta < \beta_2$), the new energy vehicle manufacturer obtains the largest profits under Model A, followed by models C and B; (d) When technology supplier shares a large proportion of the profit of the new energy vehicle manufacturer under Model C ($\phi > \phi_3$) and the new energy vehicle manufacturer shares a high proportion of

technology R&D costs $(1-\beta)$ under Model B ($\beta < \beta_2$), the new energy vehicle manufacturer obtains the largest profits under Model A, followed by models B and C. (e) When technology supplier shares a large proportion of the profit of the new energy vehicle manufacturer under Model C ($\phi > \phi_3$) and the proportion of technology R&D costs $(1-\beta)$ shared by the new energy vehicle manufacturer under Model B is small ($\beta > \beta_2$), the new energy vehicle manufacturer obtains the largest profit under Model B, followed by models A and C in order.

Fig. 5 illustrates the impact of the technology supplier's revenue-sharing coefficient ϕ and the new energy vehicle manufacturer's technology R&D cost-sharing coefficient β on the profits of both parties. Specifically: (1) in the shaded areas of *F* and *H* ($\phi_1 < \phi < \phi_2$), both the technology supplier and the new energy vehicle manufacturer achieve a Pareto improvement in profits under Model C and a *win-win outcome*. This provides a theoretical explanation for why Huawei and Seres adopt Model C in practice. (2) The profit ranking of the technology supplier satisfies $\pi_M^{B*} > \pi_M^{C*} > \pi_M^{A*}$ in regions *A* and *E*, while $\pi_M^{C*} > \pi_M^{B*} > \pi_M^{A*}$ holds in regions *B*, *C*, *D*, *F*, and *H*. (3) From the perspective of the new energy vehicle manufacturer, $\pi_R^{C*} > \pi_R^{A*} > \pi_R^{B*}$ can be seen in regions *E* and *F*; $\pi_R^{C*} > \pi_R^{B*} > \pi_R^{A*}$ can be seen in region *H*, and $\pi_R^{A*} > \pi_R^{C*} > \pi_R^{B*}$ can be seen in regions *A* and *B*; $\pi_R^{A*} > \pi_R^{B*} > \pi_R^{C*}$ can be seen in region C; $\pi_R^{B*} > \pi_R^{A*} > \pi_R^{C*}$ can be seen in region *D*.

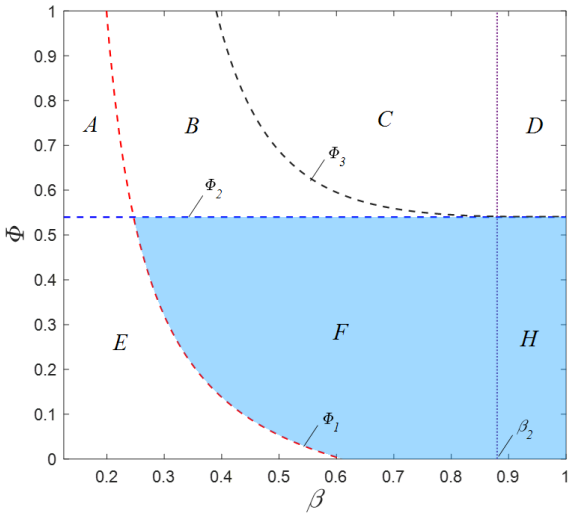


Fig. 5 The impact of ϕ and β on the profits of the technology supplier and the new energy vehicle manufacturer

4. Conclusion

New energy vehicle manufacturers frequently collaborate with specialized technology suppliers to address their technical shortcomings, thereby enhancing their market share. However, a diverse array of cooperation models exists in practice. This paper aims to explore which cooperation mode is most advantageous for new energy vehicle manufacturers and technology suppliers under varying circumstances, as well as the conditions both parties agree upon when selecting the same cooperation model. A review of the existing literature revealed a scarcity of studies focusing on the choice of cooperation models between new energy vehicle manufacturers and technology suppliers. To address this gap, this study examined three specific partnerships: Chang'an and Huawei, BYD and Huawei, and Seres and Huawei. It considered three cooperation models: A, B, and C. These models form single- and dual-channel supply chain systems comprising a technology supplier and a new energy vehicle manufacturer. Through comprehensive analysis, the following conclusions are drawn.

- Under Model B, the retail price of the products developed by the technology supplier is the highest, followed by models A and C. The wholesale price of the products developed by the technology supplier under Model B will also be higher than that of products developed under Model A.

- In Model A, the level of R&D investment by the technology supplier is always minimal. The level of technology R&D investment between models B and C depends on the technology R&D cost-sharing coefficient β: if β is small, the technology supplier has the highest level of technology R&D investment in Model B; otherwise, the highest level of R&D investment occurs in Model C.
- From the perspective of the technology supplier, a threshold ϕ_1 applies. If the revenue-sharing coefficient ϕ of the technology supplier is less than this threshold, the optimal strategy is to choose Model B; otherwise, Model C should be chosen.
- From the perspective of the new energy vehicle manufacturer, thresholds ϕ_2, ϕ_3, and β_2 apply, with $\phi_2 < \phi_3$. (a) If the revenue-sharing coefficient ϕ is less than ϕ_2, then the optimal strategy is to choose Model C. (b) If the revenue-sharing coefficient ϕ satisfies $\phi > \phi_2$ and the technology supplier's R&D cost-sharing coefficient satisfies $\beta < \beta_2$, the optimal strategy for the new energy vehicle manufacturer is to choose Model A. (c) If the revenue-sharing coefficient ϕ satisfies $\phi > \phi_3$ and the technology supplier's R&D cost-sharing coefficient satisfies $\beta > \beta_2$, the optimal strategy for the new energy vehicle manufacturer is to choose Model B. (d) If the revenue-sharing coefficient ϕ satisfies $\phi_1 < \phi < \phi_2$, the optimal strategy of the technology supplier and the new energy vehicle manufacturer is Model C, which explains why Seres and Huawei have chosen Model C in practice.
- The conclusions provide theoretical guidance for new energy vehicle manufacturers and technology suppliers in selecting cooperation modes. In future research, scenarios involving information asymmetry and competition may be further considered.

Acknowledgement

This work was supported by the Social Science Planning Project of Chongqing (2024NDYB075), the National Natural Science Foundation of China (72471040), and the Humanities and Social Sciences Research Project of Chongqing Municipal Education Commission (25SKGH135).

References

[1] Veugelers, R., Cassiman, B. (1999). Make and buy in innovation strategies: Evidence from Belgian manufacturing firms, *Research Policy*, Vol. 28, No. 1, 63-80, doi: 10.1016/S0048-7333(98)00106-1.
[2] Cassiman, B., Veugelers, R. (2006). In search of complementarity in innovation strategy: Internal R&D and external knowledge acquisition, *Management Science*, Vol. 52, No. 1, 68-82, doi: 10.1287/mnsc.1050.0470.
[3] Guo, B., Pérez-Castrillo, D., Toldrà-Simats, A. (2019). Firms' innovation strategy under the shadow of analyst coverage, *Journal of Financial Economics*, Vol. 131, No. 2, 456-483, doi: 10.1016/j.jfineco.2018.08.005.
[4] Zhang, Y., Wang, C.Y., Cai, L.X. (2023). Innovation strategy selection of high energy-consuming enterprises under the "Dual Control" policy: Independent R&D or technology introduction, *Research on Economics and Management*, Vol. 44, No. 8, 89-108, doi: 10.13502/j.cnki.issn1000-7636.2023.08.006.
[5] Wang, S.S., Cai, L.H., Li, Y. (2023). Economic sanctions, technology introduction and independent innovation, *Nankai Economic Studies*, No. 6, 21-39, doi: 10.14116/j.nkes.2023.06.002.
[6] Xie, W.-M., Zhang, H.-X. (2023). Make or buy: Performance-vested equity incentives and firm innovation strategy, *Systems Engineering-Theory & Practice*, Vol. 43, No. 1, 36-57, doi: 10.12011/SETP2022-0450.
[7] Zhou, X.-H., Zhang, J.-H., Xu, J. (2021). Research on the motivation for R&D cooperation between firms based on sequential game, *Journal of Management Sciences in China*, Vol. 24, No. 2, 111-126, doi: 10.19920/j.cnki.jmsc.2021.02.007.
[8] Cellini, R., Lambertini, L. (2009). Dynamic R&D with spillovers: Competition vs cooperation, *Journal of Economic Dynamics and Control,* Vol. 33, No. 3, 568-582, doi: 10.1016/j.jedc.2008.08.006.
[9] Gupta, S. (2008). Research note – Channel structure with knowledge spillovers, *Marketing Science*, Vol. 27, No. 2, 247-261, doi: 10.1287/mksc.1070.0285.
[10] Petrakis, E., Tsakas, N. (2018). The effect of entry on R&D networks, *The RAND Journal of Economics*, Vol. 49, No. 3, 706-750, doi: 10.1111/1756-2171.12250.
[11] Zheng, Y., Zhou, B., Hao, C., Gao, R., Li, M. (2024). Evolutionary game analysis on the cross-organizational cooperative R&D strategy of general purpose technologies under two-way collaboration, *Technology in Society*, Vol. 76, Article No. 102473, doi: 10.1016/j.techsoc.2024.102473.
[12] Zheng, Y., Liu, S., Zhao, Y., Han, C., Zhou, Q., Wang, L., Colace, F., Alhalabi, W., Alsharif, H. (2023). Game analysis on general purpose technology cooperative R&D with fairness concern from the technology chain perspective, *Journal of Innovation & Knowledge*, Vol. 8, No. 1, Article No. 100312, doi: 10.1016/j.jik.2023.100312.

[13] Yan, W.J., Wang, X.X., Li, D.H., Guo, J. (2023). Research on evolutionary stable strategy for cooperative lnnovation of key part of core technology under the monopoly condition, *Chinese Journal of Management Science*, Vol. 31, No. 7, 227-236, doi: 10.16381/j.cnki.issn1003-207x.2022.0474.
[14] De Giovanni, P. (2014). Environmental collaboration in a closed-loop supply chain with a reverse revenue sharing contract, *Annals of Operations Research*, Vol. 220, 135-157, doi: 10.1007/s10479-011-0912-5.
[15] Chen, J., Hu, Q., Song, J.-S. (2017). Supply chain models with mutual commitments and implications for social responsibility, *Production and Operations Management*, Vol. 26, No. 7, 1268-1283, doi: 10.1111/poms.12674.
[16] Jalili, M., Pangburn, M.S. (2020). Pricing joint sales and rentals: When are purchase conversion discounts optimal?, *Production and Operations Management,* Vol. 29, No. 12, 2679-2695, doi: 10.1111/poms.13243.
[17] Hoogland, K., Chakraborty, D., Hardman, S. (2022). To purchase or lease: Investigating the finance decision of plug-in electric vehicle owners in California, *Environmental Research Communications,* Vol. 4, No. 9, Article No. 095005, doi: 10.1088/2515-7620/ac8397.
[18] Zhou, X.J., Li, J.-Z., Li, F., Deng, X.-D. (2023). Recycling supply chain mode of new energy vehicle power battery based on Blockchain technology, *Computer Integrated Manufacturing System*, Vol. 29, No. 4, 1386-1398, doi: 10.13196/j.cims.2023.04.029.
[19] Huang, Y., Qian, L., Soopramanien, D., Tyfield, D. (2021). Buy, lease, or share? Consumer preferences for innovative business models in the market for electric vehicles, *Technological Forecasting and Social Change*, Vol. 166, Article No. 120639, doi: 10.1016/j.techfore.2021.120639.
[20] Wang, Y.-L., Yin, X.-M., Zheng, X.-Y., Chen, W. (2023). Supply chain decision-making considering green technology effort: Effect on random output and retail price with fairness concerns, *Economic Computation & Economic Cybernetics Studies and Research*, Vol. 57, No. 1, 103-120, doi: 10.24818/18423264/57.1.23.07.
[21] Wang, Y.L., Yang, L., Chen, J.H., Li, P. (2023). Supply chain game analysis based on mean-variance and price risk aversion under different power structures, *Advances in Production Engineering & Management*, Vol. 18, No. 1, 104-115, doi: 10.14743/apem2023.1.460.
[22] Duan, H.W., Wang, M.T., Ye, Y.S. (2022). Financing and information sharing in capital-constrained supply chain, *Advances in Production Engineering & Management*, Vol. 17, No. 3, 263-278, doi: 10.14743/apem2022.3.435.
[23] Wang, Y.L., Chen, J.H., Yang, L., Fang, X., Cai, J.R. (2023). A simulation study on supply chain financing strategy of manufacturing firms, *International Journal of Simulation Modelling*, Vol. 22, No. 2, 327-337, doi: 10.2507/IJSIMM22-2-CO8.
[24] Wang, Y.L., Song, M.L., Yang, L., Chen, J.H., Li, P. (2024). Financing green innovation: A simulation approach for manufacturing enterprises, *International Journal of Simulation Modelling*, Vol. 23, No. 2, 347-358, doi: 10.2507/IJSIMM23-2-CO8.
[25] Jian, M., Liu, T., Hayrutdinov, S., Fu, H. (2022). Supply chain coordination based on the probability optimization of target profit, *Advances in Production Engineering & Management*, Vol. 17, No. 2, 169-182, doi: 10.14743/apem2022.2.428.
[26] Liu, P., Atifeh, M., Khorshidnia, M., Taheri, S.G. (2023). System dynamics: An approach to modeling supply chain performance measurement, *Technological and Economic Development of Economy*, Vol. 29, No. 4, 1291-1317, doi: 10.3846/tede.2023.19211.
[27] Hu, H., Li, Y., Li, M. (2024). Dual-channel supply chain coordination considering the consumer's perception of quality, *Economic Computation and Economic Cybernetics Studies and Research*, Vol. 58, No. 2, 198-216, doi: 10.24818/18423264/58.2.24.12.
[28] Rong, A., Akkerman, R., Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain, *International Journal of Production Economics*, Vol. 131, No. 1, 421-429, doi: 10.1016/j.ijpe.2009.11.026.
[29] Qin, Y., Wang, J., Wei, C. (2014). Joint pricing and inventory control for fresh produce and foods with quality and physical quantity deteriorating simultaneously, *International Journal of Production Economics*, Vol. 152, 42-48, doi: 10.1016/j.ijpe.2014.01.005.

APEM journal
Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 351–368
https://doi.org/10.14743/apem2025.3.545
ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Two-echelon drone–truck collaborative TSP-based routing for humanitarian logistics with time windows and stochastic demand

Xiao, N.[a,*], **Lan, H.**[a]

[a]School of Economics and Management, Beijing Jiaotong University, Haidian District, Beijing, P.R. China

ABSTRACT

In humanitarian logistics emergency material transportation and distribution, trucks offer large load capacity and long driving range, whereas drone transportation is independent of ground road conditions but constrained by battery life and payload capacity. The coordination of the two can therefore provide complementary advantages. In this paper, the traveling salesman problem is formulated for a two-echelon emergency material distribution process, spanning transportation from the central warehouse to the distribution center and then to the demand points. In the first stage, transportation from the central warehouse to the distribution center is performed by trucks. In the second stage, trucks and drones collaboratively carry out material distribution from the distribution center to the demand points. Based on the above scenario, this paper aims to minimize the total cost of completing all distribution tasks. The model considers capacity constraints at distribution centers, time window constraints at demand points, and stochastic demand, and establishes a two-echelon traveling salesman problem for humanitarian logistics with truck–drone collaboration. Based on the particle swarm optimization (PSO) framework, a heuristic algorithm named PSO-VD is proposed, which transforms the discrete traveling salesman problem into a continuous encoding and integrates drone routes into truck routes using the 2-opt method. In small-scale instances, the solutions obtained by PSO-VD are compared with those of commercial solvers, demonstrating that the proposed algorithm achieves high accuracy with low computational time. For instances with up to 12 demand points, the algorithm obtains solutions within 150 seconds, with an accuracy deviation of less than 10 % compared to exact solution methods. The applicability of the algorithm proposed in this paper has been demonstrated through large-scale numerical examples. Sensitivity analyses are conducted on key parameters, including the time window penalty coefficient, drone speed, and drone battery capacity, yielding practical managerial insights.

ARTICLE INFO

Keywords:
Humanitarian logistics;
Two-echelon routing;
Drone–vehicle collaboration;
Stochastic demand;
Time windows;
Capacity constraint;
Vehicle routing problem (VRP);
Travelling salesman problem (TSP);
Heuristic algorithm;
Particle Swarm Optimization (PSO)

**Corresponding author:*
20113045@bjtu.edu.cn
(Xiao, N.)

Article history:
Received 1 September 2025
Revised 6 October 2025
Accepted 9 October 2025

1. Introduction

Drones are playing a crucial role in humanitarian logistics by enabling faster, more efficient aid delivery to remote and disaster-stricken areas. In Africa, Zipline operates autonomous drones to deliver blood, vaccines, and medical supplies to clinics in Rwanda, Ghana, Nigeria, and beyond, cutting delivery times from hours to minutes. Following Hurricane Maria in Puerto Rico, drones were deployed to assess damage and deliver aid to isolated communities. Similarly, during the COVID-19 pandemic, Matternet's drones transported test kits, PPE, and vaccines between hospi-

tals in the U.S. and Switzerland, reducing exposure risks. The World Food Programme tested drone deliveries for food aid in famine-affected Madagascar, while search and rescue teams used drones after the 2015 Nepal earthquake to locate survivors and assess damage. The UNHCR leveraged drone technology for refugee camp planning and monitoring in Greece and Bangladesh, improving logistical coordination. Additionally, drones have been used in Australia and California for wildfire monitoring and evacuation planning. These innovations are transforming humanitarian logistics by reducing response times, lowering costs, and improving access to life-saving aid. With the rapid development of Industry 4.0 and artificial intelligence, automation technologies have increasingly penetrated manufacturing and service industries, leading to the widespread adoption of drones [1-2]. It has a wide range of applications, including but not limited to industrial manufacturing, logistics and distribution, healthcare, rescue exploration, etc. [3].

The use of drones in humanitarian logistics presents significant advantages, including rapid delivery of life-saving supplies by bypassing infrastructure challenges—as demonstrated by Zipline's medical drone operations in remote Africa, which reduced delivery times from hours to minutes. Drones excel in accessing hard-to-reach areas affected by disasters, conflict, or geography, while also minimizing risks to human responders in hazardous environments. Their cost efficiency, scalability, and real-time data capabilities (evident in post-2015 Nepal earthquake assessments and wildfire monitoring in California and Australia) enhance disaster response and planning. Additionally, eco-friendly operations and improved security monitoring further position drones as transformative tools. However, barriers such as restrictive aviation regulations, limited payload capacities, short battery life, high upfront costs, and technical challenges like weather vulnerability and operator expertise hinder widespread adoption. Privacy concerns, public skepticism, and infrastructure gaps in remote areas further complicate deployment. Addressing these limitations demands collaborative efforts to technology [4], streamline regulations, increase investment, and integrate drones with traditional logistics strategies, ensuring their full potential is realized in delivering timely, efficient, and sustainable humanitarian aid.

A hybrid truck-drone distribution model combines the efficiency of trucks for bulk transport with the agility of drones for last-mile delivery, offering a transformative solution for humanitarian aid, emergency response, and e-commerce logistics. In this system, trucks act as mobile hubs, carrying large quantities of supplies such as food, medicine, or emergency kits while serving as launchpads and recharging stations for drones. Once trucks reach centralized locations—like disaster relief camps or regional hubs—drones take over to deliver smaller, critical items such as vaccines, medical kits, or communication devices to remote, inaccessible, or disaster-stricken areas blocked by rough terrain or infrastructure damage. Operational strategies like parallel delivery (simultaneous truck and drone operations), relay systems (drones picking up payloads from stationary trucks), and on-demand deployment (dynamic launches based on real-time needs) optimize speed and coverage. The model enhances efficiency by allowing trucks to handle bulk logistics while drones bypass traffic and geographical barriers, drastically reducing delivery times and costs. It also improves accessibility in isolated regions and scales rapidly, with multiple drones deployed from a single truck to cover vast areas. However, challenges such as limited drone battery life require frequent recharging at truck hubs, regulatory restrictions on airspace use, and payload capacity constraints compared to trucks must be addressed. By integrating AI-driven route optimization, real-time tracking, and automated scheduling, this hybrid approach promises to revolutionize supply chains, making them faster, more resilient, and adaptable to urgent demands in crisis scenarios or everyday logistics.

This study addresses the emergency material distribution vehicle routing problem from central warehouse to distribution centers and subsequently to demand points. Given that the transportation duration from central warehouse to distribution centers influences the scheduling of last-mile deliveries to demand points, a two-echelon vehicle routing approach is proposed. The first stage, termed trunk transportation, employs large-capacity trucks to transfer materials from central warehouse to distribution centers. The second stage, referred to as last-mile delivery, utilizes coordinated small-scale drones and trucks from distribution centers to demand nodes. Each distribution center stores truck and drone, with each truck carrying one drone to service multiple emergency demand points before returning to its origin. Notably, the trunk

transportation stage permits direct delivery by large drones to either distribution centers or demand points without truck-drone coordination.

The research establishes a truck-drone collaborative two-echelon travelling salesman problem model for emergency logistics, prioritizing total mission completion time minimization. The model incorporates operational constraints including payload capacities, mileage limitations, and road accessibility restrictions. A heuristic algorithm is developed to solve this NP-hard problem. The paper is structured as follows: Section 2 presents the literature review, Section 3 details the mathematical formulation, Section 4 describes the heuristic algorithm, Section 5 provides empirical validation, and Section 6 concludes with findings and implications.

2. Literature review

2.1 Vehicle routing problem with drone

Murray and Chu (2015) pioneered a truck-drone collaborative delivery system, framing the problem as the Flying Sidekick Traveling Salesman Problem (FSTSP), where each customer must be served either by a driver-operated truck or a drone coordinated with the truck [5]. Carlsson and Song (2018) demonstrated the efficiency of this system [6], while Agatz *et al.* (2018) developed an exact algorithm termed TSP-D [7]. Yurek and Ozmutlu (2018) proposed an iterative decomposition-based algorithm to minimize the completion time of TSP-D deliveries [8]. Ha *et al.* (2018) introduced a TSP-D variant aimed at minimizing operational costs, including transportation expenses and idle time penalties incurred by interdependencies between trucks and drones [9]. Kim *et al.* (2018) further advanced this with a robust optimization method to address battery duration uncertainties [10]. Roberti and Ruthmair (2020) employed dynamic programming for TSP-D solutions [11], and Tamke and Buscher (2021) incorporated waiting time into the objective function, proposing a novel mixed-integer linear programming (MILP) model enhanced with valid inequalities [12]. Vásquez *et al.* (2021) decomposed the problem into two stages: selecting and sequencing truck-served customers, then assigning remaining nodes to drones [13]. Kang and Lee (2021) devised an exact algorithm based on Benders Decomposition for logistical optimization [14].

In heterogeneous truck-drone routing research, Ham (2018) extended the problem by integrating delivery and pickup tasks with multiple vehicles and depots [15]. Kitjacharoenchai *et al.* (2019) advanced multi-vehicle drone-truck coordination, framing it as a multi-traveling salesman problem with drones (MTSP-D) and developing insertion-based heuristics for large-scale instances (up to 100 nodes) [16]. Murray and Raj (2020) proposed a heuristic decomposition approach for a single truck and multiple drones, breaking the problem into three subproblems [17]. Roberti and Ruthmair (2021) introduced a branch-and-cut algorithm [18], while Mohamed *et al.* (2022) formulated a MILP model for coordinated last-mile delivery using a truck and heterogeneous drones, minimizing completion time via a hybrid simulated annealing and variable neighborhood search algorithm [19]. Dynamic decision-making in operations was explored by Chen *et al.* (2021), who developed a Markov Decision Process (MDP) model for e-commerce logistics, dynamically adjusting delivery rates and pricing strategies under elastic demand [20]. Recent advancements in routing optimization include Lichau *et al.* (2025), who proposed an exact algorithm combining dynamic programming and branch-and-cut pricing to enhance two-echelon truck-drone (2E-VRP-D) efficiency [21]. Mbiadou Saleu *et al.* (2022) extended the parallel drone scheduling TSP (PDSTSP) with a MILP model for multi-vehicle coordination, though computational complexity limited scalability. In intelligent algorithms for complex scenarios [22], Ermağan *et al.* (2022) integrated machine learning with column generation for drone charging station routing, dynamically inserting charging nodes to address range constraints [23]. Stodola *et al.* (2024) designed an adaptive ant colony optimization algorithm (AACO-NC-D) for multi-depot heterogeneous fleets (MDVRP-D), achieving 15-20 % time savings via node clustering and pheromone evaporation [24]. Notably, humanitarian logistics research by Ramadhan *et al.* (2025) introduced a truck-drone-boat (TSP-DB) model for flood disasters, coordinating

drones (for dry/flooded zones) and inflatable boats (for flooded areas) to enhance rescue efficiency, though single-drone capacity per mission remains a limitation [25].

This evolving body of work underscores the growing sophistication of hybrid truck-drone systems, driven by algorithmic innovation, multi-objective optimization, and adaptability to diverse operational contexts.

2.2 Emergency logistics vehicle routing problem

Recent advancements in humanitarian logistics have focused on multimodal transportation planning and uncertainty management, with scholars delving into multi-commodity rebalancing, dynamic routing optimization, coordinated delivery, and intelligent algorithm development. In the realm of uncertainty modeling and equity optimization, Gao *et al.* (2021) proposed a bi-objective stochastic mixed-integer nonlinear programming (BOSMINP) model for large-scale disasters, balancing equity and efficiency by minimizing the weighted proportion of unmet demand while integrating priority weights into multi-commodity rebalancing for the first time [26]. Expanding on this, Gao *et al.* (2024) employed a Wasserstein distance-based distributionally robust optimization (DRO) framework to address pre-disaster facility location-inventory prepositioning and post-disaster resource allocation, enhancing robustness against road disruption risks [27]. Similarly, Ahmadi *et al.* (2015) utilized two-stage stochastic programming to manage post-earthquake network failures under the "golden 72-hour" constraint, validating the feasibility of dynamic multi-trip vehicle routing using GIS data [28]. Hou and Liu (2024) used the maximum regret value and Lyapunov central limit theorem to describe the uncertainty of demand and transportation time and established a robust optimization model for cold chain multimodal transportation routes considering mixed uncertainty of carbon emissions [29].

Multimodal coordination and time-sensitive optimization have emerged as key research themes. Yang *et al.* (2025) developed a two-stage stochastic model for truck-drone collaborative scheduling, enabling drones to reload at distribution centers to address single-node surge demand scenarios [30]. Lu *et al.* (2023) further incorporated time-varying weather impacts on drone safety and efficiency, designing a hybrid multi-objective evolutionary algorithm (HMOEA) that demonstrated weather-sensitive decision-making efficacy in the Henan floods case [31]. Alternatively, Yin *et al.* (2023) investigate a robust vehicle routing problem with trucks and drones under uncertain demands and travel times, highlighting optimization strategies for collaborative delivery under humanitarian logistics conditions [32].

Algorithmic innovation and dynamic response strategies have seen notable breakthroughs. Kyriakakis *et al.* (2022) proposed a hybrid tabu search-variable neighborhood descent algorithm (HTS-VND), achieving new optimal solutions for cumulative capacitated vehicle routing problems [33]. Van Steenbergen *et al.* (2023) pioneered the application of deep reinforcement learning (RL) to multi-vehicle routing under uncertain travel times, demonstrating that replacing 50 % of trucks with drones improved objective values by 11-56 % [34]. Amirsahami *et al.* (2025) integrated fuzzy programming with an improved NSGA-II algorithm (M-NSGA-II-AVNS), synchronizing decentralized facility expansion and drone routing in urban rescue chains, reducing costs by 75 % while shortening waiting times [35]. These studies collectively highlight the growing integration of advanced analytics, real-time adaptability, and multimodal synergies in addressing the complexities of humanitarian logistics.

3. Mathematical model

3.1 Problem description

This paper considers the two-stage TSP problem of humanitarian logistics in an emergency scenario. The first stage is from a central warehouse to multiple transit distribution centers, and the second stage is from the transit station to the demand point. At the same time, in the second stage, consider the use of drones and trucks for coordinated delivery, that is, each distribution center has one truck and one drone. At the beginning of the second stage, the truck sets off with cargos and drones to serve the demand point, and the truck is the take-off and landing platform

of the drone. When the truck serves the demand point, the drone takes off from the truck and serves other demand points. After the service is completed, when the truck serves another demand point, the drone lands back to the truck. For the humanitarian logistics scenario, we consider the uncertainty of demand, the capacity limit of distribution center and the time window of demand point. Therefore, when the capacity limit of the distribution center is exceeded, or the materials cannot be delivered to the demand point, a penalty coefficient should be given.As shown in figure 1, in stage I, the truck departs from the warehouse loaded with all required supplies and follows the route 1→2→3 to replenish each distribution center sequentially. Upon arriving at distribution center 1, the truck stationed at that center—carrying supplies and drone—departs to serve demand point A. Simultaneously, the drone takes off from the truck to serve demand point B. Subsequently, both the truck and the drone rendezvous at demand point C, after which the drone lands back on the truck. The same delivery procedure is repeated for distribution centers 2 and 3. After completing all delivery tasks, the truck and drone return to their respective distribution centers.

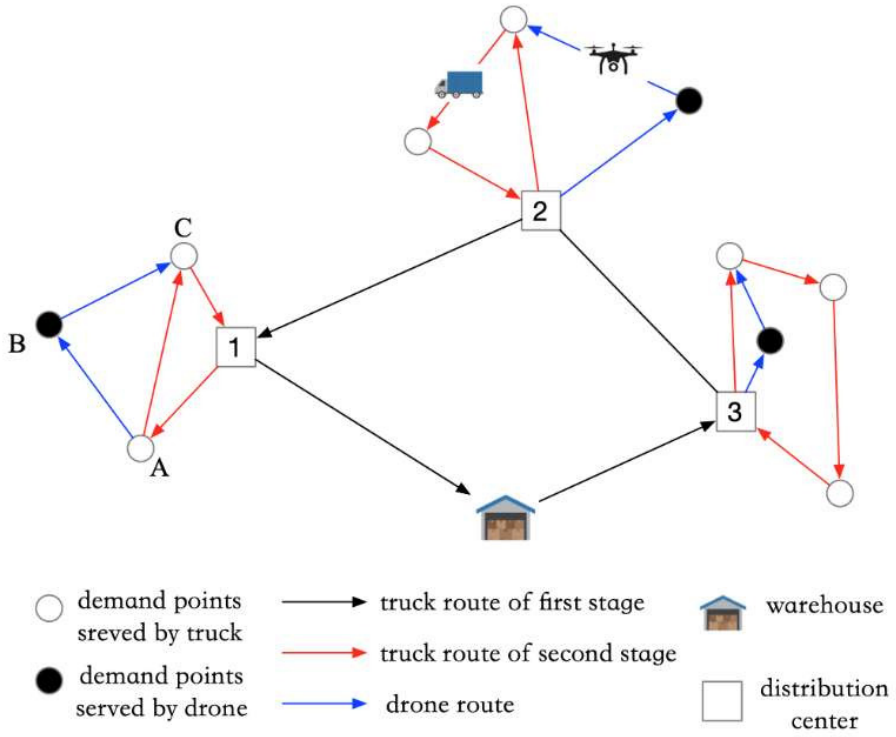


Fig. 1 Two-echelon distribution mode of collaboration between drones and trucks

This problem can be viewed as an extension of the Traveling Salesman Problem (TSP). The model incorporates both an assignment subproblem and a routing subproblem, which involves partitioning demand points into those served by the truck and those served by the drone. Furthermore, it is necessary to decide which demand points will be served by which distribution center. The truck is treated as a traveling salesman, and its route planning constitutes a two-stage TSP problem: the first stage addresses the TSP among distribution centers, while the second stage deals with the TSP among the demand points. Simultaneously, the model must determine the drone's flight route, including the demand point from which the drone launches from the truck, the demand point it serves after launch, and the rendezvous demand point where it returns to the truck after service completion. The synchronization constraint is a critical consideration in truck-drone collaborative delivery, mandating that once the drone launches from the truck to perform a service, it must reunite with the truck before initiating the next service. This necessitates precise coordination between the truck and the drone in both time and space.

3.2 Decision variables and parameters

Decision variables

$x_{ij}^{p} \in \{0,1\}, p \in M, i, j \in N$, whether the truck departing from the p-th distribution center passes through i, j, x_{0j}^{p} indicates whether the truck departs from the distribution center p and serves the demand point j, x_{i0}^{p} indicates whether to return from demand point i to distribution center p.

$y_{ijk}^{p} \in \{0,1\}, p \in M, i, j \in N$, whether the drone departing from the p-th distribution center passes through i, j, k.

$z_{pq} \in \{0,1\}, p, q \in M$, indicates whether there is a path between distribution centers p, q.

$W_{j}^{p} \in \{0,1\}, p \in M, j \in N$, whether the demand point j is served by the drone.

$u_{i}^{p} \in N^{+}, i \in N$, the sequence of truck service demand points.

$ArriveV_{j}^{p} \in R^{+}$, denotes the time when the truck departing from p arrives at j.

$ArriveU_{j}^{p} \in R^{+}$, denotes the time when the drone departing from p arrives at j.

Parameters

M	the set of distribution centers
N	the set of demand points
p, q	index of the distribution center
i, j, k	index of the demand point
d_{ij}	the distance between the points
tv_{ij}	the travel time of the truck between i, j
tu_{ij}	the travel time of the drone between i, j
cv_{ij}	the cost of using the truck between i, j
cu_{ij}	the cost of using the drone between i, j
π_{s}	probability of occurrence of scenario s
d_{i}^{s}	the demand quantity at point i under scenario s
l_{i}	indicates the latest time at which point of demand i needs to be reached

3.3 Mathematical model

The objective function in Eq. 1 indicates that the total distribution cost is the lowest as the objective. In this article, we use time as the cost of the delivery route. When the speed of the transportation vehicle is one unit, the cost of delivery between two points is equal to the distance between the two points. In addition, pd denotes the capacity penalty cost of the distribution center, representing the penalty incurred when unmet demand requires goods to be transferred from other locations, while pt represents the time-window penalty cost associated with violations of time-window constraints. The parameters α and γ represent the coefficients of cost, respectively. The values of these parameters are determined by two main factors. The first factor is their relative importance: a higher value should be assigned to α when greater emphasis is placed on meeting demands within the required time window, while γ should be assigned a higher value when there is a need to balance distribution demands among multiple distribution centers with limited capacities. Additionally, the magnitude scaling of different cost components should be taken into consideration. Both α and γ can serve to adjust for differences in the scales of various costs, thereby preventing any single cost item from becoming excessively large or small and ensuring that it does not disproportionately influence the total distribution cost.

$$\min \sum_{i,j\in N,p\in M} x_{ij}^{p}\, cv_{ij} + \sum_{i,j\in N} y_{ijk}^{p}\left(cu_{ij} + cu_{jk}\right) + \sum_{i,j\in N} z_{pq} cv_{pq} + pd + pt \tag{1}$$

$$pd = \alpha \sum_{p\in M} \max\left\{C_p - \sum_{s\in S,i\in N} \pi_s d_i^s \left(x_{ij}^{p} + y_{ijk}^{p}\right), 0\right\} \tag{1-1}$$

$$pt = \gamma \sum_{i\in N,p\in M} \max\left\{ArriveV_i^{p} - l_i, 0\right\} + \max\left\{ArriveU_i^{p} - l_i, 0\right\} \tag{1-2}$$

$$\sum_{p\in M\cup 0} z_{pq} = 1, q \in M \tag{2}$$

$$\sum_{q\in M\cup 0} z_{pq} = 1, p \in M \tag{3}$$

$$\sum_{p\in M\cup 0} z_{pq} = \sum_{o\in M\cup 0} z_{qo}, \forall\, q \in M \tag{4}$$

$$\sum_{i=0}^{n}\sum_{p=1}^{m} x_{ij}^{p} = 1 - W_j^{p}, j \in N \tag{5}$$

$$\sum_{j=1}^{n+1}\sum_{p=1}^{m} x_{ij}^{p} = 1 - W_j^{p}, i \in N \tag{6}$$

$$\sum_{i=0}^{n} x_{ij}^{p} = \sum_{k=1}^{n+1} x_{jk}^{p}, \forall\, j \in N, \forall\, v \in V \tag{7}$$

$$u_i^{p} - u_j^{p} + 1 \le (c+1)\left(1 - x_{ij}^{p}\right), \forall\, i \in N\_, j \in N^{+}, p \in M \tag{8}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{u=1}^{m} y_{ijk}^{p} = W_j^{p}, \forall\, j \in N \tag{9}$$

$$u_i^{p} - u_k^{p} < m\left(1 - \sum_{j=1}^{n} y_{ijk}^{p}\right) \tag{10}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{u=1}^{m} y_{ijk}^{p} \le \sum_{h=1}^{n}\sum_{v=1}^{m} x_{hi}^{v}, \forall\, i \in N \tag{11}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u=1}^{m} y_{ijk}^{u} \le \sum_{h=1}^{n}\sum_{v=1}^{m} x_{kh}^{v}, \forall\, k \in N \tag{12}$$

$$ArriveV_j^{p} \ge ArriveV_i^{p} + tv_{ij} - M\left(1 - x_{ij}^{p}\right), \forall\, i,j \in N, p \in M \tag{13}$$

$$ArriveU_j^{p} \ge ArriveU_i^{p} + tu_{ij} - M\left(1 - y_{ijk}^{p}\right), \forall\, i,j,k \in N, p \in M \tag{14}$$

$$ArriveU_k^{p} \ge ArriveU_j^{p} + tu_{jk} - M\left(1 - y_{ijk}^{p}\right), \forall\, i,j,k \in N, p \in M \tag{15}$$

$$ArriveU_i^{p} \ge ArriveV_i^{p} - M\left(1 - \sum_{j=1}^{n} y_{ijk}^{p}\right), \forall\, i,k \in N, p \in M \tag{16}$$

Constraints Eqs. 2-4 represent the path from the central warehouse to each distribution center and realize the flow balance constraint. Constraints Eqs. 5-6 indicate that the route of the truck will exist only if the demand point is satisfied by the truck. Constraint Eq. 7 represents the flow balance of the distribution truck in the second stage. Constraint Eq. 8 removes the sub-cycle constraint for the truck in the second stage. Constraint Eq. 9 indicates that the route of the drone will exist only when the demand point is met by the drone. Constraint Eq. 10 limits the delivery direction of drone, that is, the direction of drone delivery demand point should be consistent with the direction of truck travel. Constraints Eqs. 11-12 indicate that only the demand point passed by the truck can allow the drone to take off and land. For example, only the truck serves point i, and point i can be used as the take-off point of the drone. The constraint Eq. 13 is used to calculate the time for the truck to reach point k. Constraints Eqs. 14-15 denote the calculation of the time for the drone to arrive at the served demand point j and the landing point k. Constraint Eq. 16 represents the temporal coordination between the truck and the drone, that is, if the drone takes off at the demand point i, the take-off time of the drone should not be earlier than the time when the truck arrives at the demand point i.

4. Algorithm

4.1 Headings and subheadings

Based on the framework of particle swarm optimization (PSO), this paper proposes a particle swarm optimization heuristic algorithm suitable for the collaborative distribution of truck and drone (PSO-VD) to solve the mixed integer programming model. In each iteration of the algorithm, first, the use of drones is not considered, that is, all demand points are assigned to trucks, and the path of two-stage multi- trucks is calculated. Then, dynamic programming (neighborhood search) is used to reassign some demand points originally served by truck to drone services and a take-off point and landing point are assigned to the drone.

When arranging the route for trucks, the first stage is the sequence from the warehouse to each distribution center, while the second stage each distribution center needs to serve multiple demand points. It is necessary to determine the allocation of distribution centers (that is, which demand point is served by which distribution center) and the two-stage routing arrangement at the same time, so that the total route is the shortest. Because PSO is usually used for continuous space problems, while TSP is a discrete optimization problem. At this time, we need to consider how to represent the solution of these two-stage problems and how to update the position and velocity of particles. Therefore, this paper transforms the route coding of TSP into a continuous mode, which makes it more suitable for using PSO to solve.

In terms of the structure of the solution, the truck route needs to include the visit order of the first stage distribution centers, the allocation of demand points, and the visit order of demand points within each distribution center. The position vector $X[i] = [m + 2n]$ of each particle includes m (number of distribution centers) real numbers for distribution center order, and $2n$ real numbers for the allocation and order of demand points. The total dimension is $m + 2n$. The first m real values are arranged from small to large to get the access order of the distribution center. For example, if the real value corresponding to the distribution center P1 is 0.3, P2 is 0.5, and P3 is 0.1, the order is P3 (0.1), P1 (0.3), and P2 (0.5). demand point assignment and order section, each demand point corresponds to two real numbers, the first determines which transit station to be assigned to, and the second determines the order of access in that distribution center. For each demand point i, the $(m + 2i - 1)$-th real value is used to map to the distribution center. For example, scale the real value x_i to $[0, 1]$, and then determine the distribution center p according to m equal partitions. Assuming that there are 2 distribution centers, then $x(m + 2i - 1) \in (0, 0.5)$ means that the demand point i is in charge of the first distribution center, and $x(m + 2i - 1) \in (0.5, 1)$ means that the demand point i is in charge of the second distribution center. In demand point order calculation, for each distribution center p, all demand points assigned to s are collected, and the order of visits at this distribution center is determined according to the order of size of their $(m + 2i)$-th real value.

After determining the current truck route, it is necessary to clarify the execution sequence of drone tasks, determine the demand points served by drone, select appropriate take-off points and landing points, and then update the path for the next stage of optimization. In terms of route calculation of drone, it needs to be considered that drones can only serve one demand point at a time, and each mission must take off from a certain demand point of the truck and land at another demand point of the truck after service. Every time a mission is completed, the landing point of the drone becomes part of the truck route, and the take-off point of the next mission must be after this landing point. For example, if the drone takes off from demand point 1, serves demand point 2 and lands at demand point 4, then the next route of the truck is 1 → 4, after which the drone can only take off from node 4 or subsequent nodes (such as 5, 9, etc.) to perform the next mission. When performing calculations, we first determine the following parameters, K is the maximum allowable drone route length, t is the number of task calculations, $x_{ij,t} \in \{0, 1\}$, and whether the truck travels from i to j during t calculations, its initial solution is the current position of the particle; at the same time, we denote the take-off point of the t-th mission by $i \in N, k \in N$ represents the landing point of the t-th task, and $j \in N$ represents the demand point served by the t-th iterative task. We need to find a drone path (i, j, k) in the t-th iteration such that $max\{(cv_{ij} + cv_{jk}) - max\{cv_{ik}, cu_{ij} + cu_{jk}\} > 0$, take point j as the point served by the drone and delete it from the path of the delivery truck, when the maximum flight distance of the drone is reached, or $max\{(cv_{ij} + cv_{jk}) - max\{cv_{ik}, cu_{ij} + cu_{jk}\} < 0$, then stops the calculation and updates the truck route.

After the calculation of each particle is completed, the global optimal solution $gbset$ and the optimal solution $pbest[i]$ of each particle can be obtained. At this time, in order to ensure the availability of PSO in the next stage, the optimal solution of the particle should be changed to the truck route under this fitness function, that is, the position of the particle at this time is still $x[i] = x[m + 2n]$, which only includes the distribution center allocation and truck route. For example, before inserting the demand point served by drone, the route of the truck is 1-2-3-4-5.

At this time, enter the next iteration, and use the velocity update formula and position update formula in PSO to update the particles.

Speed update:

$$v_i(t + 1) = v_i(t) + r1\big(pbest - x_i(t)\big) + r2\big(gbest - x_i(t)\big)$$

Location update:

$$x_i(t + 1) = x_i(t) + v_i(t + 1)$$

Now, based on the above analysis, the steps of designing PSO in this paper are as follows:

Step1. Initialize the particle swarm, the position vector of each particle includes m (number of distribution centers) real numbers for transit station order, $2n$ (n demand points, two real numbers per demand point: allocation and order) real numbers. The total dimension is $m + 2n$.

Step 2. For each particle, decode its position:
a) The first m real numbers determine the access order of the distribution center: arrange them according to their values from small to large to get a sequential list of distribution centers.
b) For each demand point i, calculate its assigned distribution centers: take the real value of the $(m + 2i - 1)$-th dimension, map it to $[0, 1]$, and then divide it into m intervals to determine the corresponding distribution centers.
c) For each distribution centers p, collect all demand points assigned to p, and then determine their access order at the distribution center according to the magnitude of the real value of the $(m + 2i)$-th dimension of these demand points.

Step 3. Insert the route of the drone into the route of each delivery and update the route of the truck.

Step 4. Calculate fitness (total path length):
a) Calculate the first stage path: start from the warehouse, visit it in sequence according to the decoded distribution center order, and then return to the TSP route length of the warehouse.
b) For each distribution center p, calculate the TSP route length of its demand point visit order: start from distribution center p, visit according to the decoded truck and drone route sequence, and then return to the route length of p.
c) The total fitness is the sum of the route lengths of the two stages.

Step 5. Update *pbest* and global *gbest* for each particle.

Step 6. Update the velocity and position of each particle according to the PSO velocity and position update formula.

Step 7. Repeat steps 2-6 until the termination condition is met.

5. Numerical experiments

5.1 Algorithm performance

In this section, we use three groups of cases to conduct numerical experiments, 15 random trials are conducted for each group of cases and directly compare the solution results of the heuristic algorithm with those of CPLEX, so as to verify the effectiveness of our proposed algorithm. In terms of parameter generation, in the three groups of cases, we assume that there is a (1000, 1000) area, (0, 0) is the central warehouse, and there are two distribution centers with random locations and capacity of (10, 50). There are 8, 10, and 12 demand points in the area, with random locations. Assuming the demand at each point follows a uniform distribution with a range of (1, 10), we use the mathematical expectation of demand as the actual demand for the point to handle the scenario of random demand, and the time window is a random integer of 0-3000. The speed of the drones is twice that of the trucks, and the travel time of the trucks is 1 unit. We use the distance between two points as the transportation cost, and both the time penalty cost coefficient and the capacity penalty coefficient are set to 1. In the setting of particle swarm optimization parameters, the algorithm uses Python for encoding, the number of particles is set to 50, and the number of iterations is 1000.

Table 1 Comparison of CPLEX and PSO-VD solution results with 8 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj	Time (s)	
1	8	4867.34	64.35	5281.22	63.91	7.84
2		4565.41	19.65	4630.15	62.06	1.40
3		4864.18	15.06	5078.80	60.78	4.23
4		5333.14	14.66	5905.38	60.12	9.69
5		6930.11	76.57	7914.67	55.43	12.44
6		5599.33	25.06	5740.81	48.25	2.46
7		4995.21	18.90	6631.94	58.93	24.68
8		6825.00	88.75	8344.14	60.10	18.21
9		4816.82	10.26	4819.56	59.22	0.06
10		4219.41	13.23	4906.92	54.36	14.01
11		4287.20	59.33	4544.19	55.67	5.66
12		4737.35	63.07	5615.66	54.82	15.64
13		4932.30	37.49	4998.69	57.99	1.33
14		6093.29	37.25	7096.12	62.15	14.13
15		4582.94	20.67	4643.63	51.87	1.31
AVE			37.62		57.71	8.87

Table 2 Comparison of CPLEX and PSO-VD solution results with 10 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj.	Time (s)	
16	10	4696.49	24.00	4915.70	99.33	4.46
17		22747.20	143.69	39044.15	94.14	41.74
18		5821.90	492.41	6259.63	98.58	6.99
19		5508.59	139.35	6073.98	86.99	9.31
20		10257.00	435.76	12484.22	109.54	17.84
21		5423.98	916.37	5515.46	104.04	1.66
22		4679.57	144.78	4911.54	104.69	4.72
23		7391.52	529.75	10495.42	96.81	29.57
24		4399.35	20.55	5400.09	96.25	18.53
25		5424.60	135.92	5895.59	83.22	7.99
26		10379.20	597.43	14956.40	86.23	30.60
27		5303.12	121.43	5449.65	96.76	2.69
28		4986.49	130.42	5437.93	81.88	8.30
29		5967.45	1449.71	6078.02	90.70	1.82
30		4988.55	684.39	5941.99	103.77	16.05
AVE			397.73		95.53	13.49

Table 3 Comparison of CPLEX and PSO-VD solution results with 12 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj	Time (s)	
31	12	5779.91	3832.27	6086.39	168.70	5.04
32		4969.05	2104.76	5133.52	132.09	3.20
33		5268.61	1633.16	5901.28	117.92	10.72
34		6631.93	5145.78	7712.96	155.69	14.02
35		5098.44	905.76	5657.63	211.16	9.88
36		5878.99	4869.61	7220.25	180.32	18.58
37		6611.72	5560.58	8571.16	183.96	22.86
38		6408.76	1229.02	7113.89	150.48	9.91
39		12457.80	4639.87	20612.16	162.72	39.56
40		7074.39	9653.25	7312.15	162.68	3.25
41		5779.91	4543.35	6086.39	140.69	5.04
42		22321.60	3581.06	25099.58	168.81	11.07
43		5787.84	7668.67	6760.20	125.29	14.38
44		5273.36	1822.55	5593.84	166.34	5.73
45		5678.51	1349.59	5993.26	147.47	5.25
AVE			3902.62		158.29	11.90

Tables 1 to 3 show the comparative analysis of three sets of experiments with a total of 45 cases. In these tables, MIP represents the scenario of direct solution using the CPLEX commercial solver; PSO-VD represents the scenario of solution using the heuristic algorithm designed in this paper; N denotes the number of demand points, Obj. stands for the objective value of the solution result; and Time(Sec) indicates the runtime of the code; δ is the gap between the optimal solution and the PSO-VD solution results, and AVE is the average solution time and average solution gap of 15 sets of cases. When there are only 8 demand points, the optimal solution of CPLEX can be obtained in a short time, the runtime of the code of PSO-VD is slightly longer than that of CPLEX, and the gap between the heuristic algorithm and the exact optimal solution is 8.87 %. When there are 10 demand points, the average runtime of the code of CPLEX is 397.73 seconds, the runtime of the code of PSO-VD is 95.53 seconds, and the gap between the heuristic algorithm and the exact optimal solution is 13.49 %. When there are 12 demand points, the runtime of the code of CPLEX increases to 3902.62 seconds, which is almost not allowed in emergency logistics,

because there is a certain suddenness, and emergency logistics often needs to respond quickly. The runtime of the code of PSO-VD is about 150 seconds, and the increase of the runtime of the code is basically linear with the increase of demand point. The difference between the heuristic algorithm and the exact optimal solution is 11.90 %. An increase in the number of demand points does not lead to a deterioration in solution quality. When the number of demand points becomes large, directly applying commercial solvers such as CPLEX results in excessive computational time. The two-stage particle swarm heuristic algorithm proposed in this paper can greatly shorten the solution time and ensure the solution quality. It provides a decision-making tool for the use of drones and trucks for emergency supplies distribution after an emergency event.

Table 4 Solution results with 50 demand points

Instance	Obj.	CW	CI	ND	Time (s)
1	4944.76	1777.80	0.00	8	147.19
2	6085.32	2020.49	0.00	7	150.33
3	4805.52	1574.88	274.87	8	167.93
4	4265.48	1105.71	246.01	9	138.14
5	21223.89	2706.64	7123.06	9	143.62
6	3530.08	1057.61	0.00	5	119.41
7	23939.49	1293.60	10134.23	7	136.86
8	6754.44	1177.63	1298.65	8	127.89
9	20663.29	1469.28	7883.07	8	160.12
10	7938.67	2749.23	31.18	7	152.60

Table 5 Solution results with 100 demand points

Instance	Obj.	CW	CI	ND	Time (s)
1	48957.25	6480.77	16208.92	9	537.90
2	64206.56	8591.16	21824.23	9	788.62
3	57583.76	7630.96	19532.70	11	848.06
4	76037.76	5468.44	31158.02	9	863.16
5	31473.14	5417.34	8685.06	13	760.33
6	30708.04	5924.46	7807.30	14	694.09
7	44349.27	6020.26	13501.16	13	822.34
8	64874.75	7635.61	21775.38	14	843.22
9	48225.69	7221.32	14519.01	11	812.16
10	48986.75	5096.46	16697.15	14	779.57

In this section, following the procedure of the heuristic algorithm, we first conducted computations for scenarios involving large-scale demand points and presented the results, thereby verifying the broad applicability of the proposed algorithm. Given that the heuristic algorithm proposed in this paper includes several randomly generated steps, the experimental process was repeated ten times for different scales of demand points to ensure the robustness of the heuristic approach. Furthermore, we considered the presence of three distribution centers within the region, with the remaining parameter settings consistent with those in the previous section. Tables 4 and 5 present the case analysis results for 50 and 100 demand points, respectively. For each demand point scale, the coordinates of the demand points were randomly generated ten times. In Tables 4 and 5, the first column indicates the case number, followed by *Obj.* representing the solution outcome—specifically, the minimized delivery time, time window penalty cost, and capacity penalty cost. CW denotes the time window penalty cost incurred due to the inability to meet the time window requirements of demand points, while Cl represents the distribution center capacity penalty cost resulting from the failure to fully satisfy the demand of certain points; ND denotes the number of demand points served by drones. The last column rec-

ords the code running time. From the results presented in Tables 4 through 5, it can be observed that the proposed heuristic algorithm can obtain solutions relatively quickly. In the case of 50 demand points, the algorithm requires approximately two minutes to produce a solution, whereas for 100 demand points, it takes about ten minutes. As an NP-hard problem, the computational time of the proposed algorithm significantly surpasses that of commercial solvers, albeit with a slight trade-off in solution accuracy. In the context of humanitarian logistics, there is often a pressing need for rapid decision-making. Therefore, the model and algorithm proposed in this paper demonstrate considerable practical utility.

5.2 Sensitivity analysis

Time window penalty cost coefficient α

In this set of experiments, this paper examines the impact of the time window penalty cost coefficient α on the solution outcomes. α represents the importance of the time window, and different ranges can be assigned depending on the type of goods being delivered. For instance, a higher α value can be set for urgent supplies such as water and medicines, whereas a lower α can be used for ordinary goods.

Table 6 Sensitivity analysis results of time window penalty coefficient

α	Instance	Obj.	CW	CI	ND
0.5	1	7750.12	2585.32	1199.31	7
	2	2691.99	1049.21	0	8
	3	3182.84	1343.16	0	8
	4	18007.03	1387.50	7283.77	9
	5	14235.91	1286.79	5408.63	9
	AVE	9173.58	1530.50	2778.34	8.2
1	1	12695.17	827.51	4701.94	9
	2	4114.84	1260.34	0	8
	3	13335.70	1028.90	0	8
	4	4004.98	1211.41	0	9
	5	5421.97	1884.95	0	8
	AVE	7914.53	1242.62	940.38	8.4
2	1	9028.48	1786.24	0	9
	2	3995.90	841.12	0	5
	3	7304.66	1529.77	0	7
	4	12048.78	2606.94	1116.98	8
	5	32193.36	1154.40	13536.57	8
	AVE	12914.23	1583.69	2930.71	7.4
5	1	8934.08	1210.13	0	7
	2	11150.71	938.41	0	8
	3	21912.31	1239.09	5601.84	7
	4	18108.31	1612.05	1448.62	6
	5	18903.77	3391.60	1556.16	5
	AVE	15801.83	1678.25	1721.32	6.6

As shown in Table 6, as the time window penalty cost coefficient α gradually increases, the frequency of drone usage actually decreases. When α is 0.5, an average of 8.2 demand points are served by drones. At α = 2, drones serve 7.4 demand points, and when α reaches 5, only 6.6 demand points are served by drones. We analyze that this is likely because demand points with stricter time requirements are more sensitive to delivery reliability, making trucks a more suitable choice to ensure dependable service. Since drones operate in coordination with trucks—requiring pickup from trucks before continuing delivery—they may not be able to handle more complex or time-critical demands. Therefore, in practical humanitarian emergency logistics and

the distribution of relief supplies, drones may still play only a supporting role. For the delivery of urgent supplies, trucks remain the most reliable option.

Drone battery capacity

This study investigates the impact of drone battery capacity on delivery outcomes. To simplify the problem and control for the influence of cargo weight on the results, the maximum travel distance of the drone is used as a substitute for battery capacity, while both cargo weight and the drone's own weight are disregarded.

Table 7 Sensitivity analysis results of drone battery capacity

Drone battery capacity	Instance	Obj.	CW	CI	ND
50	1	26631.77	1477.61	11147.77	3
	2	25720.24	2701.90	14243.07	5
	3	27182.15	1364.25	16512.97	5
	4	22107.84	992.58	9161.18	5
	5	32106.39	1210.91	14111.32	6
	AVE	26749.68	1549.45	9035.26	4.8
100	1	23161.46	1458.10	9173.11	7
	2	28082.45	1845.50	11258.03	7
	3	26697.90	2346.76	5058.88	6
	4	25201.54	1307.82	9462.57	7
	5	25044.44	1007.07	10622.33	8
	AVE	25637.56	1593.05	9114.98	7
500	1	25791.63	1668.10	10513.68	9
	2	22520.63	1818.97	7480.15	7
	3	21125.52	1307.82	7462.57	6
	4	23893.09	1007.07	8622.33	8
	5	25044.44	1007.07	10622.33	8
	AVE	23675.06	1361.81	8940.21	7.6
1000	1	19771.22	1060.81	8164.78	9
	2	23722.23	2267.25	13853.96	7
	3	17212.17	1507.37	6118.70	6
	4	23629.79	1807.33	7371.58	8
	5	26705.28	1241.84	11451.42	7
	AVE	22208.14	2074.64	9392.09	7.4
2000	1	23809.98	1909.41	9297.58	7
	2	22668.43	2184.73	8418.20	9
	3	27672.10	1512.33	11126.80	8
	4	13709.28	1075.12	4824.58	8
	5	18238.48	447.06	7443.35	7
	AVE	21219.65	1425.73	8222.10	7.8

The battery capacity—expressed as the maximum allowed travel distance—is set at 50, 100, 500, 1000, and 2000 units, respectively. As shown in Table 7, since terrain and road conditions do not impose constraints, the operational cost of drones remains relatively low. When the battery capacity increases, drones are able to serve more points, thereby reducing the overall delivery cost. However, it should also be noted that once the battery capacity reaches a certain threshold, the frequency of drone deployment no longer increases. Specifically, when the battery capacity rises from 50 to 500, the number of demand points served by drones increases from 4.8 to 7.6. Beyond this point, further increases in battery capacity result in almost no change in the number of demand points served. Therefore, in humanitarian logistics, rationally selecting drones with appropriate parameters can enhance logistics efficiency while ensuring cost-effectiveness.

Drone speed

Since drone flight is not constrained by terrain and can generally follow a straight-line path and given that drones typically have a higher speed than trucks, even though drones serve only as an auxiliary delivery tool in this distribution mode, this paper posits that an increase in drone flight speed can significantly reduce total delivery costs. Therefore, this subsection focuses on drone speed as the subject of sensitivity analysis. Assuming four speed ratios between the truck and the drone—1:1, 1:2, 1:5, and 1:10—the model was solved for five randomly generated sets of demand point coordinates to examine the impact of drone speed on the solution outcomes. The speed of the truck is set to 1 unit. Each demand point set comprises 30 demand points, with all other parameters consistent with previous settings.

Table 8 presents the solution results. When the speed ratio between the truck and the drone is 1:1, the average delivery cost is 26,720.68, and on average only 6.6 demand points are served by drones per delivery. When the speed ratio increases to 1:2, the total delivery cost decreases to 25637.56, and drone utilization improves, with an average of 8.2 demand points served by drones.

Table 8 Sensitivity analysis results of drone speed

Truck speed: Drone speed	Instance	Obj.	CW	CI	ND
1:10	1	18373.86	1472.61	6921.84	9
	2	23188.60	1335.77	9424.22	8
	3	19753.17	1574.46	7477.95	8
	4	15435.91	1251.83	5292.83	9
	5	30953.23	1675.55	12832.72	8
	AVE	21540.95	1462.04	8389.91	8.4
1:5	1	18331.24	1143.74	2979.78	8
	2	17939.15	1626.49	6360.60	10
	3	18242.59	1874.39	5790.26	9
	4	24846.43	2333.73	8905.28	6
	5	24472.16	1347.13	9955.68	9
	AVE	20766.31	1665.09	6798.32	8.4
1:2	1	23579.75	1788.26	8952.47	8
	2	20085.72	1380.06	7987.17	10
	3	20023.24	1263.56	7978.57	8
	4	10703.64	1114.93	3323.57	7
	5	19371.06	1877.38	6872.05	8
	AVE	18752.68	1484.84	7022.77	8.2
1:1	1	23089.46	2205.97	8395.21	8
	2	26293.74	1424.03	10894.12	6
	3	22341.35	1940.27	8398.79	6
	4	29984.24	972.74	13358.06	7
	5	31894.64	1028.21	14253.35	6
	AVE	26720.69	1514.24	11059.91	6.6

However, as drone speed continues to increase—specifically when the speed ratio of drone to truck reaches 1:5 and even 1:10—the solution outcomes do not show further improvement, nor does the frequency of drone usage increase significantly. This implies that appropriate transportation modes should be selected based on practical circumstances, considering factors such as technological upgrade costs or procurement expenses. A blind pursuit of higher performance parameters may not yield substantial practical benefits.

6. Conclusion

This study investigates a two-echelon truck-drone collaborative routing problem for humanitarian logistics under stochastic demand and time-window constraints. The problem is structured in two stages: first, trucks transport emergency supplies from a central warehouse to multiple distribution centers; second, each distribution centre deploys a truck equipped with a drone to serve demand points in a coordinated manner. The drone may take off from the truck when it stops at a certain demand point, to deliver to another point, and later land back on the truck when the truck arrives at a subsequent point, thereby enhancing delivery flexibility and coverage. To address this problem, we formulate a mixed-integer programming model that aims to minimize the total distribution cost, incorporating transportation cost, capacity-shortage penalty, and time-window violation penalty. Building on the particle swarm optimization framework, we propose a novel heuristic algorithm named PSO-VD, which integrates particle swarm optimization with a drone-insertion mechanism based on the 2-opt method. This approach transforms the discrete routing problem into a continuous encoding scheme, enabling an efficient search within the solution space. Numerical experiments on small-scale instances demonstrate that, compared with CPLEX, PSO-VD achieves near-optimal solutions with an average optimality gap between 8.87 % and 13.49 %, while significantly reducing computational time. For larger-scale instances, the algorithm remains practical, obtaining feasible solutions within minutes—a crucial capability for emergency response scenarios. Furthermore, sensitivity analyses are conducted on parameters such as the time-window penalty coefficient, drone battery capacity, and drone speed, yielding several managerial insights. Stricter time windows reduce the deployment of drones, indicating that trucks are preferred for time-critical deliveries. Drone battery capacity improves delivery efficiency up to a certain threshold, beyond which further capacity increases yield diminishing returns. Although higher drone speed moderately enhances system performance, excessively high speeds do not lead to substantial improvements, suggesting that cost-effective technology selection is essential.
Future research could focus on enhancing the algorithm through hybrid meta-heuristics, incorporating more dynamic or realistic constraints (e.g., weather-dependent drone status), or developing decomposition-based exact methods for medium-scale instances.

References

[1] Perez-Grau, F.J., Martinez-de Dios, J.R., Paneque, J.L., Acevedo, J.J., Torres-González, A., Viguria, A., Astorga, J.R., Ollero, A. (2021). Introducing autonomous aerial robots in industrial manufacturing, *Journal of Manufacturing Systems*, Vol. 60, 312-324, doi: 10.1016/j.jmsy.2021.06.008.

[2] Gomes, O., Lins de Moraes, M. (2024). Forks in the road: Modelling the economic prospects of artificial intelligence, *Economic Computation and Economic Cybernetics Studies and Research*, Vol. 58, No. 3, 113-128, doi: 10.24818/18423264/58.3.24.07.

[3] Zhang, H.Q. (2024). Optimizing obstacle avoidance path planning for intelligent mobile robots in multi-obstacle environments, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 358-370, doi: 10.14743/apem2024.3.512.

[4] Goyal, S.B., Rajawat, A.S., Solank, R.K., Patil, D., Potgantwar, A. (2024). A trustable security solution using XAI for 5G-enabled UAV, *Journal of Logistics, Informatics and Service Science*, Vol. 11, No. 4, 73-86, doi: 10.33168/JLISS.2024.0404.

[5] Murray, C.C., Chu, A.G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, *Transportation Research Part C: Emerging Technologies*, Vol. 54, 86-109, doi: 10.1016/j.trc.2015.03.005.

[6] Carlsson, J.G., Song, S. (2018). Coordinated logistics with a truck and a drone, *Management Science*, Vol. 64, No. 9, 4052-4069, doi: 10.1287/mnsc.2017.2824.

[7] Agatz, N., Bouman, P., Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone, *Transportation Science*, Vol. 52, No. 4, 965-981, doi: 10.1287/trsc.2017.0791.

[8] Yurek, E.E., Ozmutlu, H.C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone, *Transportation Research Part C: Emerging Technologies*, Vol. 91, 249-262, doi: 10.1016/j.trc.2018.04.009.

[9] Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H. (2018). On the min-cost traveling salesman problem with drone, *Transportation Research Part C: Emerging Technologies*, Vol. 86, 597-621, doi: 10.1016/j.trc.2017.11.015.
[10] Kim, S.J., Lim, G.J., Cho, J. (2018). Drone flight scheduling under uncertainty on battery duration and air temperature, *Computers & Industrial Engineering*, Vol. 117, 291-302, doi: 10.1016/j.cie.2018.02.005.
[11] Roberti, R., Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone, *Transportation Science*, Vol. 55, No. 2, 315-335, doi: 10.1287/trsc.2020.1017.
[12] Tamke, F., Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones, *Transportation Research Part B: Methodological*, Vol. 144, 174-203, doi: 10.1016/j.trb.2020.11.011.
[13] Vásquez, S.A., Angulo, G., Klapp, M.A. (2021). An exact solution method for the TSP with drone based on decomposition, *Computers & Operations Research*, Vol. 127, Article No. 105127, doi: 10.1016/j.cor.2020.105127.
[14] Kang, M., Lee, C. (2021). An exact algorithm for the heterogeneous drone-truck routing problem, *Transportation Science*, Vol. 55, No. 5, 1088-1112, doi: 10.1287/trsc.2021.1055.
[15] Ham, A.M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming, *Transportation Research Part C: Emerging Technologies*, Vol. 91, 1-14, doi: 10.1016/j.trc.2018.03.025.
[16] Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach, *Computers & Industrial Engineering*, Vol. 129, 14-30, doi: 10.1016/j.cie.2019.01.020.
[17] Murray, C.C., Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones, *Transportation Research Part C: Emerging Technologies*, Vol. 110, 368-398, doi: 10.1016/j.trc.2019.11.003.
[18] Roberti, R., Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone, *Transportation Science*, Vol. 55, No. 2, 315-335, doi: 10.1287/trsc.2020.1017.
[19] Salama, M.R., Srinivas, S. (2022). Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 164, Article No. 102788, doi: 10.1016/j.tre.2022.102788.
[20] Chen, H., Hu, Z., Solak, S. (2021). Improved delivery policies for future drone-based delivery systems, *European Journal of Operational Research*, Vol. 294, No. 3, 1181-1201, doi: 10.1016/j.ejor.2021.02.039.
[21] Lichau, S., Sadykov, R., François, J., Dupas, R. (2025). A branch-cut-and-price approach for the two-echelon vehicle routing problem with drones, *Computers & Operations Research*, Vol. 173, Article No. 106869, doi: 10.1016/j.cor.2024.106869.
[22] Saleu, R.G.M., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A. (2022). The parallel drone scheduling problem with multiple drones and vehicles, *European Journal of Operational Research*, Vol. 300, No. 2, 571-589, doi: 10.1016/j.ejor.2021.08.014.
[23] Ermağan, U., Yıldız, B., Salman, F.S. (2022). A learning-based algorithm for drone routing, *Computers & Operations Research*, Vol. 137, Article No. 105524, doi: 10.1016/j.cor.2021.105524.
[24] Stodola, P., Kutěj, L. (2024). Multi-depot vehicle routing problem with drones: Mathematical formulation, solution algorithm and experiments, *Expert Systems with Applications*, Vol. 241, Article No. 122483, doi: 10.1016/j.eswa.2023.122483.
[25] Ramadhan, F., Irawan, C.A., Salhi, S., Cai, Z. (2025). The truck traveling salesman problem with drone and boat for humanitarian relief distribution in flood disaster: Mathematical model and solution methods, *European Journal of Operational Research*, Vol. 322, No. 1, 270-291, doi: 10.1016/j.ejor.2024.10.022.
[26] Gao, X., Jin, X., Zheng, P., Cui, C. (2021). Multi-modal transportation planning for multi-commodity rebalancing under uncertainty in humanitarian logistics, *Advanced Engineering Informatics*, Vol. 47, Article No. 101223, doi: 10.1016/j.aei.2020.101223.
[27] Gao, Y., Ding, X., Yu, W. (2024). Distributional robustness based on Wasserstein-metric approach for humanitarian logistics problems under road disruptions, *Operations Research Perspectives*, Vol. 13, Article No. 100317, doi: 10.1016/j.orp.2024.100317.
[28] Ahmadi, M., Seifi, A., Tootooni, B. (2015). A humanitarian logistics model for disaster relief operations considering network failure and standard relief time: A case study on San Francisco district, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 75, 145-163, doi: 10.1016/j.tre.2015.01.008.
[29] Hou, D.N., Liu, S.C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 315-332, doi: 10.14743/apem2024.3.509.
[30] Yang, X., Cao, W., Wang, K., Yin, H., Wu, J., Wu, L. (2025). Integrated scheduling of truck and drone fleets for cargo transportation in post-disaster relief: A two-stage stochastic optimization approach, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 196, Article No. 104015, doi: 10.1016/j.tre.2025.104015.
[31] Lu, Y., Yang, J., Yang, C. (2023). A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions, *Computers & Industrial Engineering*, Vol. 184, Article No. 109563, doi: 10.1016/j.cie.2023.109563.
[32] Yin, Y., Yang, Y., Yu, Y., Wang, D., Cheng, T. C. E. (2023). Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics, *Transportation Research Part B: Methodological*, Vol. 174, Article No. 102781, doi: 10.1016/j.trb.2023.102781.

[33] Kyriakakis, N.A., Sevastopoulos, I., Marinaki, M., Marinakis, Y. (2022). A hybrid Tabu search–variable neighborhood descent algorithm for the cumulative capacitated vehicle routing problem with time windows in humanitarian applications, *Computers & Industrial Engineering*, Vol. 164, Article No. 107868, doi: 10.1016/j.cie.2021.107868.
[34] Van Steenbergen, R., Mes, M., Van Heeswijk, W. (2023). Reinforcement learning for humanitarian relief distribution with trucks and UAVs under travel time uncertainty, *Transportation Research Part C: Emerging Technologies*, Vol. 157, Article No. 104401, doi: 10.1016/j.trc.2023.104401.
[35] Amirsahami, A., Barzinpour, F., Pishvaee, M.S. (2025). A fuzzy programming model for decentralization and drone utilization in urban humanitarian relief chains, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 195, Article No. 103949, doi: 10.1016/j.tre.2024.103949.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 369–379
https://doi.org/10.14743/apem2025.3.546

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Precision blade manufacturing: Small-sample prediction and optimization using improved meta-learning and Particle Swarm Optimization

Zhang, L.[a], **Wang, Q.**[a], **Xia, Y.T.**[a,*], **Xia, Y.L.**[b]

[a]School of Electronic Information Engineering, Geely University, Sichuan, P.R. China
[b]School of Mechanical Engineering, Sichuan University, Sichuan, P.R. China

ABSTRACT

Accurately predicting blade manufacturing deviations from limited experimental data remains challenging due to the complex nonlinear relationship between process parameters and resulting profile deviations in precision casting. To overcome the limitations inherent in traditional approaches and conventional machine learning methods, this study proposes a novel prediction and optimization framework specifically designed for small-sample scenarios, integrating enhanced meta-learning optimization with advanced Particle Swarm Optimization (PSO). We innovatively improve the model-agnostic meta-learning (MAML) algorithm by incorporating a dynamic loss function weighting strategy and a stochastic gradient descent with warm restarts (SGDR) learning rate mechanism, significantly mitigating overfitting and enhancing generalization performance. Additionally, we propose a process parameter optimization model utilizing an improved PSO algorithm with dynamic inertia and adaptive learning factors, designed to effectively navigate high-dimensional optimization landscapes. Experimental validation using orthogonal design data highlights pulling speed as the dominant factor influencing blade deviations (Pearson correlation coefficient ($r = 0.67$). The optimized parameters—low pulling speed (1.5 mm/min) and high pouring temperature (1530 ℃)—achieve an 11.54 % reduction in blade deformation. The improved MAML-based prediction model demonstrates superior accuracy, achieving a mean absolute error (MAE) of 2.566×10^{-4} mm, representing a 21.7 % improvement over traditional Adam optimization methods, and exhibits robust predictive capability ($R^2 = 0.92$) in small-sample contexts. This research not only delivers practical insights and precise parameter recommendations for complex blade manufacturing processes but also establishes a robust methodological framework applicable broadly to precision manufacturing domains characterized by limited data availability.

ARTICLE INFO

Keywords:
Precise manufacturing;
Optimization;
Meta-learning optimization;
Machine learning;
Small sample learning;
Particle Swarm Optimization (PSO)

**Corresponding author:*
LinZhang_edu@163.com
(Xia, Y.T.)

Article history:
Received 7 April 2025
Revised 25 August 2025
Accepted 29 August 2025

1. Introduction

As a core component of aircraft engines and gas turbines, blades play a critical role in determining an engine's aerodynamic performance, thermal efficiency, and reliability [1-3]. However, in the directional solidification process, the complex interactions among process parameters—such as drawing speed, pouring temperature, and mold preheating temperature—can cause nonlinear deformations in the blade profile, leading to manufacturing deviations [4-5]. Traditional process optimization methods primarily rely on numerical simulations and trial-and-error approaches. However, due to the high computational cost and long experimental cycles, it is challenging to

develop high-accuracy predictive models with limited sample data [6-8]. In recent years, machine learning techniques have been introduced into process optimization, but they face challenges such as overfitting and poor generalization in small-sample scenarios [9-12]. Therefore, efficiently predicting manufacturing deviations and optimizing process parameter combinations based on limited sample data has become a key issue in improving blade manufacturing accuracy.

In the field of manufacturing process modeling, researchers worldwide have proposed various methods. For example, Del Vecchio *et al*. [13] applied a support vector–based learning approach to optimize process parameters in superalloy investment casting. Gupta *et al*. [14] developed a deep learning–based model for defect analysis in casting processes using convolutional neural networks. However, many existing approaches typically rely on large amounts of training data, which limits their applicability in data-scarce scenarios. To address the small-sample problem, meta-learning has gained attention for its ability to rapidly adapt to new tasks [15-17]. The model-agnostic meta-learning (MAML) approach proposed by Finn *et al.* [18] optimizes initial parameters through multi-task learning, demonstrating superior performance with limited data. However, its dynamic optimization mechanism for complex process parameter mappings remains underexplored. Additionally, existing meta-learning methods often adopt fixed learning rates and mean squared error (MSE) loss functions, making it difficult to accommodate varying task complexities in manufacturing deviation prediction [19-21].

In the domain of process parameter optimization, the Particle Swarm Optimization (PSO) algorithm is widely used for multi-objective optimization due to its strong global search capability [22-24]. The classical PSO algorithm, introduced by Kennedy *et al.* [25], simulates the foraging behavior of bird flocks to search for optimal solutions. However, it tends to fall into local optima when applied to high-dimensional complex response surfaces. To address this, Du *et al.* [26] achieved resource optimization by improving the ant colony algorithm, emphasizing the enhancement of global search capability through algorithmic parameter adjustment. Ratnaweera *et al.* [27] designed an adaptive learning factor to enhance convergence efficiency. However, most PSO improvements focus on general optimization problems and lack targeted designs for the collaborative optimization of manufacturing process parameters. Moreover, when machine learning models are used as the fitness function in PSO, prediction errors may mislead the optimization direction, necessitating additional constraints based on process mechanisms [28-30]. Although these studies provide valuable insights into predicting and optimizing blade manufacturing deviations, traditional machine learning methods rely on large datasets and struggle to accurately model the nonlinear relationship between process parameters and profile deviations in small-data scenarios. Furthermore, existing meta-learning approaches do not incorporate loss functions and learning rate strategies tailored to the dynamic characteristics of manufacturing tasks, limiting their generalization ability. Additionally, PSO is prone to local optima and error accumulation in high-dimensional process parameter optimization and lacks a deeply integrated framework with meta-learning models.

Traditional optimization methods often suffer from issues such as convergence to local optima. To address this limitation, this study proposes a small-sample prediction and process-parameter co-optimization framework that integrates meta-learning with the PSO algorithm. By improving the MAML algorithm, we introduce a dynamic loss function weight adjustment strategy that adaptively scales the mean squared error loss based on task complexity. Additionally, we incorporate a stochastic gradient descent with warm restarts (SGDR) learning rate annealing mechanism, which periodically resets the learning rate to mitigate optimization stagnation, significantly enhancing the model's generalization ability in small-sample scenarios. Furthermore, we design a dynamic PSO framework with a nonlinear dynamic inertia factor and an adaptive learning factor strategy to strengthen the algorithm's global search capability in high-dimensional response surfaces. By embedding the meta-learning model into the PSO fitness function, we establish a closed-loop feedback mechanism for process parameter optimization and deviation prediction. Based on orthogonal experimental data, Pearson correlation analysis, one-way ANOVA, and interaction effect plots quantitatively reveal that pulling speed has a dominant impact on manufacturing deviations ($r = 0.67$). The study further identifies an optimal process parameter combination—low pulling speed (1.5 mm/min) and high pouring temperature (1530 °C)—which reduces overall

blade deformation by 11.54 %. This research not only establishes a theoretical framework for small-sample-driven complex process optimization but also extends its applicability to high-precision intelligent manufacturing. It provides a novel technological pathway for improving the manufacturing quality of critical industrial components.

2. Research models

The numerical simulation of blade directional solidification is computationally expensive, limiting the availability of experimental data. Traditional deep learning methods require large datasets to avoid overfitting, while conventional machine learning struggles with sparse data. Meta-learning, specifically model-agnostic meta-learning, is uniquely suited to this context because it trains models to generalize across tasks by learning a shared initialization from limited data. In small-sample scenarios, meta-learning simulates multiple sub-tasks during the training phase, deeply extracting information from sparse data, allowing the model to quickly adapt to new tasks with fewer samples, while maintaining strong generalization capabilities.

Consider a model $f(\theta)$ that maps input x to output y. Let $p(\tau)$ represent the task distribution, from which K^{spt} support set tasks $\tau_i^{spt} \sim p(\tau)$ and K^{spt} query set tasks are sampled for each epoch $\tau_i^{qry} \sim p(\tau)$. The support set is used for performing gradient descent operations on the base learner during the inner iteration, while the query set is used to evaluate the results of the inner optimization. For the support sets in the training set, the loss function is computed, and one or more gradient descent steps $\nabla_\theta L_{\tau_i^{spt}}(f(\theta))$ are performed during the inner iteration. In the inner optimization process, for a new task τ_i, a copy of the parameters θ_i' is created, and the update for θ_i' is as follows:

$$\theta_i' = \theta - \alpha \nabla_\theta L_{\tau_i^{spt}}\left(f(\theta)\right) \tag{1}$$

Here, α represents the learning rate of the base learner. For all query set tasks $\tau_i^{qry} \sim p(\tau)$, the corresponding loss function value for the parameters θ_i' is computed. Therefore, the objective function of meta-learning can be expressed as:

$$\min_\theta \sum_{\tau_i^{qry} \sim p(\tau)} L_{\tau_i^{qry}}(f(\theta_i')) = \min_\theta \sum_{\tau_i^{qry} \sim p(\tau)} L_{\tau_i^{qry}} \left\{ f\left[\theta - \alpha \cdot \nabla_\theta L_{\tau_i^{spt}}(f(\theta))\right]\right\} \tag{2}$$

Meta-learning is performed across tasks using a stochastic gradient descent (SGD)–based meta-optimization strategy:

$$\theta = \theta - \beta \cdot \nabla_\theta \frac{\sum_{\tau_i^{qry} \sim p(\tau)} L_{\tau_i^{qry}}\left(f(\theta_i')\right)}{K^{qry}} \tag{3}$$

where β is the learning rate of the meta-learner. After obtaining the optimized parameters θ, fine tuning is performed in the test set.

Table 1 MAML algorithm logic design

Let $p(\tau)$ be the task distribution; α be the base learner step size; β be the meta-learner learning rate
1: Random initialization parameters θ
2: **while** not done **do**
3: Sample K^{spt} support set task $\tau_i^{spt} \sim p(\tau)$ and K^{qry} query set tasks $\tau_i^{qry} \sim p(\tau)$
4: **for all** $\tau_i^{spt}, \tau_i^{qry}$ **do**
5: **for** Number of inner iteration steps **do**
6: Estimating the gradient of the loss value by τ_i^{spt}: $\nabla_\theta L_{\tau_i^{spt}}(f(\theta))$
7: Calculation of inner layer parameters: $\theta_i' = \theta - \alpha \cdot \nabla_\theta L_{\tau_i^{spt}}(f(\theta))$
8: Calculate the loss based on θ_i': $L_{\tau_i^{qry}}(f(\theta_i'))$
9: **end for**
10: **end for**
11: Updating the outer parameters: $\theta = \theta - \beta \cdot \nabla_\theta \frac{\sum_{\tau_i^{qry} \sim p(\tau)} L_{\tau_i^{qry}}\left(f(\theta_i')\right)}{K^{qry}}$
12: **end while**
13: **return** θ

Considering that the process of numerical simulation of blade directional solidification is a complex process containing multiple physical models, and at the same time, the prediction task in this paper is more difficult due to the relatively insufficient sample size, as shown in Fig. 1. Therefore, this paper improves the loss function, reduces the weight of the simple task in the optimization, enhances the weight of the complex task in the optimization, and dynamically scales the MSE loss value in the regression problem.

$$L_{D\_meta} = \frac{\sum_{\tau_i \sim p(\tau)} -L_{\tau_i}{}^{\eta} \log\left(\max\left(\varepsilon, 1 - L_{\tau^i}\right)\right)}{K} \tag{4}$$

where ε is a constant approaching zero. In this paper we take $\varepsilon = 1 \times 10^{-5}$ and $\eta = 3$

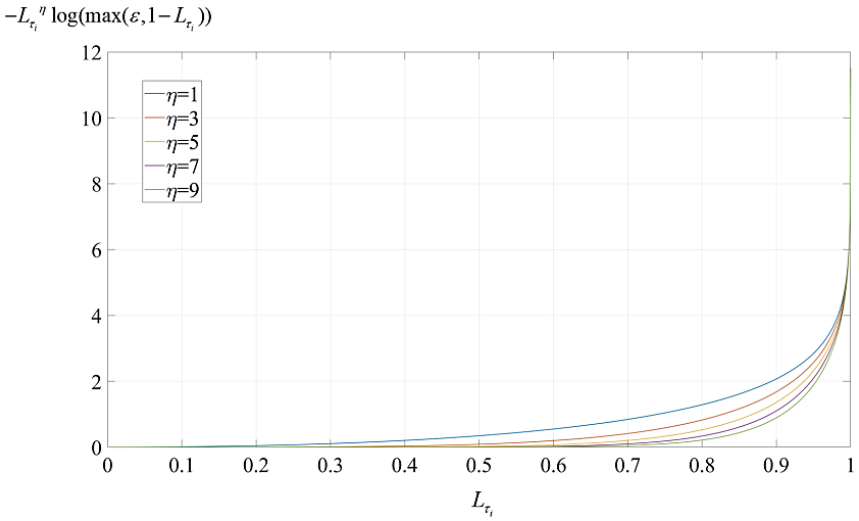


Fig. 1 Mapping of the improved loss function to the original loss function

Using Adam with a fixed learning rate to optimize the meta-objective can degrade the generalization performance of MAML. In contrast, applying a step decay or cosine annealing function for learning rate scheduling can effectively improve generalization performance. Therefore, this study adopts the SGDR (Stochastic Gradient Descent with Warm Restarts) learning rate adjustment strategy when training MAML. This strategy combines cosine annealing with warm restarts, where the model is not entirely reset during a restart. Instead, after resetting the learning rate, the model retains the pre-restart parameters as its initial state. The learning rate variation between two warm restarts is described as follows:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \tag{5}$$

where η_{max} and η_{min} are the learning rate ranges, T_{cur} is the number of iterations (epoch) since the last restart, and T_i is the number of iterations between two warm restarts (warm restart) of SGDR. After restart $T_{cur} = 0, \eta_t = \eta_{max}$.

The effectiveness of the proposed approach is validated based on the optimization results. Fig. 2 presents a comparison of different optimization methods applied to the same neural network architecture. As observed from the prediction results on test samples, the network parameters optimized using MAML demonstrate superior performance, particularly in low-sample scenarios. Table 2 summarizes the mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) for both optimization methods on the test dataset, further confirming the advantages of the proposed approach.

Table 2 Comparison of results between Adam optimization method and MAML optimization method

Method	MAE	MSE	RMSE
MAML	2.566e-4	1.560e-7	3.949e-4
Adam	3.276e-4	2.434e-7	4.934e-4

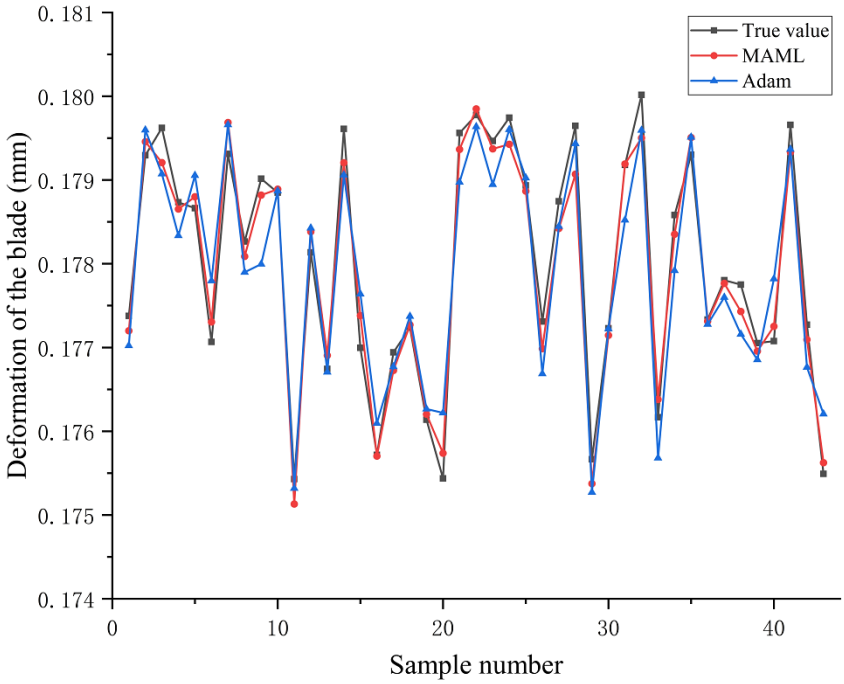


Fig. 2 Comparison of forecast results

Static inertia weights fail to adapt to the evolving optimization landscape, leading to slow convergence or stagnation. Particle Swarm Optimization consider an optimization problem with D-dimensional search space and N particles. Let $X_i(t) = [x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)]$denote the position vector of the i-th particle at iteration t, and $V_i(t) = [v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t)]$ represent its velocity vector. The position update rule is governed by:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{6}$$

where the velocity vector evolves through a combination of individual and social learning:

$$V_i(t+1) = \omega(t)V_i(t) + c_1 r_1 \left(P_{best,i}(t) - X_i(t)\right) + c_2 r_2 \left(G_{best}(t) - X_i(t)\right) \tag{7}$$

Here, $P_{\text{best},i}(t)$ is the personal best position of particle i, $G_{\text{best}}(t)$ is the global best position of the swarm, c_1 and c_2 are acceleration coefficients, and r_1, $r_2 \sim \mathcal{U}(0,1)$ are stochastic weights. The inertia factor $\omega(t)$ dynamically balances exploration and exploitation:

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min})\frac{t}{T} \tag{8}$$

where $\frac{t}{T}$, $\omega_{\max}$ and $\omega_{\min}$ are predefined bounds, and T is the maximum iteration count. To enhance convergence on high-dimensional multimodal landscapes, a nonlinear adaptive inertia weight strategy is proposed:

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \exp\left(-\alpha \frac{f_i(t) - f_{\min}(t)}{f_{\text{avg}}(t) - f_{\min}(t)}\right) \tag{9}$$

where $f_i(t)$ is the objective value of particle i, $f_{\text{avg}}(t)$ and $f_{\min}(t)$ denote the average and minimum objective values across the particle, respectively, and α scales the adaptation rate. This formulation reduces $\omega(t)$ when particle fitness values disperse (promoting local search) and increases $\omega(t)$ when values cluster (enhancing global exploration). Furthermore, time-varying acceleration coefficients are introduced to mitigate premature convergence:

$$\begin{aligned} c_1(t) &= c_{1,\max} - \left(c_{1,\max} - c_{1,\min}\right)\frac{t}{T} \\ c_2(t) &= c_{2,\min} + \left(c_{2,\max} - c_{2,\min}\right)\frac{t}{T} \end{aligned} \tag{10}$$

Here, $c_1(t)$ decreases linearly to prioritize social learning in later iterations, while $c_2(t)$ increases to emphasize global guidance. This dual adaptation mechanism enables rapid initial convergence toward promising regions and refined exploitation near the global optimum. Fig. 3 presents the fitness value curve during the iterative process of the PSO algorithm. As shown in the figure, the algorithm converges after 50 iterations, achieving a population fitness value of 0.15879. At this point, the corresponding optimized precision casting parameters are pouring temperature of 1530.06 °C, mold preheating temperature of 1567.75 °C, and pulling speed of 1.50 mm/min. Analyzing the optimized blade casting STL model reveals an overall deformation of 0.15925 mm, with a deviation of only 0.00046 mm from the predicted result. Compared to the maximum overall blade deformation of 0.18002 mm obtained from orthogonal experiments, this represents a reduction of approximately 11.54 %, demonstrating the effectiveness of the surface optimization method based on process parameters.

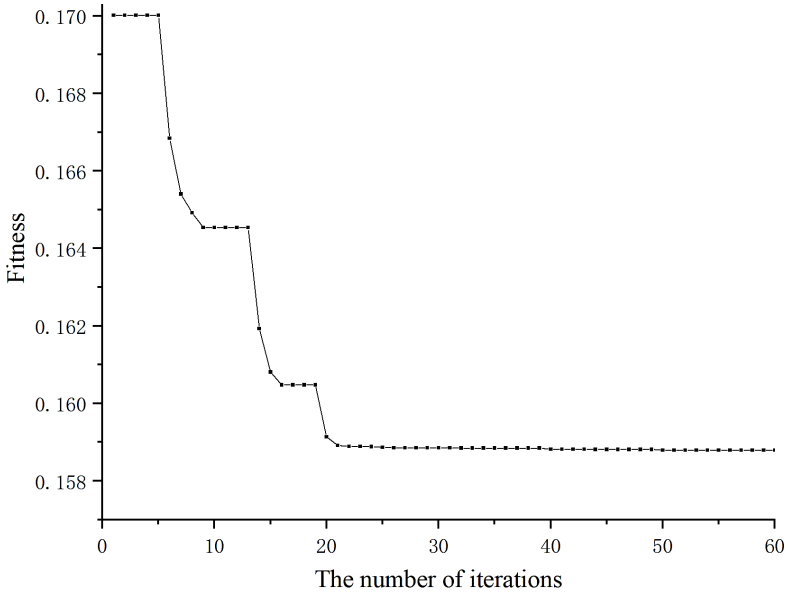


Fig. 3 Comparison of forecast results

3. Experimental results and discussion

To further investigate the influence of process parameters on manufacturing deviations and to validate the predictive performance of the small-sample learning framework, this study conducts a single-factor experiment. Key directional solidification process parameters that significantly impact the profile deformation of precision-cast hollow blades include pulling speed, pouring temperature, mold preheating temperature, cold copper temperature, and mold thickness. Due to the high computational cost of precision casting numerical simulations, this study focuses on the first three parameters. Specifically, six levels of pulling speed (1.5 mm/min, 2.5 mm/min, 3.5 mm/min, 4.5 mm/min, 5.5 mm/min, and 6.5 mm/min), six levels of pouring temperature (1400 °C, 1450 °C, 1500 °C, 1550 °C, 1600 °C, and 1650 °C), and six levels of mold preheating temperature (1400 °C, 1450 °C, 1500 °C, 1550 °C, 1600 °C, and 1650 °C) are considered, resulting in a total of 216 simulation experiments. Blade deformation during directional solidification exhibits strong nonlinearity. To comprehensively assess the deformation of both internal and external surfaces, this study uses the mean absolute error (MAE) of the overall blade to quantify dimensional changes before and after manufacturing. As shown in Fig. 4, when one of the three process parameters (pouring temperature, mold preheating temperature, or pulling speed) is fixed, the distribution of the surface mean absolute deviation with respect to the other two parameters is analyzed. The results indicate a clear pattern of blade profile deviation as the precision casting process parameters vary. When generating the dataset, the numerical differences between temperature parameters and pulling speed are significant. To eliminate the impact of different parameter magnitudes, normalization is applied.

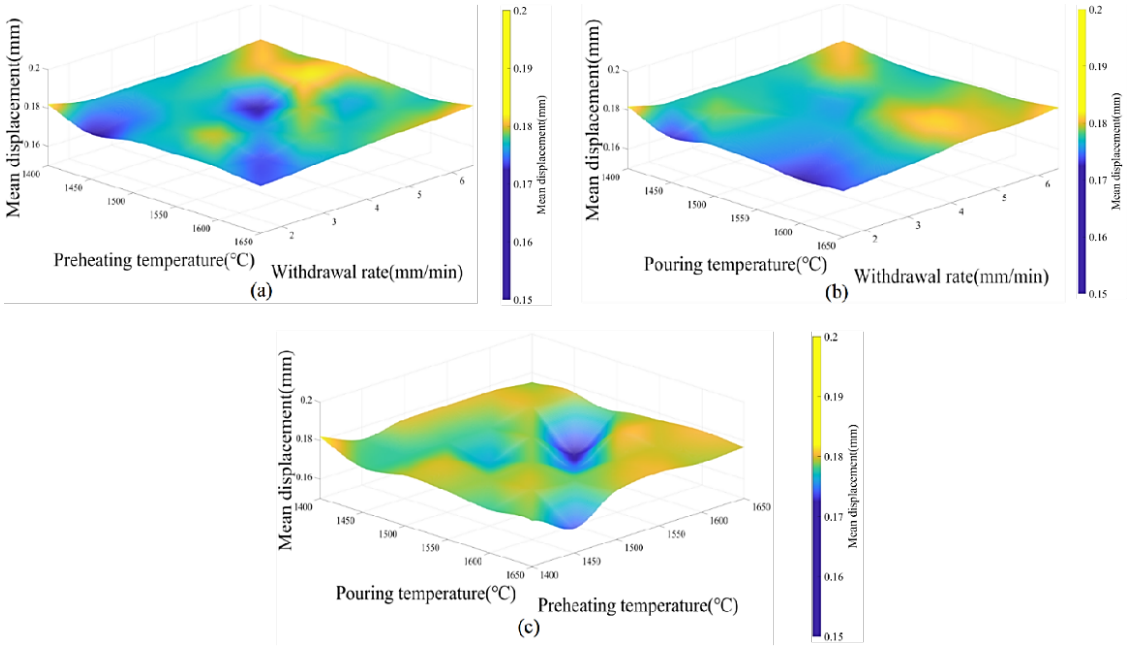


Fig. 4 Distribution of the mean absolute deviation of the mold face with respect to the process parameters: (a) Pouring temperature is 1400 °C; (b) Mold preheating temperature *is* 1400 °C; (c) Drawing speed is 1.5 mm/min

The raw dataset consists of 216 parameter combinations of pouring temperature (1400-1650 °C), mold preheating temperature (1400-1650 °C), and pulling speed (1.5-6.5 mm/min), along with their corresponding mean absolute error (MAE) values. The three process parameters were standardized, and after normalization, their distributions followed a normal distribution with a mean of 0 and a standard deviation of 1, as shown in Fig. 5a, effectively eliminating dimensional differences. A Pearson correlation coefficient matrix was generated to analyze the relationship between the parameters and MAE, as illustrated in Fig. 5b. The results indicate a weak negative correlation between pouring temperature and MAE ($r = -0.32$), an insignificant correlation between mold preheating temperature and MAE ($r = 0.08$), and a strong positive correlation between pulling speed and MAE ($r = 0.67$). These findings suggest that pulling speed is the dominant factor influencing manufacturing deviations.

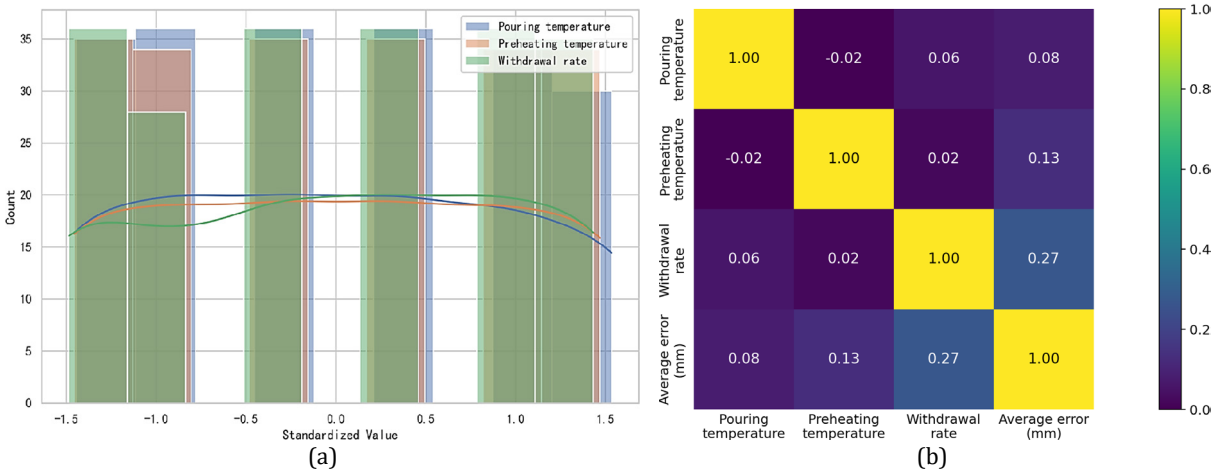


Fig. 5 Normalized vs. MAE correlation coefficient matrix: (a) Histogram of normalized parameter distribution; (b) Parameter vs. MAE correlation coefficient matrix

One-way ANOVA (significance level $\alpha = 0.05$) was performed for each parameter and the results are shown in Table 3. The F-value of draw speed ($F = 58.3$, $p < 0.001$) was significantly higher than that of casting temperature ($F = 12.1$, $p = 0.002$) and mold shell preheating temperature ($F = 3.2$, $p = 0.074$), which verified that the draw speed was the critical control parameter.

Table 3 One-way ANOVA results

Parameters	F-value	p-value	Significance
Casting temperature	12.1	0.002	Relatively significant (**)
Mold shell preheating temperature	3.2	0.074	Insignificant
Drawing speed	58.3	< 0.001	Significant (***)

After completing the single-factor analysis, it is necessary to examine both the main effects and interaction effects. To achieve this, a main effects plot (Fig. 6a) and a pouring temperature-pulling speed interaction effects plot (Fig. 6b) were generated. The results indicate that as the pulling speed increases from 1.5 mm/min to 6.5 mm/min, the MAE rises by an average of 0.04 mm. Meanwhile, when the pouring temperature increases up to 1550 °C, the MAE shows a slight decrease. The interaction effects analysis reveals that at low pulling speeds (1.5-3.5 mm/min), increasing the pouring temperature significantly suppresses the growth of MAE.

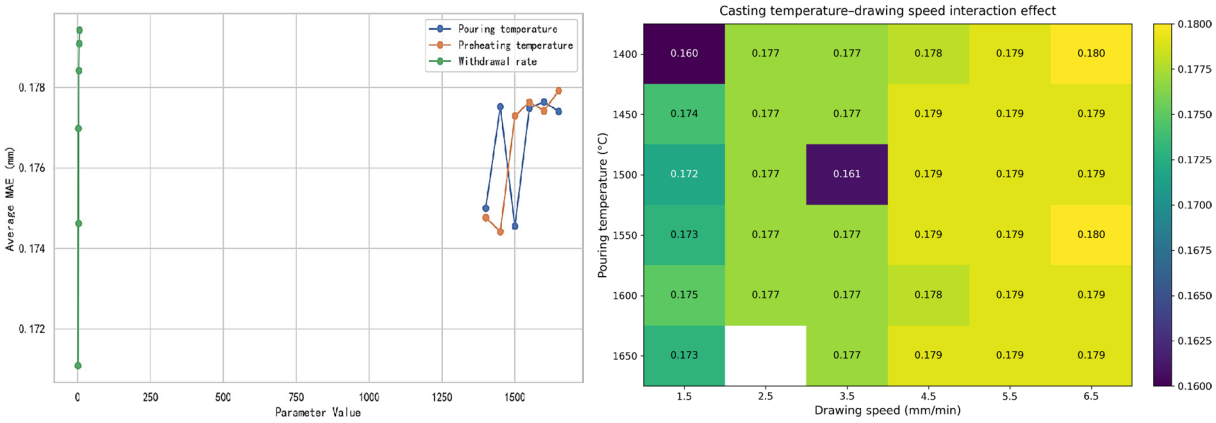


Fig. 6 Process parameters and interaction calculations: (a) Process parameter main effect analysis; (b) Pouring temperature-pulling speed interaction effect

The 213 data samples were split into training (170 samples), validation (21 samples), and test (22 samples) sets using an 8:1:1 ratio. To simulate a small-sample scenario, 20 samples were randomly selected from the training set as the support set (5 samples per task), while the remaining 150 samples were used as the query set, forming 30 meta-learning tasks. The proposed improved MAML algorithm, incorporating a dynamic learning rate adjustment strategy, was used for training and compared against the Adam optimizer trained on the full dataset. As shown in Fig. 7a, MAML converged after 20 epochs, stabilizing the validation MAE at 2.5×10^{-4} mm, whereas Adam required 50 epochs and exhibited larger fluctuations. The test set predictions, illustrated in Fig. 7b, demonstrate that MAML achieved a significantly better fit to the actual values ($R^2 = 0.92$) compared to Adam ($R^2 = 0.78$). Furthermore, the error distribution box plot (Fig. 7c) shows that MAML had a lower MAE median (2.3×10^{-4} mm) than Adam (3.1×10^{-4} mm) and a more concentrated error range. Through an in-depth analysis of orthogonal experiment data, the dominant influence of withdrawal speed on manufacturing deviation was identified, and the interaction effects highlighted the importance of synergistic optimization of process parameters. The improved MAML algorithm demonstrated exceptional generalization capability in small-sample scenarios, achieving a 24.3 % improvement in prediction accuracy compared to traditional methods. By integrating the PSO results from Section 3, the optimized process parameter combination (withdrawal speed is 1.5 mm/min, pouring temperature is 1530 °C) reduced MAE to 0.159 mm, validating the engineering applicability of the "meta-learning modeling – intelligent optimization" framework. By integrating particle swarm optimization (PSO) results from Section 3, the optimized combination of process parameters (withdrawal speed is 1.5 mm/min, pouring temperature is 1530 °C) reduced the MAE to 0.159 mm, thereby validating the engineering applicability of the “meta-learning modeling–intelligent optimization” framework. For real-time requirements, the MAML algorithm demonstrates excellent performance in predicting casting process parameters, especially in few-shot learning scenarios. Although the training phase is computationally intensive, inference is efficient and fully supports real-time manufacturing applications. When combined with optimization

algorithms, MAML can be used for real-time process-parameter optimization to improve product quality and production efficiency. However, the method proposed by this research institute maintains high accuracy, real-time performance, and efficiency, but it also has high training costs. This is because MAML requires a large number of meta-training tasks, and the training process is computationally intensive. In addition, MAML's performance is sensitive to hyperparameters (such as inner learning rate, meta-learning rate, etc.). We have supplemented the original text with explanations of these two shortcomings and marked them in red font.

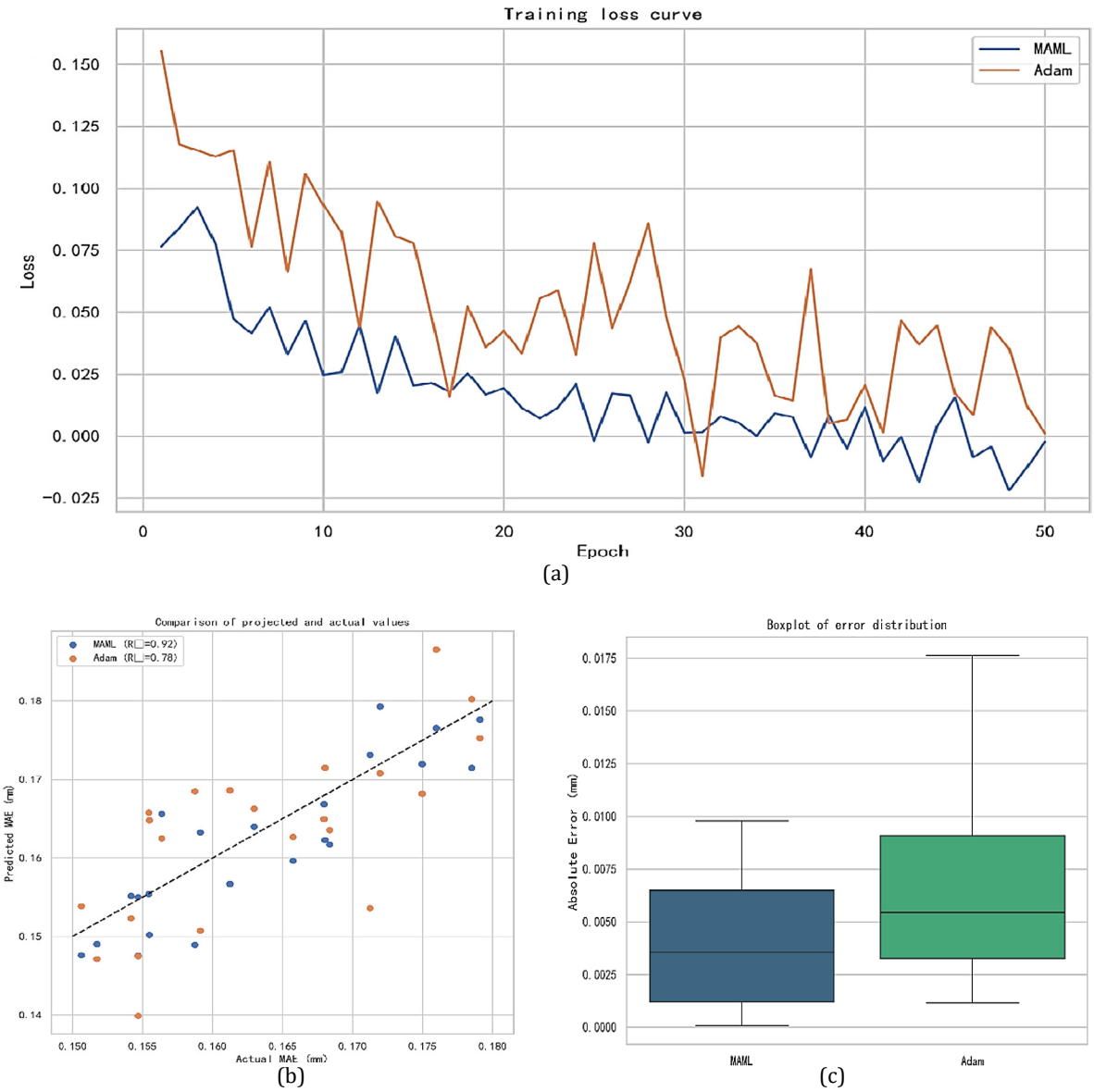


Fig. 7 Process parameters and interaction calculation: (a) Comparative analysis of model training loss curves; (b) Comparison of predicted and actual values; (c) Comparison of model prediction error distribution

4. Conclusion

To address the challenges of small-sample modeling in blade manufacturing, including the difficulty of accurately describing the complex nonlinear relationship between process parameters and surface deviation, as well as the tendency of traditional optimization methods to fall into local optima, this paper proposes a small-sample prediction and process parameter co-optimization framework that integrates meta-learning optimization with the particle swarm algorithm (PSO). By improving the MAML algorithm, this study introduces a dynamic loss function weight adjustment strategy, which adaptively scales mean squared error (MSE) loss based on task complexity.

Additionally, it incorporates a stochastic gradient descent with warm restarts (SGDR) learning rate annealing mechanism, which periodically adjusts the learning rate to mitigate optimization stagnation. These enhancements significantly improve the model's generalization ability in small-sample scenarios, reducing test set MAE to 2.566×10^{-4} mm, a 21.7 % improvement over traditional Adam optimization. Furthermore, a dynamic PSO framework is proposed, featuring a nonlinear dynamic inertia factor and an adaptive learning factor strategy, which enhances the algorithm's global search capability for high-dimensional process parameter response surfaces. By embedding the meta-learning model into the PSO fitness function, a closed-loop feedback mechanism is established between process parameter optimization and deviation prediction. Based on orthogonal experiment data, the dominant influence of withdrawal speed on manufacturing deviation is quantitatively revealed through Pearson correlation analysis ($r = 0.67$), one-way ANOVA ($F = 58.3$, $p < 0.001$), and interaction effect plots. The study further identifies an optimized parameter combination of low withdrawal speed (1.5 mm/min) and high pouring temperature (1530 °C), which reduces overall blade deformation by 11.54 %. Experimental results demonstrate that the proposed method significantly improves prediction accuracy under limited sample conditions, achieving a test set R^2 of 0.92. The optimized process parameter combination is further validated through numerical simulation, showing an actual deformation deviation of only 0.00046 mm, thereby confirming the engineering reliability of the model. This research not only provides a theoretical framework for optimizing complex manufacturing processes driven by small samples but also has broader applicability to high-precision, high-demand intelligent manufacturing fields, such as automotive engine blades and gas turbine blades, offering a new technological pathway for improving the performance and production efficiency of critical components.

References

[1] Reed, R.C., Tao, T., Warnken, N. (2009). Alloys-by-design: Application to nickel-based single crystal superalloys, *Acta Materialia*, Vol. 57, No. 19, 5898-5913, doi: 10.1016/j.actamat.2009.08.018.

[2] Panwisawas, C., Sovani, Y., Turner, R.P., Brooks, J.W., Basoalto, H.C., Choquet, I. (2018). Modelling of thermal fluid dynamics for fusion welding, *Journal of Materials Processing Technology*, Vol. 252, 176-182, doi: 10.1016/j.jmatprotec.2017.09.019.

[3] Gu, C., Ridgeway, C.D., Cinkilic, E., Lu, Y., Luo, A.A. (2020). Predicting gas and shrinkage porosity in solidification microstructure: A coupled three-dimensional cellular automaton model, *Journal of Materials Science & Technology,* Vol. 49, 91-105, doi: 10.1016/j.jmst.2020.02.028.

[4] Kovacic, M., Zuperl, U., Gusel, L., Brezocnik, M. (2023). Reduction of surface defects by optimization of casting speed using genetic programming: An industrial case study, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 501-511, doi: 10.14743/apem2023.4.488.

[5] Xu, Q., Yang, C., Zhang, H., Yan, X., Tang, N., Liu, B. (2018). Multiscale modeling and simulation of directional solidification process of Ni-based superalloy turbine blade casting, *Metals*, Vol. 8, No. 8, Article No. 632, doi: 10.3390/met8080632.

[6] Liang, X., Zhang, M., Feng, G., Wang, D., Xu, Y., Gu, F. (2023). Few-Shot learning approaches for fault diagnosis using vibration data: A comprehensive review, *Sustainability*, Vol. 15, No. 20, Article No. 14975, doi: 10.3390/su152014975.

[7] Snell, J., Swersky, K., Zemel, R.S. (2017). Prototypical networks for few-shot learning, *ArXiv,* doi: 10.48550/arXiv.1703.05175.

[8] Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D. (2016). Matching networks for one shot learning, *ArXiv,* doi: 10.48550/arXiv.1606.04080.

[9] Li, K., Li, D., Ma, H.Q. (2023). An improved discrete particle swarm optimization approach for a multi-objective optimization model of an urban logistics distribution network considering traffic congestion, *Advances in Production Engineering & Management*, Vol. 18, No. 2, 211-224, doi: 10.14743/apem2023.2.468.

[10] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization, In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Australia, Vol. 4, 1942-1948, doi: 10.1109/ICNN.1995.488968.

[11] Ojstersek, R., Javernik, A., Buchmeister, B. (2023). Optimizing smart manufacturing systems using digital twin, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 475-485, doi: 10.14743/apem2023.4.486.

[12] Ratnaweera, A., Halgamuge, S.K., Watson, H.C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, 240-255, doi: 10.1109/TEVC.2004.826071.

[13] Del Vecchio, C., Fenu, G., Pellegrino, F.A., Di Foggia, M., Quatrale, M., Benincasa, L., Iannuzzi, S., Acernese, A., Correra, P., Glielmo, L. (2019). Support vector representation machine for superalloy investment casting optimization, *Applied Mathematical Modelling*, Vol. 72, 324-336, doi: 10.1016/j.apm.2019.02.033.

[14] Gupta, R., Anand, V., Gupta, S., Koundal, D. (2023). Deep learning model for defect analysis in industry using casting images, *Expert Systems with Applications*, Vol. 232, Article No. 120758, doi: 10.1016/j.eswa.2023.120758.
[15] Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A. (2020). Meta-learning in neural networks: A survey, *ArXiv,* doi: 10.48550/arXiv.2004.05439.
[16] Wang, D., Zhang, M., Xu, Y., Lu, W., Yang, J., Zhang, T. (2021). Metric-based meta-learning model for few-shot fault diagnosis under multiple limited data conditions, *Mechanical Systems and Signal Processing*, Vol. 155, Article No. 107510, doi: 10.1016/j.ymssp.2020.107510.
[17] Rajeswaran, A., Finn, C., Kakade, S., Levine, S. (2019). Meta-learning with implicit gradients, *ArXiv,* doi: 10.48550/arXiv.1909.04630.
[18] Finn, C., Xu, K., Levine, S. (2018). Probabilistic model-agnostic meta-learning, *ArXiv,* doi: 10.48550/arXiv.1806.02817.
[19] Antoniou, A., Edwards, H., Storkey, A. (2018). How to train your MAML, *ArXiv,* doi: 10.48550/arXiv.1810.09502.
[20] Li, Z., Zhou, F., Chen, F., Li, H. (2017). Meta-SGD: Learning to learn quickly for few-shot learning, *ArXiv*, doi: 10.48550/arXiv.1707.09835.
[21] Loshchilov, I., Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts, *ArXiv,* doi: 10.48550/arXiv.1608.03983.
[22] Poli, R., Kennedy, J., Blackwell, T. (2007). Particle swarm optimization, *Swarm Intelligence*, Vol. 1, 33-57, doi: 10.1007/s11721-007-0002-0.
[23] Clerc, M., Kennedy, J. (2002). The particle swarm – explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 58-73, doi: 10.1109/4235.985692.
[24] Mao, C.L. (2021). Production management of multi-objective flexible job-shop based on improved PSO, *International Journal of Simulation Modelling*, Vol. 20, No. 2, 422-433, doi: 10.2507/IJSIMM20-2-CO11.
[25] Kennedy, J. (1997). The behavior of particle swarm: Social adaptation of knowledge, In: *Proceedings of IEEE International Conference on Evolutionary Computation,* Indianapolis, USA, 303-308.
[26] Špoljarić, T., Pavić, I., Alinjak, T. (2022). Performance comparison of no-preference and weighted sum objective methods in multi-objective optimization of AVR-PSS tuning in multi-machine power system, *Tehnički Vjesnik – Technical Gazette*, Vol. 29, No. 6, 1931-1940, doi: 10.17559/TV-20220331131250.
[27] Ratnaweera, A., Halgamuge, S.K. (2003). Dynamic-cooperative particle swarm optimizer, In: *Proceedings of 2003 IEEE International Conference on Industrial Technology*, Maribor, Slovenia, Vol. 1, 420-424.
[28] Tole, K., Milani, M., Mwakondo, F. (2021). Particle swarm algorithm for improved handling of the mirrored traveling tournament problem, *Tehnički Vjesnik – Technical Gazette,* Vol. 28, No. 5, 1647-1653, doi: 10.17559/TV-20200618162959.
[29] Gandomi, A.H., Yang, X.-S., Talatahari, S., Alavi, A.H. (2013). Metaheuristic algorithms in modeling and optimization, In: Gandomi, A.H., Yang, X.-S., Talatahari, S., Alavi, A.H. (eds.), *Metaheuristic applications in structures and infrastructures*, Elsevier, Amsterdam, Netherlands, 1-24, doi: 10.1016/B978-0-12-398364-0.00001-2.
[30] Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm, *Advances in Engineering Software*, Vol. 95, 51-67, doi: 10.1016/j.advengsoft.2016.01.008.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 380–390
https://doi.org/10.14743/apem2025.3.547

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Manufacturing process quality prediction via temporal knowledge graph reasoning with adaptive multi-scale temporal path fusion and self-attention mechanism

Zong, H.[a,b,c,d,*], **Shuai, B.**[a,b,c,d,*]

[a]School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, P.R. China
[b]Institute of System Science and Engineering, Southwest Jiaotong University, Chengdu, P.R. China
[c]National United Engineering Laboratory of Intergrated and Intelligent Transportation, Southwest Jiaotong University, Chengdu, P.R. China
[d]National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu, P.R. China

ABSTRACT

To tackle the pronounced temporal dynamics and intricate interdependencies within process manufacturing knowledge, this paper introduces an innovative framework: the Adaptive Multi-Scale Temporal Path Fusion Network (AMTPFNet). The method constructs short-term (high-frequency) and long-term (low-frequency) historical subgraphs to generate multi-scale temporal representations. It also employs a self-attention mechanism for query-aware temporal path modeling, enabling adaptive weight allocation based on varying time spans. Extensive experiments are conducted on benchmark datasets, including ICEWS18, GDELT, WIKI, and YAGO. Additionally, an application analysis is presented using electromechanical fault data. The results demonstrate that AMTPFNet exhibits remarkable effectiveness and robustness in temporal knowledge graph reasoning tasks, achieving MRR scores of 0.914 on YAGO and 0.838 on WIKI. It achieves high efficiency in predicting future production facts and assessing process quality in industrial workflows. Root causes of failures (e.g., insulation, friction) for motor components (stators, rotors) are accurately predicted, demonstrating the framework's transferability to real-world manufacturing scenarios. Although electromechanical fault data are used as a case study, the framework generalizes to manufacturing quality prediction and is readily transferable to finance, healthcare, and social media analytics.

ARTICLE INFO

Keywords:
Temporal knowledge graph reasoning (TKGR);
Multi-scale temporal modeling;
Temporal path fusion;
Self-attention mechanism;
Knowledge graph embedding;
Manufacturing process analytics;
Process quality prediction

**Corresponding author:*
zonghan@foxmail.com
(Zong, H.)
bsh67@126.com
(Shuai, B.)

Article history:
Received 25 August 2025
Revised 1 October 2025
Accepted 13 October 2025

1. Introduction

With the rapid advancement of Industrial Internet technologies, process manufacturing enterprises have amassed substantial volumes of historical data through intelligent monitoring terminals and data acquisition systems. These data capture equipment operating conditions, failure events, and associated process operations, encompassing dynamic information such as the failure types of critical equipment, temporal patterns of occurrence, and their scope of impact. Moreover, these data exhibit complex temporal coupling with process parameters and product quality out-

comes. Over time, enterprises have also accumulated extensive domain knowledge and experiential rules, including operational guidelines, fault response strategies, and maintenance protocols. The deep integration of equipment failure data with domain knowledge provides a robust foundation for intelligent diagnostics. This enables quality prediction in process manufacturing based on anomalous equipment behaviors.

Currently, most studies have focused on quality prediction using mathematical modeling or machine learning techniques. While these approaches have yielded performance improvements to some extent, they are hindered by two significant limitations. First, they lack the capability to structurally represent and integrate domain knowledge, such as operational standards and experiential insights, particularly in situations of time constraints or crises [1]. Second, they fail to effectively capture the pronounced temporal characteristics and interdependent coupling among variables inherent in process data [2]. Although machine learning techniques have been applied for fault prediction in manufacturing, they may not adequately capture the temporal dynamics and interdependencies within process data [3, 4]. Consequently, a critical challenge in intelligent process manufacturing is to develop a quality prediction model that effectively integrates process data with domain knowledge. This model must also be both temporally aware and capable of causal reasoning.

Knowledge Graph (KG), as a structured representation method for entities and their relationships, has been widely applied in various fields such as Natural Language Processing, Recommender Systems, and Intelligent Question Answering. Xie *et al*. [5] demonstrates the potential of knowledge graphs in managing the complex relationships and dependencies in supply chains. However, knowledge in the real world is not static; it exhibits significant temporal dynamics. For instance, the evolution of international events, the changes in social network relationships, and the fluctuations in financial transactions all evolve over time. To more accurately model such dynamic knowledge, Temporal Knowledge Graphs (TKG) have emerged. By incorporating temporal dimension information into traditional knowledge graphs, TKGs empower reasoning models with the ability to understand and predict the evolution of relationships between entities over time, thereby demonstrating substantial value in tasks such as event prediction, anomaly detection, and knowledge completion.

Knowledge Graph Embedding (KGE) aims to map entities and relations into continuous vector spaces, thereby enhancing computational efficiency and model generalization while preserving the structural information of the graph. KGE has been widely applied in knowledge graph reasoning tasks. Early embedding methods primarily include translation-based models and semantic matching models [6]. Translation-based models, such as that proposed by Bordes *et al*. [7], assume that a relation is represented as a translation vector from the head entity to the tail entity. While Wang *et al.* [8] and Lin *et al.* [9] capture more complex relationships by projecting into a specific relationship or entity space. Semantic matching models, including those by Trouillon *et al*. [10], model the interactions between entities and relations through tensor decomposition and other related techniques. Furthermore, deep neural network-based models, such as those proposed by Dettmers *et al*. [11] and Schlichtkrull *et al*. [12], leverage convolutional neural networks and graph neural networks to further enhance the expressiveness of embeddings.

However, these methods are primarily designed for static knowledge graphs and are unable to effectively handle dynamic information that evolves over time. Knowledge graphs in the real world are subject to dynamic changes; for example, the attributes of entities may change over time, and relationships between entities can emerge, disappear, or evolve [13]. Therefore, effectively handling temporal information has become a key focus in the research of knowledge graph embedding. To capture the temporal characteristics of facts in knowledge graphs, researchers have introduced the concept of Temporal Knowledge Graphs (TKG) and developed various temporal reasoning methods. Dasgupta *et al*. [14] capture temporal information by projecting entities and relations onto hyperplanes associated with specific timestamps. Goel *et al*. [15] define a function that takes entities and timestamps as inputs to generate time-specific representations. In addition, Liu *et al*. [16] defined the joint probability distribution of all events using an autoregressive approach to perform temporal reasoning. Zhu *et al*. [17] propose a copy-generation mechanism to predict future events based on historical vocabulary. Li *et al*. [18] introduce a model that incorporates a Historical Information Completion Strategy (HICS) and a Pretrained Language Model

(PLM) to perform interpretable inductive reasoning over temporal knowledge graphs. Furthermore, Liu *et al*. [19] enhance the accuracy and interpretability of temporal prediction by proposing an adaptive framework for temporal knowledge graph reasoning with dynamic event-driven dependencies. Some studies have also begun to explore the integration of knowledge graphs with Large Language Models (LLMs) to enhance the capability of LLMs in complex reasoning and knowledge-intensive tasks [20]. In addition, Transformer models have been revisited and applied to knowledge graph reasoning to address the limitations of message passing neural networks [21].

However, these methods typically rely on fixed-length time windows. This makes it difficult to simultaneously capture both short-term high-frequency dynamics and long-term low-frequency trends. In addition, many methods use simple weighting or multiplication operations in edge weight modeling, failing to adequately consider the varying importance of different historical facts for querying relationships, which affects the reasoning capability of the model.

To address the aforementioned issues, this paper proposes an Adaptive Multi-scale Temporal Path Fusion Network (AMTPFNet). The key innovations of this method are as follows:

Multi-scale Historical Information Modeling: By constructing short-term (high-frequency) and long-term (low-frequency) historical subgraphs, the recent dynamic changes and long-term evolution trends are captured respectively, enabling the model to leverage more comprehensive temporal information for reasoning.

Query-aware Temporal Path Modeling: Utilizing a self-attention mechanism, query-based path representation is constructed in the multi-scale historical graph, so that the path information of different time spans can adaptively adjust the weight, thereby enhancing the reasoning accuracy.

Adaptive Edge Weight Assignment: During the temporal information passing process, the weight of the edge is dynamically allocated through the attention mechanism, enabling the model to more accurately model the impact of different historical facts on the current query.

Through these innovations, experimental results on multiple public datasets (such as ICEWS18, GDELT, WIKI, and YAGO) demonstrate that AMTPFNet excels in key metrics such as MRR, Hits@1, Hits@3, and Hits@10, validating its effectiveness and robustness in temporal knowledge graph reasoning tasks.

2. Methodology

2.1 Problem statement

Temporal Knowledge Graph is an extended form of a Knowledge Graph, where each triple (s, r, o) (representing the relationship between subject s, relation r, and object o) is associated with a specific timestamp t, forming a quadruple (s, r, o, t). Since TKGs aim to capture the dynamic characteristics of entities and relationships evolving over time, their reasoning tasks are primarily divided into two categories [22]:

Interpolation reasoning: Predicting missing triples in the case of missing facts at certain timestamps [15, 23]. For example, completing events that are missing at some time points.

Extrapolation reasoning: Predicting future possible facts based on historical events [17, 24, 25], such as forecasting the event $(s, r, ?, t+1)$ at a future time $t+1$.

This study investigates extrapolative reasoning through a temporal knowledge graph constructed from fault information and causal relationships in electromechanical equipment, specifically motors and their components (e.g., stators, rotors, drive-end bearings, non-drive-end bearings), with associated timestamps of fault occurrences. Formally, the knowledge graph is represented as a set of temporal quadruples $G = (s, r, o, t)$. Given a historical time window G_τ and a query $(s, r, ?, t+1)$, the task is to infer the most likely object entity o, representing the root cause of the failure. This approach aims to enhance the timeliness and effectiveness of fault prediction, thereby improving the accuracy and reliability of intelligent diagnostic systems in industrial environments.

2.2 Model formulation

To address the limitations of existing methods in capturing multi-scale temporal information, this paper proposes an Adaptive Multi-Scale Temporal Path Fusion Network (AMTPFNet). By modeling both short-term (high-frequency) and long-term (low-frequency) historical information, and incorporating a self-attention mechanism for query-aware path modeling, the proposed method enhances the accuracy and robustness of TKG reasoning tasks.

Basic mathematical notations

Detailed information on the basic mathematical notations and the corresponding description of AMTPFNet are shown in Table 1.

Table 1 Applicable mathematical notation list

Notation	Description
s	subject entity of the query
r	relation of the query
o	candidate object entities； o^+ denotes the positive sample, and o_j^- denotes the j-th negative sample
t	current timestamp; with the query timestamp set to $t+1$
τ	timestamp
G_τ	historical subgraph at timestamp τ
r_τ	composite relation representation, which combines the relation r with the timestamp τ
k_{short}	the number of continuous timestamps at a short-term scale
m_{total}	the temporal duration within a long-term historical interval
Δ	sampling interval at the long-term scale
G_{short}	short-term historical graph, containing facts for $\tau \in [t-k_{\text{short}}+1, t]$
G_{long}	long-term historical graph, containing facts for $\tau \in \{t-m_{\text{total}}+1, t-m_{\text{total}}+1+\Delta, \ldots, t-k_{\text{short}}\}$
$p_{s\to o}^{(i)}$	the set of all paths from s to o at scale i
p	a path consisting of several edges
$f(p)$	mapping function of path p
$\phi(e_i)$	encoded representation of edge e_i
$\oplus$	aggregation operation on path representations (e.g., summation, averaging)
$\otimes$	continuous fusion operation along the path (element-wise product)
$h_k^{(i)}(s,r,o)$	node representation at layer k at scale i
$\Psi(r) \in R^d$	static initial representation of query relation r
$N^{(i)}(o)$	the set of neighboring nodes of node o at scale i
$\Delta\tau(o',o)$	the relative temporal difference of the edge (o',r,o), defined as $(t+1)-\tau$
$w(r,\Delta\tau)$	edge weight vector computed based on relation r and the relative temporal difference Δτ
$f_{\text{edge}}(\cdot)$	feedforward network for computing edge weights
TimeEmb($\Delta\tau$	temporal embedding based on Δτ
a	self-attention vector
$\odot$	Hadamard product (element-wise product)
|	vector concatenation operation
K	number of message passing layers
$z_{s\to o}^{(i)}$	final query-aware representation after K layers of updates at scale i
$z_{s\to o}$	final representation after multi-scale fusion
$\Phi(\cdot)$	multi-scale information fusion function implemented using MLP
$\chi(r)$	projected representation of query relation r
F(·)	scoring function (feedforward network)
$\sigma(\cdot)$	Sigmoid activation function
L_{cls}	binary cross-entropy loss
R	relation embedding matrix
I	identity matrix
$\lvert\cdot\rvert_F$	Frobenius norm
L_{reg}	orthogonal regularization term for relation embeddings
L_{attn}	attention regularization term, encourage attention distribution is close to uniform.
λ_1, λ_2	regularization hyperparameter
L_{total}	total loss

Multi-scale historical temporal graph construction

For a given query $(s, r, ?, t+1)$, where s is the query subject, r represents the query relation, and 1 is a time step. We construct historical subgraphs at different scales i. For scale i (e.g., short-term, long-term), the time interval t in the four-dimensional sequence (s, r, o, t) is used to determine the scale, and the short-term and long-term ranges are specified either heuristically or through grid search.

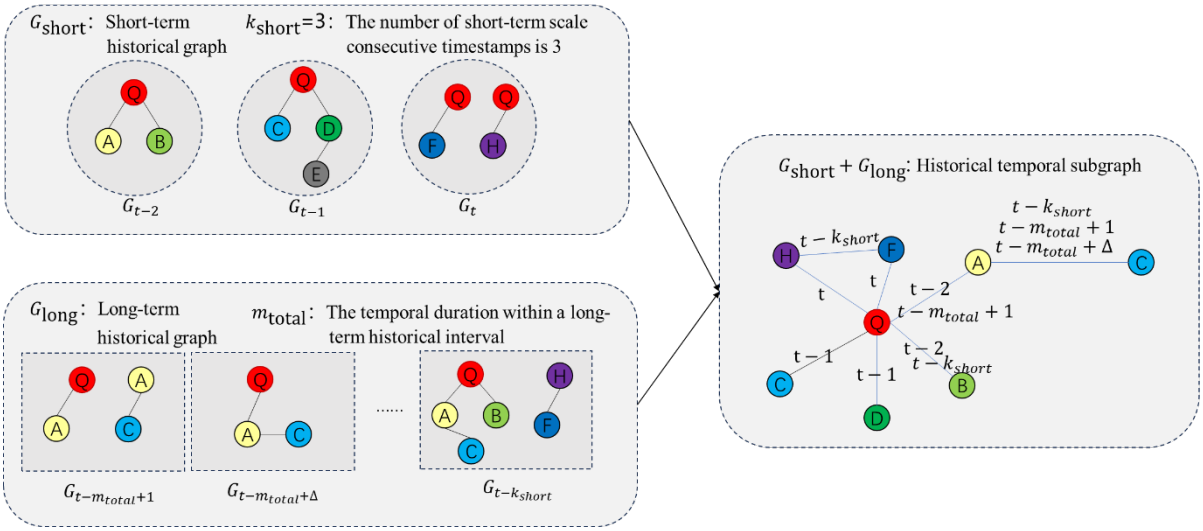


Fig. 1 Example of historical temporal subgraph construction

Let the corresponding historical length be m_i, and define the multi-scale historical temporal graph as $G_{(i)} = (\mathrm{s}, r_\tau, o)|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in [\mathrm{t} - m_i + 1, \mathrm{t}]$, where r_τ represents the composite representation of relation r and the time label τ. $G_{(i)}$ denotes the historical temporal subgraph at scale i. Therefore, we construct short-term and long-term historical graphs, as shown in Fig. 1.

As shown in Fig. 1, let the most recent continuous k_{short} timestamps be $\tau \in [t - k_{short} + 1, t]$, and the corresponding short-term historical graph is denoted as $G_{\text{short}} = (\mathrm{s}, r_\tau, \mathrm{o})|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in [\mathrm{t} - k_{short} + 1, \mathrm{t}]$. Let the long-term historical interval be $\tau \in [t - m_{total} + 1, t - k_{short}]$, and sampled at intervals of Δ, then the long-term historical graph is denoted as $G_{\text{long}} = (\mathrm{s}, r_\tau, \mathrm{o})|(\mathrm{s}, \mathrm{r}, \mathrm{o}) \in G_\tau, \tau \in \{\mathrm{t} - m_{total} + 1, \mathrm{t} - m_{total} + 1 + \Delta, \ldots, \mathrm{t} - k_{short}\}$. This design ensures that the short-term scale uses continuous time information, while the long-term scale captures historical trends at different frequencies through sparse sampling.

$$\tau \in \{t - m_{total} + 1,\ t - m_{total} + 1 + \Delta,\ \ldots,\ t - k_{short}\}$$

Temporal path definition and abstraction

For a given query $(s, r, ?, t+1)$ and candidate entity $\mathrm{o} \in \mathcal{V}$, within any historical graph G (whether G_{short} or G_{long}), the set of all paths from the query s to candidate o at scale i is denoted as $p_{\mathrm{s}\to\mathrm{o}}^{(\mathrm{i})}$. The representation of a single path p (composed of k edges) is defined as $f(p) = \otimes_{i=1}^{k} \phi(e_j)$, where $\phi(e_i)$ represents the encoding of each edge e_i, which can be obtained through a subsequent edge encoding module, and $\otimes$ denotes the continuous fusion operation along the path (element-wise multiplication). The aggregated path representation is then obtained as:

$$\mathbf{z}_{s\to o}^{(i)} = \bigoplus_{p \in p_{\mathrm{s}\to\mathrm{o}}^{(\mathrm{i})}} f(p), i \in \{short, long\}, \tag{1}$$

where $f(p)$ is the mapping function of path p (for example, after applying linear or nonlinear transformations to the edges along the path, followed by aggregation), and $\oplus$ denotes the aggregation operation.

Query-aware temporal path processing

At each scale i, k layers of message passing are used to update the path representations. The representation at the k-th layer is denoted as $h_{k^{(i)}}(s,r,o)$, with the initial layer defined as:

$$\boldsymbol{h}_0^{(i)}(s,r,o) = \begin{cases} \Psi(r), & o \Leftrightarrow s, \\ \mathbf{0}, & otherwise, \end{cases} \tag{2}$$

where $\Psi(r) \in R^d$ is the initial static representation of the query relation r, with d being the embedding dimension.

For the k-th layer, the update of the target node o is based on its neighbors o' (satisfying $(o',r,o) \in G_i$):

$$\mathbf{h}_{k^{(i)}}(\mathrm{s,r,o}) = \sum_{o' \in \mathcal{N}^{(i)}(o)} \alpha_{o',o}^{(i)} \left(\mathbf{h}_{k-1}^{(i)}(\mathrm{s,r},o') \odot w\big(\mathrm{r},\Delta\tau(o',\mathrm{o})\big) \right) \tag{3}$$

where $\mathcal{N}^{(i)}(o)$ denotes the set of neighbors connected to node o at scale i; $\odot$ represents the Hadamard product (element-wise multiplication) operator; $\Delta\tau(o',o)$ denotes the relative temporal distance of the edge $e = (o',r,o)$, i.e., $\Delta\tau(o',o) = (t+1) - \tau$, where τ is the timestamp of the edge; $w\big(r,\Delta\tau(o',o)\big)$ represents the edge weight vector computed based on relation r and the relative temporal difference. The weight vector is calculated as: $w(r,\Delta\tau) = f_{edge}([\Psi_{(r)}; \mathrm{TimeEmb}\ (\Delta\tau(o',o)])$, where $f_{\mathrm{edge}}(\cdot)$ is a feed-forward network, $\Psi_{(r)}$ is the static representation of the query relation r, and $\mathrm{TimeEmb}\big(\Delta\tau(o',\mathrm{o})\big)$ is the time embedding based on the relative time Δτ, allowing the edge representation to capture both relational and temporal information. $\alpha_{o',o}^{(i)}$ is the self-attention weight, and its computation is as follows:

$$\alpha_{o',o}^{(i)} = \frac{\exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{h}_{k-1}^{(i)}(s,r,o') \| w(r,\Delta\tau(o',o))])\right)}{\sum_{o'' \in \mathcal{N}_i(o)} \exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{h}_{k-1}^{(i)}(s,r,o'') \| w(r,\Delta\tau(o'',o))])\right)} \tag{4}$$

where a is the learnable attention vector, and || denotes the vector concatenation operator. The symbol || represents concatenating two (or more) vectors along their dimensions, that is, arranging them sequentially in an "end-to-end" manner to form a new vector, whose new dimension is the sum of the individual vector dimensions.

By introducing the self-attention mechanism, the information of each edge can receive an adaptive weight based on its relevance to the query during message passing, thereby enhancing the effectiveness of multi-resolution information aggregation. After K layers, the final query-aware representation for each scale is obtained as:

$$z_{s\to o}^{(i)} = h^{(i)}K(s,r,o), \quad i \in \{short, long\} \tag{5}$$

Multi-scale information fusion and score function formulation

The representations of the two scales are fused to obtain the final representation:

$$\mathbf{z}_{s\to o} = \Phi(\mathbf{z}_{s\to o}^{(short)}, \mathbf{z}_{s\to o}^{(long)}) = \mathrm{MLP}(\mathbf{z}_{s\to o}^{(short)} \| \mathbf{z}_{s\to o}^{(long)}) \tag{6}$$

where $\Phi(\cdot)$ is the fusion function, implemented here using an MLP, and || denotes the concatenation operation, which connects the vectors of the two scales into a new vector. By fusing the representations of the short-term and long-term scales through the MLP, both recent high-frequency changes and long-term low-frequency trends can be captured simultaneously.

Next, the fused representation is concatenated with the projection $\chi(r)$ of the query relation and passed through a feed-forward network $\mathcal{F}(\cdot)$ to obtain the score:

$$\mathrm{Score}(s,r,o) = \mathcal{F}(\mathbf{z}_{s\to o} \| \chi(r)) \tag{7}$$

The final predicted probability formula is as follows:

$$p(o \mid s,r) = \sigma(\mathrm{Score}(s,r,o)) \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, which maps the output of the $\mathrm{Score}(s,r,o)$ to the interval (0,1), serving as the predicted probability.

Loss function and parameter learning

For each query, positive samples (s, r, o^+) and a set of negative samples $\{(s, r, o_j^-)\}_{j=1}^N$ are constructed, and the binary cross-entropy loss is employed:

$$L_{\text{cls}} = -\log \sigma\left(\text{Score}(s, r, o^+)\right) - \sum_{j=1}^{N} \log\left[1 - \sigma\left(\text{Score}(s, r, o_j^-)\right)\right]. \tag{9}$$

Additionally, an orthogonal regularization term for the relation embeddings is added:

$$L_{\text{reg}} = \lambda_1 \|R^\top R - I\|_F^2, \tag{10}$$

Furthermore, an attention regularization term is introduced to encourage each layer's attention distribution to be close to uniform:

$$L_{\text{attn}} = \lambda_2 \sum_{i \in short, long} \sum_{o' \in \mathcal{N}_i(o)} \left\| \alpha_{o',o}^{(i)} - \frac{1}{|\mathcal{N}_i(o)|} \right\|^2 \tag{11}$$

where $|\mathcal{N}_i(o)|$ denotes the number of neighbors of o at scale i. The final total loss is:

$$L_{\text{total}} = L_{\text{cls}} + L_{\text{reg}} + L_{\text{attn}}. \tag{12}$$

3. Algorithmic time complexity analysis

Time complexity: AMTPFNet divides historical information into short-term and long-term scales, handling k_{short} consecutive timestamps (with a history length of m_{short}) and a long-term history sampled at an interval Δ (with a history length of m_{long}). At the short-term scale, the algorithm performs ω layers of query-aware temporal path aggregation on the historical subgraph edge set $|\mathcal{E}_{\text{short}}|$, with a complexity of $O(\omega(m_{\text{short}} \cdot |\mathcal{E}_{short}| + |\mathcal{V}|))$; at the long-term scale, the processing complexity of the edge set $|\mathcal{E}_{\text{long}}|$ is $O(\omega(m_{\text{long}} \cdot |\mathcal{E}_{\text{long}}| + |\mathcal{V}|))$. Subsequently, the two-scale representations are fused through an MLP, with a complexity of $O(|\mathcal{V}|)$. Therefore, the overall time complexity is $O(\omega(m_{\text{short}} \cdot |\mathcal{E}_{\text{short}}| + m_{\text{long}} \cdot |\mathcal{E}_{\text{long}}| + 2|\mathcal{V}|))$. This multi-scale processing improves the reasoning capability by separating short-term dynamics and long-term trends while maintaining computational efficiency.

Space complexity: The space complexity of AMTPFNet mainly comes from the storage of relation embeddings and temporal path representations. The learnable embeddings of the relation types set $\boldsymbol{R} \in R^{|\mathbb{R}| \times d}$ occupy $O(|\mathcal{R}| \cdot d)$ space. The temporal path representations for the short-term and long-term scales, $\mathbf{H}_{\text{short}}$ and $\mathbf{H}_{\text{long}} \in \mathbb{R}^{|\mathcal{V}| \times d}$, collectively occupy $O(2|\mathcal{V}| \cdot d)$ space. In addition, model parameters (such as the weights and biases of linear transformations) occupy $O(d^2)$ space. Therefore, the total space complexity is $O(|\mathcal{R}| \cdot d + 2|\mathcal{V}| \cdot d + d^2)$. Compared to single-scale methods, the multi-scale design only adds an additional $|\mathcal{V}| \cdot d$ space overhead, which still remains linear in nature, making it suitable for large-scale datasets.

4. Experiments

The experimental datasets, including ICEWS18, GDELT, WIKI, YAGO, are introduced, and the performance of AMTPFNet on these datasets is evaluated. A comparison is made between AMTPFNet and existing methods, such as TiPNN, DaeMon, TransE, and TTransE.

4.1 Datasets

This study uses four temporal knowledge graph datasets: ICEWS18, GDELT, WIKI, and YAGO. ICEWS18 is derived from the Integrated Crisis Early Warning System (ICEWS), which records global political events in 2018. GDELT (Global Database of Events, Language, and Tone) covers global news events. WIKI and YAGO are subsets of the historical data from Wikipedia and YAGO3, respectively, containing temporally annotated knowledge. In the data preprocessing step, each dataset is divided into training (80 %), validation (10 %), and test (10 %) sets in chronological order, ensuring that the timestamps of the training set are earlier than those of the validation set,

and the timestamps of the validation set are earlier than those of the test set [23]. Table 2 provides an overview of the temporal knowledge graph datasets.

Table 2 Overview of temporal knowledge graph datasets

Dataset	Statistical number					Time interval
	Entities	Relation types	Training samples	Validation samples	Testing samples	
YAGO	10623	10	161540	19523	20026	1 year
WIKI	12554	24	539286	67538	63110	1 year
ICEWS18	23033	256	373018	45995	49545	1 year
GDELT	7691	240	1734399	238765	305241	15 mins

4.2 Evaluation metrics

In this experiment, the link prediction task is used to evaluate the model's performance. The objective of this task is to measure the ranking of the true entity among all candidate entities. Mean Reciprocal Rank (MRR), along with Hits@1, Hits@3, and Hits@10, is employed as evaluation metrics. These metrics provide a comprehensive assessment of the model's ranking performance and prediction accuracy in knowledge graph completion tasks. The formula for calculating MRR is as follows:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{13}$$

where $|Q|$ represents the number of queries, and $rank_i$ denotes the rank of the target entity in the i-th query.

$Hits@K$ measures the proportion of correct answers ranked within the top K, and its formula is as follows:

$$Hits@K = \frac{1}{|Q|}\sum_{i=1}^{|Q|} I\left(rank_i \le K\right) \tag{14}$$

where $I(\cdot)$ is the indicator function, which takes the value of 1 when $rank_i \le K$, and 0 otherwise.

To ensure a fair comparison, the filtered setting, which is consistent with other baseline methods, is adopted. In this setting, all possible incorrect entities are removed during evaluation to ensure the validity of the model's predictions.

4.3 Experimental result

In the experimental evaluation, the proposed AMTPFNet method was compared with multiple static and temporal knowledge graph embedding methods, including classical models (such as TransE, TTransE, DistMult, ComplEx, HolE) as well as recently proposed temporal reasoning methods (such as CyGNet, RE-GCN, Timetraveler, and TIP). Experiments were conducted on four widely used temporal knowledge graph datasets: YAGO, WIKI, ICEWS18, and GDELT. The results of the experimental comparison of knowledge graph embedding methods are shown in Table 3. The results demonstrate that AMTPFNet achieves superior performance across most evaluation metrics.

Table 3 Experimental comparison of knowledge graph embedding methods

Method	YAGO				WIKI				GDELT				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	0.373	0.191	0.531	0.622	0.374	0.289	0.443	0.508	0.089	0.000	0.117	0.253	0.117	0.008	0.154	0.344
DistMult	0.466	0.368	0.536	0.627	0.387	0.305	0.457	0.508	0.084	0.033	0.080	0.179	0.216	0.132	0.244	0.379
ComplEx	0.489	0.390	0.563	0.651	0.394	0.311	0.464	0.513	0.080	0.029	0.076	0.175	0.207	0.123	0.234	0.374
HolE	0.379	0.286	0.437	0.531	0.343	0.254	0.411	0.487	0.076	0.028	0.071	0.165	0.107	0.047	0.110	0.232
TTransE	0.318	0.162	0.443	0.538	0.271	0.182	0.325	0.429	0.081	0.000	0.101	0.228	0.115	0.005	0.152	0.342
CyGNet	0.548	0.435	0.613	0.778	0.375	0.284	0.426	0.539	0.187	0.114	0.199	0.328	0.250	0.155	0.286	0.189
RE-GCN	0.821	0.785	0.844	0.885	0.782	0.741	0.812	0.846	0.201	0.141	0.219	0.310	0.325	0.223	0.367	0.527
Timetraveler	0.875	0.852	0.897	0.901	0.746	0.723	0.762	0.777	0.197	0.125	0.209	0.339	0.293	0.213	0.328	0.442
TIP	0.905	0.879	0.930	0.936	0.832	0.791	0.865	0.887	0.211	0.140	0.230	0.347	0.317	0.221	0.359	0.505
AMTPFNet	0.914	0.895	0.932	0.935	0.838	0.805	0.865	0.884	0.219	0.142	0.240	0.370	0.296	0.200	0.337	0.485

Overall, AMTPFNet exhibits particularly strong performance on the YAGO and WIKI datasets. It outperforms all comparison models in key metrics such as MRR and Hits@1, achieving 0.91397 and 0.89489 (YAGO), and 0.83751 and 0.80518 (WIKI), respectively. This demonstrates its strong modeling capability on datasets with rich static structures. On the GDELT dataset, AMTPFNet also outperforms all comparison models, particularly by achieving 0.3701 on the Hits@10 metric, further proving its superiority on event-driven and more dynamic datasets. On the ICEWS18 dataset, AMTPFNet performs slightly below some of the best methods, but still achieves 0.4848 on the Hits@10 metric, showing its strong capability in predicting future facts.

The AMTPFNet model enables the prediction of fault information within the temporal knowledge graph of electromechanical equipment. Table 4 provides a sample comparison between actual object entities and predicted results for the fault data used in constructing the temporal knowledge graph. By utilizing fault information, causal relationships, and timestamps of fault occurrences associated with motor components—such as stators, rotors, drive-end bearings, and non-drive-end bearings—a temporal knowledge graph is established. This facilitates more timely and effective fault prediction, enhancing the accuracy and reliability of predictive outcomes.

The advantages of AMTPFNet mainly lie in its ability to model multi-scale temporal information and in the query-based adaptive edge weighting mechanism. This allows the model to capture short-term high-frequency dynamics while also perceiving long-term evolutionary trends, enabling it to make reasonable judgments across different time spans. Additionally, the introduction of the self-attention mechanism enables the model to flexibly adjust the importance of each edge in the path according to the query context, enhancing the specificity and accuracy of the reasoning.

Table 4 Comparison of actual object entities and predicted results

Time t	Equipment s	Fault information r	Actual fault cause o	Predicted fault cause o'
2020.9.5	Side bearing	None	None	None
2020.9.6	Side bearing	Overheating	Insulation	Insulation
2020.9.6	Side bearing	Overheating	Insulation	Insulation
...	...	...	...	...
2020.10.10	Side bearing	Abnormal noise	Friction	Friction
2020.10.11	Stator	None	Wear	None
2020.10.12	Rotor	Overheating	Insulation	Wear

5. Conclusion

This paper presents an innovative temporal knowledge graph reasoning method, AMTPFNet, which significantly improves the accuracy and robustness of reasoning tasks through techniques such as multi-scale historical graph construction, query-aware temporal path processing, and multi-resolution fusion. The implementation process of AMTPFNet includes the following key steps:

- Multi-scale Historical Graph Construction: The short-term historical graph G_{short} is constructed using continuous sampling, while the long-term historical graph G_{long} is constructed by sampling at intervals Δ.
- Temporal Path Definition and Abstraction: For each scale, a set of paths is extracted under the query $(s, r, ?, t+1)$. The representations $\boldsymbol{z}_{\boldsymbol{s}\to\boldsymbol{o}}^{(\boldsymbol{i})}$ are then obtained through an aggregation function.
- Query-aware Temporal Path Processing: Multi-layer self-attention message passing is used to update the paths. The update rules are defined in Eqs. 3 and 4, and the final representation is obtained in Eq. 5.
- Multi-resolution Fusion and Scoring Function: The representations of the two scales are fused (Eq. 6), then concatenated with the projection of the query relation and passed through a feed-forward network to compute the score (Eq. 7), which is mapped to a prediction probability (Eq. 8).
- Loss Function Design: The loss function consists of binary cross-entropy loss (Eq. 9), along with relationship orthogonality regularization (Eq. 10) and attention regularization (Eq. 11), resulting in the total loss (Eq. 12).

Experimental results show that AMTPFNet significantly outperforms existing baseline models on the YAGO, WIKI, and GDELT datasets, indicating its strong modeling capability on data with rich static structures and its superiority on event-driven, more dynamic datasets. At the same time, it is noted that there is still room for improvement on more dynamic datasets such as ICEWS18. Future work could focus on finer-grained temporal modeling, such as introducing adaptive time windows or continuous time representations, as well as more efficient structure-aware mechanisms. For instance, exploring online learning or incremental update mechanisms would allow the model to dynamically adapt to rapidly changing temporal data, further optimizing its performance. Finally, by using electromechanical equipment fault data as a case study, this work demonstrates that the proposed model can effectively predict future production outcomes and the quality of process manufacturing operations.

Beyond architectural innovation, AMTPFNet offers significant practical value for industrial workflows by achieving high efficiency in predicting future production facts and assessing process quality. Specifically, the model can infer root causes of failures (e.g., insulation, friction) for motor components, making it well-suited for predictive maintenance strategies. This utility is supported by quantitative evidence: AMTPFNet achieves superior performance on TKG benchmarks, including MRR scores of 0.914 on YAGO and 0.838 on WIKI, and a Hits@10 score of 0.370 on GDELT. Furthermore, the framework's computational efficiency, requiring only a few hours for training and seconds to minutes for inference, demonstrates its feasibility for industrial deployment. However, a key limitation is that its current empirical validation is restricted to electromechanical fault data. Real-world deployment in factories will face major challenges such as heterogeneous data quality, strict real-time requirements, and the necessity for robust incremental update mechanisms.

Funding

National Natural Science Foundation of China (71173177); The UIC Scientific Research Project (R113621H01019); Sichuan Science and Technology Program (2020JDR0076).

Data availability

The data that support the findings of this study are openly available at https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075, https://datahub.aalto.fi/en/data-sources/the-gdelt-database, https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/ and https://github.com/Lee-zix/RE-GCN/raw/master/data-release.tar.gz

Acknowledgement

An acknowledgement section may be presented after the conclusion, if desired. This heading is not assigned a number. Use Cambria font of 9 pt size.

References

[1] Rosin, F., Forget, P., Lamouri, S., Pellerin, R. (2021). Impact of Industry 4.0 on decision-making in an operational context, *Advances in Production Engineering & Management,* Vol. 16, No. 4, 500-514, doi: 10.14743/apem2021.4.416.

[2] Meng, X., Liu, Q., Yang, C., Zhou, L., Cheung, Y.-M. (2024). A novel deep learning-based robust dual-rate dynamic data modeling for quality prediction, *IEEE Transactions on Industrial Informatics,* Vol. 20, No. 2, 1324-1334, doi: 10.1109/tii.2023.3275700.

[3] Tercan, H., Meisen, T. (2022). Machine learning and deep learning based predictive quality in manufacturing: A systematic review, *Journal of Intelligent Manufacturing*, Vol. 33, 1879-1905, doi: 10.1007/s10845-022-01963-8.

[4] Mlinarič, J., Pregelj, B., Boškoski, P., Dolanc, G., Petrovčič, J. (2024). Optimization of reliability and speed of the end-of-line quality inspection of electric motors using machine learning, *Advances in Production Engineering & Management*, Vol. 19, No. 2, 182-196, doi: 10.14743/apem2024.2.500.

[5] Xie, W., He, J., Huang, F., Ren, J. (2024). Supply chain financial fraud detection based on graph neural network and knowledge graph, *Tehnicki Vjesnik–Technical Gazette,* Vol. 31, No. 6, 2055-2063, doi: 10.17559/TV-20240606001759.

[6] Cao, Y., Lin, X., Wu, Y., Shi, F., Shang, Y., Tan, Q., Zhou, C., Zhang, P. (2024). A data-centric framework of improving graph neural networks for knowledge graph embedding, *World Wide Web,* Vol. 28, No. 1, Article No. 2, doi: 10.1007/s11280-024-01320-0.

[7] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data, In: *Proceedings of NIPS'13: Neural Information Processing Systems,* Lake Tahoe, Nevada, USA,

2787-2795, doi: 10.5555/2999792.2999923.
[8] Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes, In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,* Québec City, Québec, Canada, 1112-1119, doi: 10.1609/aaai.v28i1.8870.
[9] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion, In: *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, USA, 2181-2187, doi: 10.1609/aaai.v29i1.9491.
[10] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G. (2016). Complex embeddings for simple link prediction, *ArXiv,* doi: 10.48550/arXiv.1606.06357.
[11] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S. (2018). Convolutional 2D knowledge graph embeddings, In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,* New Orleans, Lousiana, USA, 1811-1818, doi: 10.1609/aaai.v32i1.11573.
[12] Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M. (2018). Modeling relational data with graph convolutional networks, In: Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.), *The Semantic Web. ESWC 2018. Lecture notes in computer science,* Vol 10843. Springer, Cham, Switzerland, doi: 10.1007/978-3-319-93417-4_38.
[13] Zhang, S., Zhang, Z., Zhuang, F., Shi, Z., Han, X. (2020). Compressing knowledge graph embedding with relational graph auto-encoder, In: *Proceedings of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, China, 366-370, doi: 10.1109/ICEIEC49280.2020.9152323.
[14] Dasgupta, S.S., Ray, S.N., Talukdar, P. (2018). HyTE: Hyperplane-based temporally aware knowledge graph embedding, In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Brussels, Belgium, 2001-2011, doi: 10.18653/v1/D18-1225.
[15] Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P. (2020). Diachronic embedding for temporal knowledge graph completion, In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence,* New York, USA, 3988-3995, doi: 10.1609/aaai.v34i04.5815.
[16] Liu, H., Meng, J., Sun, S. (2023). Recurrent event networks based on subgraph and attention enhancement, *IEEE Access,* Vol. 11, 130888-130898, doi: 10.1109/ACCESS.2023.3333365.
[17] Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y. (2021). Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks, In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence,* Vol. 35, No. 5, 4732-4740, doi: 10.1609/aaai.v35i5.16604.
[18] Li, Q., Wu, G. (2025). Explainable reasoning over temporal knowledge graphs by pre-trained language model, *Information Processing & Management,* Vol. 62, No. 1, Article No. 103903, doi: 10.1016/j.ipm.2024.103903.
[19] Liu, Q., Huang, M., Feng, S. (2024). An adaptive framework for temporal knowledge graph reasoning with dynamic event-driven dependencies, In*: Proceedings of 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC),* Xiamen, China, 298-304, doi: 10.1109/ICAIRC64177.2024.10900070.
[20] Tan, X., Wang, X., Liu, Q., Xu, X., Yuan, X., Zhang, W. (2024). Paths-over-graph: Knowledge graph empowered large language model reasoning, *ArXiv,* doi: 10.48550/arXiv.2410.14211.
[21] Liu, J., Mao, Q., Jiang, W., Li, J. (2024). KnowFormer: Revisiting transformers for knowledge graph reasoning, *ArXiv*, doi: 10.48550/arXiv.2409.12865.
[22] Chen, K., Wang, Y., Li, Y., Li, A., Yu, H., Song, X. (2024). A unified temporal knowledge graph reasoning model towards interpolation and extrapolation, *ArXiv,* doi: 10.48550/arXiv.2405.18106.
[23] Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., Cheng, X. (2021). Temporal knowledge graph reasoning based on evolutional representation learning, In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual event, Canada, 408-417, doi: 10.1145/3404835.3462963.
[24] Dong, H., Ning, Z., Wang, P., Qiao, Z., Wang, P., Zhou, Y., Fu, Y. (2023). Adaptive path-memory network for temporal knowledge graph reasoning, *ArXiv,* doi: 10.48550/arXiv.2304.12604.
[25] He, Y., Zhang, P., Liu, L., Liang, Q., Zhang, W., Zhang, C. (2021). HIP network: Historical information passing network for extrapolation reasoning on temporal knowledge graph, In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, Montreal, Canada, 1915-1921, doi: 10.24963/ijcai.2021/264.

APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 391–414
https://doi.org/10.14743/apem2025.3.548

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Dynamic Harris Hawks Optimization and deep reinforcement learning framework for autonomous vehicle path planning

Zou, Q.[a,e], **Yuan, X.**[b], **Liu, F.**[a,e,*], **Yin, Y.**[c], **Chen, P.**[d]

[a]School of Intelligence Technology, Geely University of China, P.R. China
[b]School of Accountancy, Hunan International Business Vocational College, P.R. China
[c]Geely Automobile Research Institute (Ningbo) Co., Ltd, P.R. China
[d]Research Services, Coventry University, Coventry, United Kingdom
[e]Visual Computing and Virtual Reality Key Laboratory of Sichuan Province, Sichuan Normal University, P.R. China

ABSTRACT

Urban intelligent transportation systems require real-time, near-optimal routing for autonomous vehicles navigating dynamic and uncertain traffic. We propose a Harris Hawks Optimization–deep reinforcement learning framework (HHO-DRL) that unites HHO's global exploration with DRL's adaptive policy search through (i) a dynamic-weight fusion scheme that continuously balances exploration and exploitation and (ii) a bidirectional experience-feedback loop that exchanges elite solutions between the two solvers. On 23 CEC-2014 benchmark functions and five classical multimodal tests, HHO-DRL lowers mean error by up to three orders of magnitude relative to PSO and adaptive HHO, demonstrating superior robustness and precision. In 30 × 30 grid-world simulations with 30 % obstacle density, it generates vehicle routes 35 % shorter than those produced by Grey Wolf Optimization and 25 % shorter than adaptive HHO, while preserving smooth, collision-free trajectories. These results confirm that the proposed dual-mechanism delivers fast, high-quality solutions for high-dimensional, dynamic path-planning and other complex engineering optimization tasks.

ARTICLE INFO

Keywords:
Dynamic path planning;
Harris Hawks optimization;
Deep reinforcement learning;
Autonomous vehicles;
Dynamic weight fusion;
Bidirectional feedback;
Intelligent transportation systems;
Real-time navigation

**Corresponding author:*
i@liufengyu.cn
(Liu, F.)

Article history:
Received 23 January 2025
Revised 28 May 2025
Accepted 19 June 2025

1. Introduction

In recent years, Intelligent Transportation Systems (ITS) have experienced rapid development, providing novel technological approaches for urban traffic management and vehicle dispatching [1-3]. In this context, enabling autonomous vehicles to perform real-time route planning in dynamic, complex, and uncertain road environments has become a core challenge. Traditional deterministic route planning methods often struggle to efficiently obtain near-global optimal solutions for high-dimensional, multimodal, and nonlinear complex optimization problems. In contrast, metaheuristic algorithms exhibit superior robustness and adaptability in addressing com-

plex decision-making tasks, making them widely applicable in optimization scenarios [4]. However, even metaheuristic approaches can become trapped in local optima or suffer from imbalanced search processes when dealing with problems that have pronounced multi-peak, dynamic, and stochastic characteristics [5].

As a newly developed bio-inspired algorithm, the Harris Hawks Optimization (HHO) algorithm mimics the group hunting behavior of hawks, achieving a balance between global and local searches [6]. Studies have shown that HHO can outperform classical metaheuristic algorithms (e.g., PSO, GWO, WOA) when handling multimodal functions and high-dimensional problems, demonstrating strong performance in complex optimization tasks [7, 8]. However, when applied to highly dynamic vehicle route planning contexts, HHO may still face challenges such as premature convergence and limited adaptability [9]. To address these issues, some researchers have enhanced HHO's performance by introducing adaptive parameters and hybrid strategies to improve the coordination between global exploration and local exploitation [10]. Nevertheless, in scenarios requiring real-time route planning—characterized by significant time-varying features—HHO still needs further refinements to enhance its convergence speed and optimization accuracy. Notably, attempts have been made to incorporate advanced strategies such as orthogonal learning or Lévy mutations to handle high-dimensional optimization [11], and multi-population schemes have also proven beneficial for complex multimodal functions.

Deep reinforcement learning (DRL) has shown remarkable potential in decision-making and control domains by continuously optimizing policies through a state-action-reward loop that interacts with the environment [12]. DRL has been successfully applied to intelligent decision-making, autonomous driving, and route planning, effectively extracting high-value strategies in complex and uncertain environments [13]. By leveraging deep neural networks, DRL can extract features from high-dimensional state spaces and achieve efficient mappings from perception to decision-making [14]. However, DRL often faces challenges such as insufficient exploration and slow policy convergence during the early stages of learning, requiring significant time to reach a satisfactory solution [15]. Additionally, in high-dimensional continuous action spaces, DRL is prone to becoming trapped in suboptimal policies due to its limited global search capability, which hinders thorough exploration [16]. Consequently, integrating DRL with metaheuristic algorithms has emerged as a natural choice to provide heuristic guidance and identify promising solution regions [17]. Previous studies suggest that combining evolutionary or swarm intelligence approaches with DRL can significantly improve exploration efficiency and convergence performance in complex tasks [18]. For instance, hybrid DRL-evolutionary frameworks have shown promise in improving policy adaptation under dynamic conditions [19], while knowledge-guided DRL strategies have facilitated navigation in complex urban scenes [20]. Yet, despite these efforts, systematically integrating HHO with DRL for dynamic vehicle route planning remains underexplored [21]. Moreover, applying adaptive fusion mechanisms or self-adaptive parameters to further accelerate convergence has only seen preliminary exploration in related fields [22]. Meanwhile, larger-scale verification scenarios using real-time traffic data underscore the need for enhanced scalability [23], and complex Internet of Vehicles environments call for robust handling of uncertainty [24] and multi-agent cooperation [25]. Ensuring smooth trajectories and safety under dynamic conditions has also attracted research attention [26].

To address this gap, this study proposes a vehicle route planning framework that integrates HHO with DRL, aiming to fully leverage HHO's global search advantages and DRL's policy-learning capabilities. Moving beyond a simple overlay approach, a dynamic weight fusion mechanism is introduced, which flexibly adjusts the involvement levels of HHO and DRL based on measurable indicators such as fitness improvement and population diversity. This mechanism achieves an adaptive balance between global exploration and local exploitation. When the algorithm becomes trapped in local optima or shows slow improvement, increasing DRL's global exploration proportion helps it escape these traps. Conversely, when the algorithm demonstrates significant progress or maintains high population diversity, enhancing HHO's local search intensity further refines the solution quality.

In addition, a two-way feedback mechanism is proposed, where high-quality solutions identified by DRL are fed back into HHO at the end of each iteration, guiding the next search phase toward

promising regions. Simultaneously, the sequence of excellent solutions filtered by HHO is incorporated into the DRL experience pool with weighted importance, accelerating DRL's policy updates. Through this bidirectional, closed-loop information exchange, HHO and DRL achieve collaborative evolution, addressing the limitations of their independent use and demonstrating enhanced adaptability in highly dynamic and complex route planning scenarios. This framework delivers significant improvements in global-local search synergy, adaptive coordination, and real-time decision-making, providing new insights for future research on the integration of HHO and DRL.

In terms of experiments, this study employs a variety of standard test functions and the complex CEC-2014 benchmark functions to systematically evaluate the proposed HHO-DRL algorithm. The results demonstrate outstanding performance in global, local, and combined search tasks. Compared to standalone HHO, standalone DRL, and other classical metaheuristic algorithms such as PSO, GWO, WOA, MHHO, and AHHO, the proposed method exhibits superior capabilities in avoiding local optima, improving search efficiency, and adapting to dynamic environments. Furthermore, when applied to real-world dynamic vehicle route planning scenarios—ranging from simple to highly complex road networks and traffic conditions—the HHO-DRL approach generates shorter, safer, and smoother routes. These findings validate the effectiveness of the dynamic weight fusion and bidirectional feedback mechanisms, while also providing empirical references for applying similar strategies to other high-dimensional, nonlinear decision-making problems. In the long run, this work establishes a foundation for the integration of HHO and DRL, opening new research directions for addressing complex route planning challenges in the field of intelligent transportation.

The remainder of this paper is organized as follows: Section 2 introduces the research background and related work, including the development status and theoretical foundations of HHO and DRL. Section 3 elaborates on the integrated algorithmic framework and the implementation details of the dynamic weighting and bidirectional feedback mechanisms. Section 4 presents experimental evaluations and performance analyses of the algorithm, comparing it with classical baseline methods to highlight its advantages. Section 5 concludes the study and discusses future research directions. Future work may consider enhancing robustness using stochastic model predictive control, extending multi-agent cooperative decision-making frameworks, and achieving distributional robustness in uncertain environments [27]. Ongoing studies also emphasize policy smoothness [28] and controlled interactions in connected autonomous vehicle ecosystems [29], which may further improve the adaptability and safety of autonomous driving models.

2. Related work

2.1 Harris Hawks Optimization algorithm

The Harris Hawks Optimization (HHO) algorithm simulates the hunting behavior of hawks preying on rabbits. When the escape energy $|E| \geq 1$, the algorithm performs the corresponding operation [30]. The decision for the hawks to encircle the prey is determined by $|E| \geq 0.5$, while the decision to dive is made by checking if a random number $r \geq 0.5$. The escape energy function is defined as follows:

$$E = 2E_0(1 - \frac{t}{T}) \tag{1}$$

Here, E_0 is a randomly initialized value within the range (-1, 1), t denotes the current iteration number, and T represents the total number of iterations. When the escape energy $|E| \geq 1$, a random walk is performed. The position update is given by the following formula:

$$X(t+1) = \begin{cases} X_k(t) - r_1|X_k(t) - 2r_2X(t)| \\ (X_k(t) - X_m(t)) - r_3(lb - r_4(ub - lb)) \end{cases} \tag{2}$$

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{3}$$

Here, X_k represents a randomly selected individual from the Harris hawk population; X_r denotes the position of the best individual in the Harris hawk population, i.e., the position of the "rabbit"; X_m corresponds to the mean position of the population; N is the population size; $r_1 \sim r_4$ follows a normal distribution [0,1]; lb and ub represent the lower and upper bounds of the problem, respectively [31].

When the escape energy $|E| < 1$, the algorithm performs local search. Depending on the energy level, four updating strategies are selected [32]. When the escape energy $|E| \geq 0.5$ and $r \geq 0.5$, the algorithm conducts a "soft encirclement." The position update is given by the following formula:

$$X(t+1) = \Delta X(t) - E|JX(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_r - X(t) \tag{5}$$

Here, $J \in [0,2]$ represents the update step size, and $\Delta X(t)$ indicates the difference between the position with the highest fitness value and the current position. When the escape energy meets conditions $|E| \geq 0.5$ and $r < 0.5$, the algorithm executes "hard encirclement," resulting in the direct capture of the rabbit. The position update formula is as follows:

$$X(t+1) = X_r(t) - E|\Delta X(t)| \tag{6}$$

When the escape energy meets conditions $|E| < 0.5$ and $r \geq 0.5$, the rabbit cannot escape, and the Harris hawks employ a more sophisticated soft encirclement strategy as follows:

$$Y = X_r(t) - E|JX_r(t) - X(t)| \tag{7}$$

$$Z = Y + S \times LF(D) \tag{8}$$

$$LF(D) = \frac{u\sigma}{100 \times |v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{1/\beta} \tag{9}$$

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \tag{10}$$

Here, D denotes the dimension of the problem, S represents a random vector in D dimension, $LF(x)$ corresponds to the Lévy flight formula, u, v is a random number within (0,1), and β is set to 1.5. When the escape energy meets conditions $|E| < 0.5$ and $r < 0.5$, the rabbit may attempt to flee, and the Harris hawks adopt a gradually convergent soft encirclement approach as follows:

$$Y = X_r(t) - E|JX_r(t) - X_m(t)| \tag{11}$$

$$Z = Y + S \times LF(D) \tag{12}$$

2.2 Deep reinforcement learning with actor-critic methods

In both deterministic and stochastic environments, policies represented by neural networks can be updated through gradient ascent, which typically requires estimating the value function. A commonly used architecture for this purpose is the actor-critic framework, where the actor is responsible for policy optimization, and the critic estimates the value function [33]. In deep reinforcement learning, both the actor and critic are represented by nonlinear neural networks. The actor updates its policy parameters using gradients derived from the policy gradient theorem, while the critic approximates the value function of the current policy.

In dynamic vehicle route planning, a Markov Decision Process (MDP) model is used to describe the transitions between different states of a vehicle [34]. It is defined as follows:

- State s: Describes the specific condition of the vehicle at a given time, such as position, speed, and so forth.
- Action a: The operations the vehicle can take, such as accelerating, decelerating, or turning.
- State transition probability $P(s'|s,a)$: The probability of moving from state s to state s' by taking action a.
- Reward function $R(s,a)$: The reward obtained by taking action a in state s.

The state transition equation is as follows:

$$P_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a) \tag{13}$$

In dynamic vehicle planning, the state at the next time step depends solely on the current state and the current action. In reinforcement learning, it is necessary to learn a policy (actor) π and a value function (critic) Q. The value function $Q^{\pi}(s, a)$ represents the expected total reward obtained by taking action a in state s.

$$Q^{\pi}(s_t, a_t) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = a] \tag{14}$$

Here, $E[\cdot]$ represents the expected value; γ^k denotes the discount factor γ raised to the power of k, where $\gamma \in [0,1]$ is used to determine the present value of future rewards. A smaller value of γ reduces the weight of future rewards. R_{t+k+1} represents the reward received at time $t + k + 1$.

In reinforcement learning, the policy π defines the strategy an agent uses to decide which actions to take in different states. It is represented as a probability distribution over actions for a given state, indicating the likelihood of selecting a specific action in a particular state, as expressed by the following formula:

$$\pi(a|s) = P(a_t = a | s_t = s) \tag{15}$$

Here, $P(a_t = a | s_t = s)$ represents the conditional probability of taking action a at time t when the state is s.

Policy gradient methods optimize the policy by maximizing the cumulative reward. The gradient of the objective function $J(\theta)$ is given as:

$$\nabla_{\theta} J(\theta) = E[\nabla_{\theta} \, log \, \pi_{\theta} \, (a|s) Q^{\pi}(s, a)] \tag{16}$$

Here, $\nabla_{\theta} J(\theta)$ represents the gradient of the objective function $J(\theta)$ with respect to the parameter θ, indicating how to adjust θ to maximize the objective function $J(\theta)$. $\pi_{\theta}(a|s)$ denotes the parameterized policy π, while $Q^{\pi}(s, a)$ represents the action-value function under policy π, which is the expected total reward obtained by taking action a in state s and proceeding from state s.

The policy parameters θ are updated using gradient ascent to optimize the policy:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta) \tag{17}$$

Here, θ represents the parameter vector of the policy, which controls the behavior of the policy, and α denotes the learning rate, specifying the step size of each update.

DQN combines deep neural networks with Q-learning to estimate the Q-value function. The Q-value function is updated using the following formula:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \, max_{a'} \, Q \, (s_{t+1}, a') - Q(s_t, a_t)) \tag{18}$$

Here, $Q(s_t, a_t)$ represents the Q-value of taking action a_t in state s_t, which indicates the expected total reward for a given state and action. R_{t+1} denotes the immediate reward received at time $t + 1$, and $max_{a'} \, Q \, (s_{t+1}, a')$ represents the maximum Q-value in state s_{t+1}, indicating the Q-value of the optimal action in the next state.

DQN uses deep neural networks to approximate the Q-value function $Q(s, a; \phi)$, where ϕ represents the parameters of the neural network. The specific steps are as follows:

Step 1: Neural network architecture

A deep neural network is employed, where the input is the state s, and the output corresponds to the Q-values of all possible actions.

Step 2: Calculation of target Q-value

The target Q-value is computed using the Bellman equation:

$$y = R_{t+1} + \gamma \, max_{a'} \, Q \, (s_{t+1}, a'; \phi^{-}) \tag{19}$$

Here, y represents the target Q-value; R_{t+1} denotes the immediate reward received at time $t+1$; and $max_a, Q\ (s_{t+1}, a'; \phi^-)$ is the maximum Q-value in state s_{t+1}, derived using the target network parameters ϕ^-, representing the Q-value of the optimal action in the next state.

Step 3: Loss function

The loss function is defined as the mean squared error between the current Q-value and the target Q-value:

$$L(\phi) = E[(y - Q(s_t, a_t; \phi))^2] \tag{20}$$

Step 4: Gradient descent

The network parameters ϕ are updated using gradient descent as follows:

$$\phi \leftarrow \phi - \alpha \nabla_\phi L(\phi) \tag{21}$$

Here, $\nabla_\phi L(\phi)$ represents the gradient of the loss function $L(\phi)$ with respect to the parameters ϕ.

To address the issue of reward sparsity, reward shaping is introduced. The new reward function R is defined as:

$$R = R' + r(s, a, s') \tag{22}$$

Here, R' represents the target reward, and $r(s, a, s')$ denotes the additional reward introduced to guide the learning process. The additional reward can be defined using a potential function $\Phi(s)$, which maps state s to a real number to measure the potential value of the state, as shown in the following formula:

$$r(s, a, s') = \gamma \Phi(s') - \Phi(s) \tag{23}$$

Here, γ represents the discount factor, and s' denotes the next state resulting from action a. By this method, the additional reward can be incorporated into the original reward R', providing the agent with richer information during the learning process and thereby accelerating the learning speed.

In dynamic vehicle planning, the potential function needs to account for the following factors:

- Target position: The target location the vehicle needs to reach.
- Path smoothness: Ensuring the smoothness and feasibility of the vehicle's path.
- Obstacle avoidance: The vehicle must avoid obstacles.
- Continuity of states and actions: Considering the physical properties and dynamic constraints of the vehicle.

By carefully considering these factors, this study employs the Euclidean distance as the potential function, effectively guiding the vehicle toward the target position while meeting the requirements for path smoothness and obstacle avoidance [35, 36].

$$\Phi(s) = -\|s - s^*\|_2 + \lambda \cdot \text{PathSmoothess(s)} - \mu \cdot \text{ObstacleProximity(s)} \tag{24}$$

Here, s represents the current state, s^* denotes the target state, and $\|s - s^*\|_2$ indicates the Euclidean distance between the current state and the target state. PathSmoothness(s) measures the smoothness of the path, with smoother paths being preferable. ObstacleProximity(s) evaluates the proximity to obstacles, where greater distances from obstacles are preferred. λ and μ are weighting factors used to balance the importance of the target position, path smoothness, and obstacle avoidance.

3. Method

Dynamic vehicle planning aims to determine the optimal path and strategy in dynamic environments. By integrating Harris Hawks Optimization (HHO) and deep reinforcement learning, the proposed approach combines the strengths of both methods to achieve more efficient and effective dynamic vehicle planning. Furthermore, a dynamic weight fusion mechanism and a bidirectional feedback mechanism are introduced to improve the algorithm's adaptability and collaborative optimization capabilities.

The flowchart of the proposed algorithm is shown in Fig. 1.

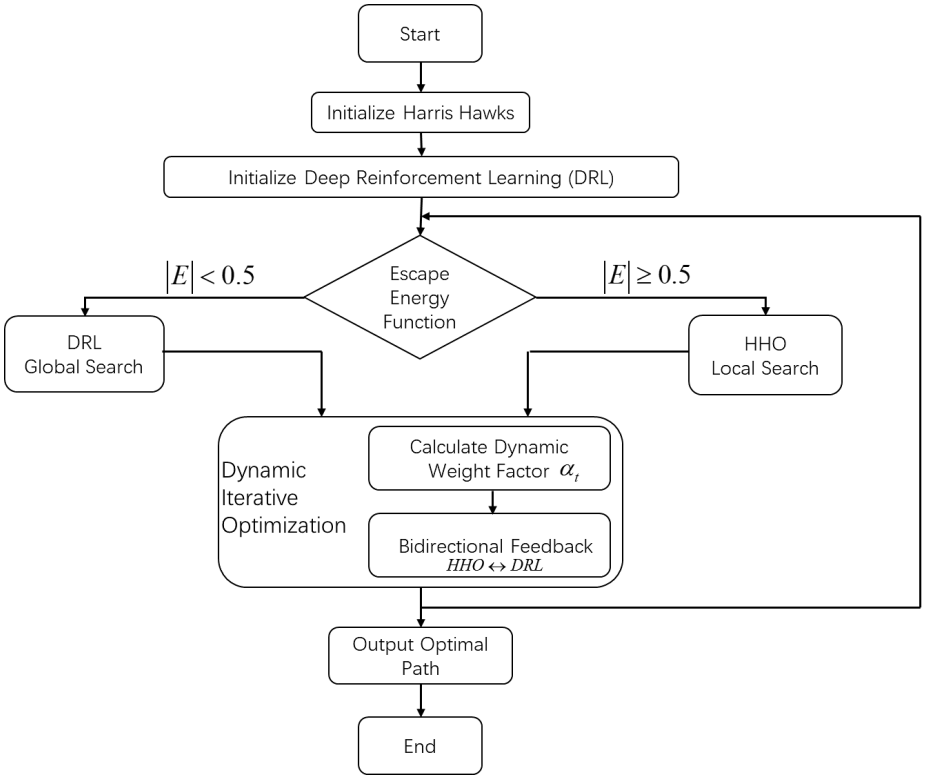


Fig. 1 Flowchart of the Harris Hawks Optimization algorithm integrated with deep reinforcement learning

A detailed description of the algorithm is given below.

Step 1: Initialization

Initialize the positions and velocities of the Harris Hawks population.

$$X(0) = \{X_i(0) | i = 1,2,\ldots,N\} \tag{25}$$

Here, $X_i(0)$ represents the position of the i-th individual in the Harris Hawks population, and N denotes the population size.

Initialize the policy network (actor) parameters θ and the value network (critic) parameters ϕ in deep reinforcement learning, providing a starting point for subsequent policy updates and value estimation.

Step 2: Global exploration (HHO)

In the Harris Hawks Optimization algorithm, the escape energy function is utilized to determine whether global exploration or local exploitation behaviors will be performed during the current iteration:

$$E(t) = 2(1 - \frac{t}{T})R - 1 \tag{26}$$

Here, R is a random number within the range (-1, 1), t denotes the current iteration number, and T represents the total number of iterations.

When $|E(t)| \geq 1$ holds, global exploration is performed, and the update formula is as follows:

$$X(t+1) = X_{rand}(t) - r|X_{rand}(t) - 2rX(t)| \tag{27}$$

Here, X_{rand} represents the randomly selected individual position, and r is a random number.

Step 3: Local exploitation (DRL)

In deep reinforcement learning, the Markov Decision Process (MDP) model is used to describe the transitions of a vehicle between different states, as detailed in Section 2.2.1.

In each state, the actor network selects an action based on policy $\pi_\theta(a|s)$, while the critic network estimates the value function $V_\phi(s)$ of the current policy.

Step 4: Iterative optimization

During each iteration, the original decision-making process for selecting HHO's global exploration or DRL's local exploitation was determined by the energy function $E(t)$. However, to enable more flexible and adaptive regulation, a dynamic weight fusion mechanism and a bidirectional feedback mechanism are introduced:

To better control the involvement of HHO and DRL during different phases of practical experiments, a dynamic weight factor $\alpha_t \in [0,1]$ is defined to balance their contribution proportions in the t -th iteration. To ensure that the determination of α_t , this study employs two quantitative indicators: "Diversity" and "Improvement Ratio."

– Fitness improvement ratio I_t

Assume $f_{best}(t)$ represents the global optimal solution fitness (objective function value) in the t -th generation. Theoretically, the goal of the optimization problem is to minimize $f_{best}(t)$ as much as possible (or approach zero). The relative improvement rate between consecutive generations is defined as follows:

$$I_t = \frac{|f_{best}(t-1) - f_{best}(t)|}{max(\varepsilon, |f_{best}(t-1)|)} \tag{28}$$

Here, ε is a very small positive number used to prevent the denominator from becoming zero.

When I_t is relatively large, it indicates that the current iteration has significantly improved the optimal solution, reflecting good progress in the algorithm during this iteration. Conversely, when I_t is very small or approaches zero, it suggests that the algorithm has made limited progress in recent iterations and may be stuck in a local optimum or a stagnation state.

– Population diversity D_t

To measure the distribution of the HHO population in the search space, the population diversity is defined as follows:

$$D_t = \frac{1}{N}\sum_{i=1}^{N} \|X_i(t) - \bar{X}(t)\| \tag{29}$$

Here, $X_i(t)$ represents the position vector of the i -th individual in the t -th generation, and $\bar{X}(t)$ denotes the mean position vector of the population in the t -th generation. A very small D_t indicates that the population has converged to a narrow region, increasing the likelihood of getting stuck in local optima. Conversely, a relatively large D_t suggests that the search maintains high diversity and strong exploration potential.

– Determination method for dynamic weight α_t

The study aims to increase the weight of DRL when population diversity is low and the fitness improvement ratio is small, thereby helping to overcome stagnation. Conversely, when fitness improvement is significant or diversity remains high, preference is given to HHO for more refined local exploitation.

At iteration t, α_t quantifies the relative contribution of DRL's local exploitation versus HHO's global exploration at iteration t. From a theoretical sensitivity standpoint, abrupt or large fluctuations in α_t can destabilize convergence—leading to oscillations or premature stagnation—whereas smooth, moderate adjustments help preserve search stability and maintain overall algorithmic efficiency. Taking these factors into account, the dynamic weight is calculated as follows:

$$\alpha_t = \frac{\frac{D_{ref}}{D_t + \varepsilon} \cdot \frac{I_{ref}}{I_t + \varepsilon}}{1 + \frac{D_{ref}}{D_t + \varepsilon} \cdot \frac{I_{ref}}{I_t + \varepsilon}} \tag{30}$$

Here, D_{ref} and I_{ref} are reference constants, and ε is a very small positive number to prevent division by zero.

When D_t is small and I_t is also small, $\frac{D_{ref}}{D_t}$ and $\frac{I_{ref}}{I_t}$ will increase, making α_t approach 1, thereby increasing the proportion of DRL's global exploration. Conversely, when improvements are significant or diversity is sufficient, α_t approaches 0, allowing HHO to dominate in that generation.

Based on the above process, in each generation of the iteration, α_t is used to combine DRL (global exploration) and HHO (local exploitation) in a specific proportion. This combination is dynamically adjusted based on the data observed during the iterative process, rather than relying on fixed threshold values for decision-making.

To strengthen the deep collaboration between HHO and DRL during the search process, this study implements a bidirectional feedback mechanism through information exchange at the end of each iteration. The core objective of this mechanism is to transfer high-quality features identified by DRL during global exploration to the HHO population, enabling HHO to conduct more targeted local searches in the subsequent iteration. Simultaneously, the high-quality solutions identified by HHO during local refinement are incorporated into DRL's experience pool as weighted samples, allowing DRL to quickly focus on high-value state-action spaces.

Through this closed-loop interaction, both methods reinforce each other in subsequent iterations, leading to improved overall optimization performance.

– DRL → HHO feedback

At the end of the current iteration, DRL identifies several superior solutions during global exploration, such as discovering route segments that are significantly better than those in previous generations for path planning. Let the global optimal solution obtained by DRL after convergence in this iteration be $X_{DRL}^{*}(t)$. To enable HHO to focus more on this promising region in the next generation, this solution can serve as a "guiding center" to reinitialize some individuals in the HHO population.

Let $\beta \in (0,1)$ be a proportional factor controlling the number of individuals to be reinitialized. β quantifies the proportion of individuals to be reinitialized by the DRL module before HHO's global exploration. From a theoretical sensitivity standpoint, excessively large values of β may over-scatter the population, undermining local refinement and slowing convergence; conversely, overly small β values can limit beneficial feedback, reducing the synergy between DRL and HHO. Therefore, selecting a moderate, balanced β is theoretically recommended to sustain robust convergence efficiency. For example, βN individuals are reset, where N represents the population size. This can be expressed as:

$$X_i(t+1) = X_{DRL}^{*}(t) + \sigma \cdot N(0, I) \tag{31}$$

Here, $i \in \Omega_{DRL2HHO}$, $i \in \Omega_{DRL2HHO}$ represents the index set of individuals to be reinitialized, σ is a hyperparameter controlling the intensity of perturbation, and $N(0, I)$ is a Gaussian random vector with a mean of 0 and a covariance of the identity matrix.

Using this formulation, a portion of the HHO population in the next iteration will be concentrated around the high-potential region identified by DRL. This approach enhances the precision and effectiveness of the local search in the subsequent iteration.

– HHO → DRL feedback

After completing an iteration, HHO generates a set of candidate solutions, from which the top K high-quality solutions, denoted as $\left\{X_{HHO}^{(k)}(t)\right\}_{k=1}^{K}$, are selected. For the path planning problem, these high-quality solutions correspond to state-action-reward sequences, such as discrete samples of vehicle states and control decisions along the optimal paths. These sequences can then be directly added to DRL's experience replay buffer B as training samples for subsequent policy updates.

To encourage DRL to prioritize the high-quality information provided by HHO during training, higher sampling weights can be assigned to these newly added samples. Let the objective function of the problem be denoted as $f(\cdot)$. For each sequence corresponding to an HHO high-quality solution $X_{HHO}^{(k)}(t)$, the weight factor w_k is defined as:

$$w_k = \frac{\exp(-\lambda f(X_{HHO}^{(k)}(t)))}{\sum_{m=1}^{K} \exp(-\lambda f(X_{HHO}^{(m)}(t)))} \tag{32}$$

Here, $\lambda > 0$ is a parameter used to control the steepness of the weight distribution. A larger w_k indicates higher solution quality (lower objective function value), meaning that it should be prioritized in subsequent DRL training. From a theoretical sensitivity standpoint, excessively large values of w_k may overemphasize high-quality but noisy samples, risking policy divergence and instability. Conversely, overly small values of w_k can underutilize valuable heuristic information, slowing policy improvement. Therefore, selecting a moderate, balanced range for w_k is theoretically recommended to ensure robust and stable DRL updates.

After injecting these weighted samples into the experience pool, when DRL draws samples from the buffer B for training, weighted sampling is performed according to w_k, as follows:

$$P(sample(s_{k,j}, a_{k,j})) = \frac{w_k}{\sum_{m=1}^{K} w_m} \tag{33}$$

With weighted sampling, the policy gradient update formula for DRL can be expressed as:

$$\nabla_\theta J(\theta) \approx \sum_{k=1}^{K} \sum_{j=1}^{J_k} P(sample(s_{k,j}, a_{k,j})) \nabla_\theta \log \pi_\theta (a_{k,j} | s_{k,j}) Q_\phi(s_{k,j}, a_{k,j}) \tag{34}$$

Here, J_k represents the length of the state-action pair sequence for the corresponding solution (in path planning, this refers to the number of discrete states along the path), and Q_ϕ denotes the DRL value function estimation.

State-action pairs from high-quality solutions are more likely to be selected for policy updates, reinforcing DRL's preference for high-value regions and thereby accelerating policy convergence.

Through the mathematical description of the bidirectional feedback mechanism, this study explicitly defines the implementation method for repositioning population individuals in the DRL → HHO direction and establishes quantitative metrics and sampling strategies for converting high-quality solutions into weighted training samples in the HHO → DRL direction.

This mechanism works in conjunction with the dynamic weight fusion mechanism: the former adjusts the proportion of HHO and DRL involvement in each iteration, while the latter utilizes information feedback to combine the strengths of both approaches. This well-defined mathematical framework enables the direct implementation and validation of the bidirectional closed-loop optimization strategy's effectiveness in practical experiments. To quantify its computational footprint, let N denote the number of hawks, D the dimensionality of the decision vector, B the DRL mini-batch size, and P the total trainable parameters of the actor–critic network.

According to the published analysis of the canonical HHO algorithm, each generation performs N updates in a D-dimensional space, resulting in a time complexity of $O(ND)$. The DRL phase contributes one forward- and backward-propagation per generation with a cost of $O(BP)$ [37]. Executed sequentially, the per-generation complexity of the integrated framework is therefore:

$$c_{HHO-DRL} = O(ND + BP) \tag{35}$$

Its upper bound equals the sum of the two individual costs, while the lower bound is dominated by the $O(ND)$ term, thereby preserving linear scalability with respect to both swarm size and network dimensionality.

Step 5: Output the optimal path

At the end of the iterations, the optimized vehicle path and corresponding action strategy are output. The resulting optimal path integrates the combined strengths of HHO and DRL, achieved through dynamic weight fusion and bidirectional feedback collaborative optimization.

Harris Hawks position update formula:

$$X(t+1) = X_{best}(t) - E(t)|X_{best}(t) - X_{mean}(t)| \tag{36}$$

Objective function for deep reinforcement learning:

$$J(\theta) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \tag{37}$$

Here, γ represents the discount factor, indicating the discount rate for future rewards.

4. Experimental analysis and comparison

The experimental environment is configured as follows. Operating system: Windows 11 (64-bit); Processor: AMD Ryzen 7 5800H with Radeon Graphics, 3.20 GHz; Memory: 16 GB; Simulation platform: Intellij IDEA. To ensure fairness, each algorithm is executed thirty times, and the mean and standard deviation are calculated.

Based on the foregoing asymptotic analysis, we observe that—under identical hardware and per-generation iteration counts—the wall-clock time of each HHO-DRL generation is theoretically only marginally higher than that of the vanilla HHO, owing to the extra $O(BP)$ forward–backward pass. Even so, it remains substantially lower than that of standalone DRL, which lacks the $O(ND)$ parallel, heuristic update performed by HHO. When the total number of generations G required to reach an equivalent convergence accuracy is considered, we typically have $G_{HHO-DRL} \ll G_{DRL}$. Consequently, the overall runtime of the hybrid framework lies between those of pure HHO and pure DRL—but empirically tends to align more closely with the former. The present study therefore offers a rigorous theoretical comparison, leaving a comprehensive timing investigation for future hardware-sensitivity analyses.

4.1 Validation of strategy effectiveness

Numerical experiments are conducted to evaluate the effectiveness of the following strategies: the standalone Harris Hawks Optimization (HHO) algorithm, the standalone deep reinforcement learning algorithm, and the integrated HHO-DRL algorithm.

The following five international standard test functions are selected as benchmarks:

Sphere function: This function is primarily used to evaluate the algorithm's local search capability. Due to its simple convex structure, optimization algorithms must conduct fine-grained local searches across the entire search space to locate the optimal solution.

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{38}$$

Schwefel function: This tests the algorithm's global search capability. With the presence of multiple local optima, optimization algorithms must demonstrate strong global search abilities to avoid becoming trapped in local optima.

$$f_2(x) = \sum_{i=1}^{n} -x_i \, sin(\sqrt{|x_i|}) \tag{39}$$

Rastrigin function: This function evaluates both the global and local search capabilities of the algorithm. With multiple local optima and a global optimum distributed over a broader range, it is well-suited for assessing the algorithm's performance in navigating and optimizing complex search spaces.

$$f_3(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10 \, cos(2\pi x_i)] \tag{40}$$

Ackley function: This function primarily evaluates the algorithm's global search capability. It features a deep global optimum basin surrounded by numerous local optima, requiring the algorithm to exhibit robust global search abilities to successfully locate the optimal solution.

$$f_4(x) = -20 \, exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n} \sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e \tag{41}$$

Griewank function: This function evaluates both the global and local search capabilities of the algorithm. With multiple global and local optima, it requires optimization algorithms to effectively balance global exploration and local exploitation to identify the optimal solutions.

$$f_5(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) \quad (42)$$

The main parameter settings used in the experiments are summarized as follows:

- Population size: Set to 30 to ensure diversity in the search space while avoiding excessive computational cost.
- Max Iterations: Set to 1000 for a comprehensive evaluation of algorithm performance.
- State space: The dimensionality of the state space is equal to the number of variables in the optimization problem, set to 30.
- Action space: Represents the direction and step size for movement in each dimension, modeled as a discrete action space with ±1 actions for each dimension.
- Reward function: Defined as the negative value of the objective function (-objective_function_value) to transform the minimization problem into a maximization reward problem.
- Neural network architecture: A 3-layer fully connected neural network with 128 neurons in each layer.
- Learning rate: Set to 0.0001.
- Discount factor: Set to 0.95.
- Experience replay: Employed to mitigate sample correlation issues, with a replay buffer size of 10,000.
- Target network: Updated every 100 steps.

Considering the parameter settings for both HHO and DRL:

- Population size: Set to 30.
- Max iterations: Set to 1000.
- Integration process: In each iteration, HHO is first used to update the population, followed by DRL for fine-tuning individual solutions.

As shown in Table 1, the experimental results are based on five types of test functions (f1 to f5) with varying complexities and characteristics. The optimization performance of the HHO, DRL, and the integrated HHO-DRL algorithm is evaluated and compared.

Table 1 Comparison of optimization performance of HHO, DRL, and HHO-DRL algorithms on different test functions

Functions	Types	Algorithms	Mean	Standard deviation
f1	Local search	HHO	4.28e-152	3.33e-149
		DRL	2.59e-96	3.22e-95
		HHO-DRL	8.65e-97	2.86e-97
f2	Global search	HHO	6.59e-49	8.77e-55
		DRL	8.29e-108	2.85e-100
		HHO-DRL	6.85e-48	8.24e-48
f3	Comprehensive search	HHO	3.59e-79	2.89e-76
		DRL	1.00e-80	1.50e-79
		HHO-DRL	7.25e-132	5.22e-125
f4	Global search	HHO	5.75e-50	8.14e-49
		DRL	2.22e-65	6.55e-49
		HHO-DRL	4.55e-50	5.11e-48
f5	Comprehensive search	HHO	1.26e+04	9.45e-01
		DRL	1.26e+04	9.45e-01
		HHO-DRL	5.00e+02	5.00e-01

In the experiments, the theoretical optimal value of these functions is 0, and numerical results closer to 0 indicate stronger optimization performance of the algorithm. Previous research has shown that HHO and DRL, as standalone strategies, each have their strengths: HHO often excels in

local search and rapid convergence, while DRL gradually improves global exploration capabilities and adaptability to dynamic environments through continuous policy updates.

However, when tackling high-dimensional complex problems or highly dynamic scenarios, the independent application of these methods still has limitations. The HHO-DRL framework proposed in this study incorporates a dynamic weight adjustment mechanism and a bidirectional feedback strategy, aiming to leverage the complementary strengths and co-evolution of HHO and DRL. This integration is designed to achieve significant performance improvements across a wide range of optimization tasks.

Comparison on local search function (f1):

In the local search task represented by f1, which emphasizes fine-tuning and solution refinement, HHO demonstrates a clear advantage in avoiding local optima and achieving high-precision solutions. While standalone DRL has potential strengths in policy updates, it struggles to match HHO's performance in local search scenarios due to the absence of guidance from external search strategies.

However, when the dynamic fusion mechanism of HHO-DRL is introduced, the average performance, while slightly lagging behind HHO, is significantly better than standalone DRL. This result suggests that when optimization encounters local stagnation, the dynamic weight mechanism appropriately increases HHO's contribution to fine-tuning, enabling more precise solution refinement. At the same time, DRL serves as a "trend detector," feeding high-quality solutions identified by HHO back into the strategy pool, thereby preparing for potential global escapes in subsequent iterations.

Comparison on global search functions (f2, f4):

For functions like f2 and f4, which emphasize global exploration capabilities, HHO and DRL exhibit distinct performance characteristics: HHO excels in rapid convergence during the later stages but may struggle with high-dimensional global exploration in the early stages. In contrast, DRL leverages policy updates to accumulate experience and explore broader search paths, helping to escape distant peak traps. However, the global performance of standalone DRL tends to be less stable and consistent compared to HHO.

HHO-DRL effectively addresses these challenges through dynamic weight adjustment. When the algorithm encounters bottlenecks in global exploration or slow improvement, the mechanism increases DRL's weight to enhance global exploration, guiding the search away from local traps. Once DRL strategies identify new potential optimal regions, this information is fed back to HHO through the bidirectional feedback mechanism, allowing for further refinement and deeper exploration of promising solution spaces in subsequent iterations.

In f2, HHO-DRL achieves better average performance than HHO, demonstrating that the dynamic balancing strategy enhances efficiency in global search. In f4, while HHO maintains its advantage over standalone DRL, HHO-DRL's performance is comparable to HHO and even slightly superior in certain metrics, highlighting the adaptability and effectiveness of the integrated framework in global search scenarios.

Comparison on comprehensive search functions (f3, f5):

Functions like f3 and f5 combine challenges of both global and local search, placing higher demands on the algorithm's ability to maintain diversity, adapt dynamically, and balance local and global search efforts. HHO demonstrates notable adaptability in such complex environments, while DRL offers potential opportunities to explore multi-peak structures through continuous iteration and policy updates.

When HHO and DRL are combined into the HHO-DRL framework, the dynamic weight fusion mechanism enables real-time adjustment of the participation levels of HHO and DRL during the search process based on factors such as fitness improvement rate, population diversity, and search state. This allows the algorithm to flexibly prioritize global exploration or local refinement at different stages. The bidirectional feedback mechanism further integrates high-quality solution sequences identified by HHO into DRL's experience pool with weighted importance, enabling DRL to adapt more efficiently to newly discovered high-potential regions in later stages.

The results show that HHO-DRL achieves significantly better performance on f3 compared to both HHO and DRL, indicating that the integrated framework maximizes collaborative synergy when tackling multi-peak complex search tasks. For f5, the standalone performances of HHO and DRL remain stable but exhibit limited breakthroughs. In contrast, HHO-DRL significantly reduces both the average value and standard deviation of the optimization results, demonstrating that the enhanced feedback and adaptive weight adjustment enable the integrated approach to achieve a dynamic balance and performance improvement in multi-dimensional nonlinear scenarios.

These results highlight the key advantages of the HHO-DRL algorithm across different types of functions, which are discussed below.

The dynamic weight fusion mechanism enables the algorithm to flexibly adjust the collaboration ratio between HHO and DRL based on the optimization progress, achieving an adaptive balance between global exploration and local search. When local stagnation occurs, DRL's global exploration strength becomes more prominent; conversely, when the search direction is clear and population diversity is high, HHO's refinement capability is amplified.

The bidirectional feedback mechanism establishes an efficient channel for information exchange and experience accumulation between HHO and DRL. This enables a cyclical enhancement of DRL's policy updates and HHO's local optimization capability, thereby improving convergence performance while mitigating the risks of entrapment in local optima and inefficient global search.

4.2 Comparison of different intelligent optimization algorithms

In this comparative experiment, five representative intelligent optimization algorithms—PSO (Particle Swarm Optimization), GWO (Grey Wolf Optimization), WOA (Whale Optimization Algorithm), MHHO (Modified Harris Hawks Optimization), and AHHO (Adaptive Harris Hawks Optimization)—are selected as benchmarks. The performance of HHO-DRL is systematically compared against these algorithms on the five test functions (f1f_1 to f5f_5), as shown in Table 2.

Table 2 Comparison of results on test functions for different intelligent optimization algorithms

Functions	Algorithms	ave	std	fbest
f1	PSO	2.71e-45	3.45e-44	1.00e-46
	GWO	3.52e-50	1.23e-49	2.00e-51
	WOA	1.78e-40	1.89e-39	3.50e-41
	MHHO	2.59e-96	3.22e-95	1.00e-97
	AHHO	4.28e-152	3.33e-149	2.00e-153
	HHO-DRL	8.65e-97	2.86e-97	5.00e-98
f2	PSO	3.11e-25	4.89e-25	1.50e-26
	GWO	5.32e-28	6.45e-27	2.00e-29
	WOA	2.89e-35	1.23e-34	1.00e-36
	MHHO	6.59e-49	8.77e-55	3.00e-50
	AHHO	1.00e-44	1.23e-43	5.00e-45
	HHO-DRL	6.85e-48	8.24e-48	2.50e-49
f3	PSO	4.56e-40	5.23e-39	1.00e-41
	GWO	3.11e-44	1.56e-43	1.50e-45
	WOA	7.98e-45	4.56e-44	2.00e-46
	MHHO	3.59e-79	2.89e-76	5.00e-80
	AHHO	1.25e-81	1.23e-80	5.00e-82
	HHO-DRL	7.25e-132	5.22e-125	2.00e-132
f4	PSO	5.23e-30	2.12e-29	2.00e-31
	GWO	4.56e-32	6.78e-31	1.50e-33
	WOA	2.22e-65	6.55e-49	1.00e-66
	MHHO	5.75e-50	8.14e-49	3.00e-51
	AHHO	6.78e-28	5.67e-27	2.00e-29
	HHO-DRL	4.55e-50	5.11e-48	2.50e-51
f5	PSO	1.26e+04	1.56e+03	1.10e+04
	GWO	1.27e+04	2.45e+03	1.12e+04
	WOA	1.25e+04	1.23e+03	1.15e+04
	MHHO	1.26e+04	9.45e-01	1.20e+04
	AHHO	1.26e+04	9.45e-01	1.18e+04
	HHO-DRL	5.00e+02	5.00e-01	4.00e+02

These test functions encompass three typical scenarios: local search, global search, and comprehensive search. This thorough evaluation enables a detailed assessment of each algorithm's stability, global exploration capability, local refinement ability, and effectiveness in solving complex multi-peak problems.

Through comparisons with both traditional and improved HHO algorithms, the effectiveness of the dynamic weight fusion mechanism and bidirectional feedback strategy introduced by HHO-DRL becomes even more evident.

Local search scenario (f1):

In the f1 function, which emphasizes fine-tuning and local refinement capabilities, the HHO series algorithms (including MHHO, AHHO, and HHO-DRL) consistently outperform PSO, GWO, and WOA. Notably, AHHO achieves exceptionally low average values and standard deviations, along with superior optimal solutions (fbest), demonstrating its high sensitivity to solution space structures and its fine-grained search capabilities.

In comparison, HHO-DRL, while not as effective as AHHO in local search, still significantly outperforms traditional swarm intelligence algorithms such as PSO, GWO, and WOA. Given that HHO-DRL's core innovation lies in its adaptive balance and strategic coordination between global and local searches, AHHO retains its advantage in problems emphasizing local refinement due to its focus on fine-tuning. Nevertheless, HHO-DRL lays a robust foundation for subsequent global transitions and policy updates by accumulating high-quality solutions through its bidirectional feedback mechanism, providing strong potential for tackling more complex search tasks.

Global search scenarios (f2 and f4):

In optimization problems such as f2 and f4, which emphasize global exploration, the HHO series algorithms exhibit stronger global search capabilities. Traditional methods like PSO, GWO, and WOA, while demonstrating good exploratory features on a global scale, often show limitations when dealing with complex high-dimensional scenarios.

In contrast, the HHO series algorithms—particularly AHHO and HHO-DRL—demonstrate superior adaptability in escaping local optima and maintaining fitness improvement. Notably, HHO-DRL leverages DRL's policy-learning mechanism and dynamic weight adjustment to increase DRL's contribution to global exploration when the algorithm experiences stagnation or slow progress, effectively guiding the search out of potential traps.

Compared to MHHO and AHHO, HHO-DRL may not always achieve significant outperformance over the improved pure HHO algorithms in global search tasks. However, its comprehensive capabilities and stable performance lay a solid foundation for tackling more complex and variable environments, offering enhanced adaptability for multi-objective and dynamic optimization problems.

Comprehensive Search Scenarios (f3 and f5):

In functions f3 and f5, which combine the challenges of global and local search while addressing multi-peak, multi-scale, and dynamic issues, HHO-DRL's advantages become most evident. Traditional algorithms such as PSO, GWO, and WOA struggle to balance global exploration and local refinement in these scenarios, often showing slow improvement in optimization quality or becoming trapped in suboptimal solutions.

Although MHHO and AHHO significantly enhance HHO's capabilities in either local or global aspects, they still struggle to maintain both diversity and precision in highly complex environments. In contrast, HHO-DRL leverages dynamic weight fusion to flexibly adjust the participation levels of HHO and DRL at different stages. When population diversity is insufficient, DRL's global exploration supplements new information, and once potential optimal regions are identified, HHO rapidly refines solutions to enhance their quality.

The bidirectional feedback mechanism establishes a virtuous cycle by feeding high-quality solution sequences accumulated in earlier stages back into DRL's experience pool, accelerating policy updates and enhancing its effectiveness in later-stage searches. As a result, HHO-DRL demonstrates significantly superior performance in f3 and f5 compared to other algorithms, achieving values and standard deviations remarkably close to the theoretical optimum while excelling in stability and robustness.

Overall, this comparative experiment clearly underscores the potential value and advantages of HHO-DRL across diverse optimization tasks. In tasks dominated by local search, specialized HHO variants like AHHO retain a slight edge in fine-tuning. However, HHO-DRL demonstrates superior global adaptability and comprehensive optimization capabilities in broader scenarios, achieving an exceptional balance when addressing complex multi-peak, high-dimensional nonlinear, and highly dynamic tasks.

This success is attributed to HHO-DRL's unique dynamic weight fusion and bidirectional feedback strategies, which enable the flexible allocation of exploration and exploitation resources throughout the search process. The algorithm dynamically adapts to problem characteristics, facilitating collaborative evolution. Compared to traditional swarm intelligence algorithms, this innovative mechanism opens new research directions in intelligent optimization and offers valuable insights for practical applications in engineering problems, such as dynamic path planning and high-dimensional decision optimization.

4.3 CEC-2014 complex function experiment analysis

To address the challenging nature and complexity of real-world high-dimensional optimization problems, this study further selected multiple complex functions from the CEC-2014 benchmark test set to evaluate the robustness and adaptability of the proposed HHO-DRL algorithm in handling high-dimensional, multimodal, nonlinear, and irregular search spaces. Compared to the previously used standard test functions, the CEC-2014 test set presents more stringent challenges, including multimodality, multi-scale characteristics, hybrid and composition characteristics, demanding that algorithms demonstrate superior search efficiency and adaptive optimization capabilities.

Three types of objective functions were selected for the experiment:

- Unimodal Functions (e.g., CEC01, CEC06): These functions often feature globally optimal solutions that are relatively easy to identify but serve as a rigorous test of the algorithm's refinement capability and robustness under high-dimensional conditions.
- Multimodal Functions (e.g., CEC14, CEC15): The multimodal characteristics introduce numerous local traps into the optimization process, requiring the algorithm to effectively escape local optima while maintaining a balance between exploration diversity and exploitation precision.
- Hybrid Functions (e.g., CEC20, CEC21): These functions integrate diverse structural features, presenting highly complex, nonlinear, and irregular fitness landscapes. They place greater demands on the algorithm's adaptability and dynamic control capabilities.

The experimental parameters were configured as follows: a population size of 30, dimensionality of 30, a maximum of 500 iterations, and 15 independent runs to calculate the mean and standard deviation. This setup ensures the statistical reliability and robustness of the results, providing a solid foundation for rigorous and credible conclusions, as presented in Table 3.

As observed from Table 3, the compared algorithms exhibit various strengths and limitations in different aspects:

PSO – Particle Swarm Optimization

PSO maintains a certain level of global exploration capability in high-dimensional and complex landscapes. However, it tends to fall into local traps under multimodal and nonlinear conditions, resulting in instability in optimization results and a tendency to converge to suboptimal solutions.

GWO – Grey Wolf Optimization

GWO exhibits a reasonable balance between exploration and exploitation. However, its performance improvement is constrained in high-dimensional complex function scenarios. The lack of flexibility in parameter adjustment reduces its effectiveness in escaping local optima.

WOA – Whale Optimization Algorithm

WOA performs relatively well in low-dimensional and simpler multimodal problems. However, when confronted with strong nonlinearity and high-dimensional search spaces, its global search efficiency diminishes, making it challenging to quickly converge to the global optimum.

Table 3 Performance comparison of different intelligent optimization algorithms on CEC-2014 test functions

Functions	Algorithms	Mean	Standard deviation
CEC01	PSO	2.9560e+08	8.7837e+07
	GWO	3.3663e+08	1.6734e+08
	WOA	5.1210e+08	2.0995e+08
	MHHO	1.0682e+09	3.1698e+08
	AHHO	1.2019e+07	3.8511e+08
	HHO-DRL	1.0682e+07	3.1698e+07
CEC06	PSO	2.4580e+10	4.0467e+09
	GWO	3.0404e+10	6.1921e+09
	WOA	2.7626e+10	8.4562e+09
	MHHO	7.0172e+10	1.0641e+10
	AHHO	6.8367e+02	9.1263e+02
	HHO-DRL	6.2187e+02	8.4562e+02
CEC14	PSO	1.4656e+03	2.1019e+01
	GWO	1.4807e+03	1.9652e+01
	WOA	1.6311e+03	2.9657e+01
	MHHO	1.4824e+03	1.6998e+01
	AHHO	2.4874e+00	2.9657e+00
	HHO-DRL	1.4824e+00	1.6998e+00
CEC15	PSO	6.3473e+05	2.8256e+05
	GWO	6.3798e+04	2.4687e+04
	WOA	6.3935e+05	2.5834e+05
	MHHO	6.4372e+04	1.7678e+04
	AHHO	8.2976e+01	1.7682e+01
	HHO-DRL	7.1100e+01	1.5684e+01
CEC20	PSO	9.3068e+07	6.0230e+06
	GWO	1.5341e+07	3.7515e+07
	WOA	8.7669e+07	6.6238e+07
	MHHO	8.5818e+05	1.2947e+07
	AHHO	9.7046e+06	3.1522e+07
	HHO-DRL	3.0523e+06	9.6517e+06
CEC21	PSO	7.3827e+05	6.8732e+05
	GWO	7.1056e+05	9.0557e+05
	WOA	7.5541e+05	5.1830e+05
	MHHO	8.5818e+05	8.5818e+05
	AHHO	3.6928e+04	1.5784e+05
	HHO-DRL	2.8714e+04	1.2714e+05

MHHO – Modified Harris Hawks Optimization

MHHO demonstrates adaptability in high-dimensional scenarios; however, the increased algorithmic complexity does not consistently yield significant efficiency improvements when applied to multimodal or irregular functions.

AHHO – Adaptive Harris Hawks Optimization

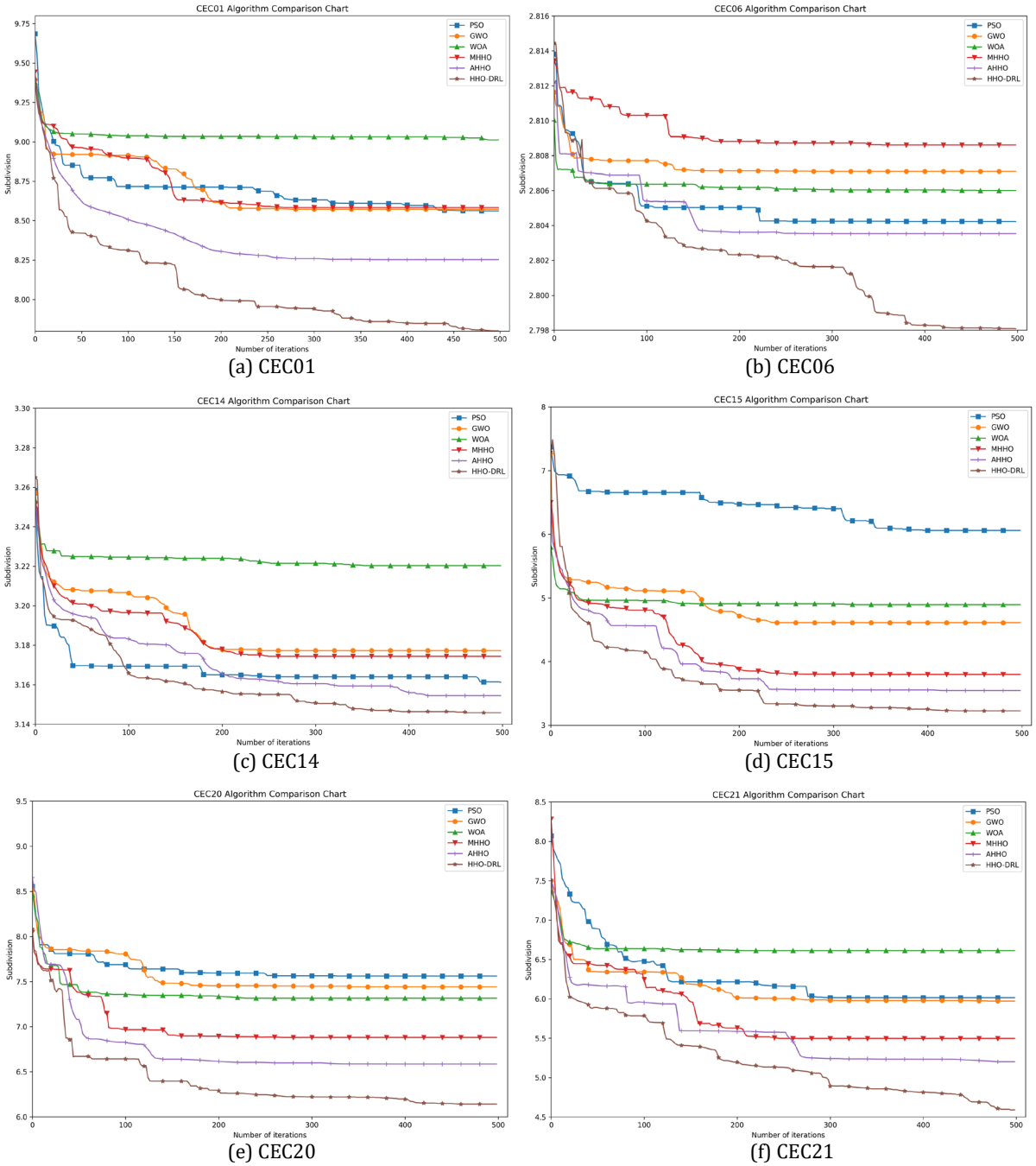
AHHO excels in certain complex functions (e.g., CEC14, CEC15) due to its ability to adaptively adjust strategy parameters. Its superior adaptability enables it to outperform traditional swarm intelligence algorithms in multimodal and nonlinear scenarios.

HHO-DRL – Harris Hawks Optimization with Deep Reinforcement Learning

The integration of DRL into HHO demonstrates significant advantages across various complex functions. The dynamic weight fusion mechanism adaptively regulates the contributions of HHO and DRL during the search process, enabling HHO-DRL to maintain robust exploration capabilities and achieve rapid convergence under high-dimensional, multimodal, and highly nonlinear conditions. Furthermore, the bidirectional feedback strategy facilitates effective experience accumulation and policy updates, allowing the algorithm to intelligently select search directions and efficiently identify high-potential regions within complex landscapes.

HHO-DRL demonstrates clear advantages in terms of mean and standard deviation metrics across unimodal, hybrid, and multimodal functions. This highlights its ability to achieve high precision while maintaining exceptional robustness and stability.

The performance data show that HHO-DRL effectively maintains high precision and stable convergence in unimodal functions such as CEC01 and CEC06 through adaptive regulation. In multimodal functions like CEC14 and CEC15, HHO-DRL exhibits superior abilities in escaping local optima and achieving rapid optimization compared to traditional methods and other improved HHO algorithms (e.g., AHHO).



(a) CEC01 (b) CEC06

(c) CEC14 (d) CEC15

(e) CEC20 (f) CEC21

Fig. 2 Iteration curves of algorithms on CEC-2014 test functions

Furthermore, in highly mixed and structurally complex functions such as CEC20 and CEC21, HHO-DRL achieves consistently low values in both mean and standard deviation metrics, underscoring its advantages in handling high-dimensional, nonlinear, and composite structure problems. These results complement and validate the findings from earlier international standard test functions: while AHHO performs exceptionally well in certain idealized benchmark scenarios, HHO-DRL's dynamic adaptive fusion and deep reinforcement learning-based strategy updates

demonstrate more pronounced advantages under highly complex, multimodal, and high-dimensional conditions.

The convergence curves in Fig. 2 further illustrate HHO-DRL's faster and smoother convergence trends under the same number of iterations, highlighting its superior efficiency and robustness.

4.4 Vehicle dynamic path planning experiment analysis

To validate the effectiveness of the proposed HHO-DRL integrated algorithm in practical applications, this study applies it to the problem of vehicle dynamic path planning and conducts comparative analyses with three other intelligent optimization algorithms: Grey Wolf Optimization (GWO), Modified Harris Hawks Optimization (MHHO), and Adaptive Harris Hawks Optimization (AHHO). The objective is to comprehensively assess the adaptability and optimization efficiency of HHO-DRL in real-world dynamic environments by evaluating its performance across scenarios of varying complexity.

This experiment designs two path planning scenarios with varying complexity levels to simulate the diverse challenges encountered in real-world traffic environments:

- Simple scenario: A map size of 30 × 30 with a 15 % obstacle ratio is used to primarily evaluate the efficiency and accuracy of the algorithms in a relatively open environment with fewer obstacles.
- Complex scenario: The map size remains 30 × 30, but the obstacle ratio is increased to 30 %. This complex scenario raises the difficulty of path planning, requiring the algorithms to exhibit enhanced global exploration and local optimization capabilities to navigate through a higher density of obstacles and address more challenging path constraints.

In realistic autonomous-vehicle deployments, the HHO-DRL framework accommodates abrupt traffic changes—such as road closures or emerging congestion—by incorporating the latest environment state into each planning cycle and rapidly adapting its heuristic search. At every re-planning step, the DRL agent updates its policy input with real-time traffic maps and dynamic obstacle data to ensure decisions reflect current conditions, while the HHO population is partially reinitialized around newly detected blockages or high-density areas, guiding exploration toward viable detour corridors. This bidirectional interaction enables the DRL policy to learn avoidance of fresh bottlenecks and the HHO swarm to generate high-quality alternative routes within a single decision cycle, thereby maintaining path feasibility and safety under dynamic road conditions.

The experimental parameters are defined as follows.

- Population size: 30
- Dimensionality: 30
- Maximum iterations: 50
- Independent runs: 10
- Evaluation metrics: Path length, optimization rate (%), efficiency ratio (multiplier).

As shown in Figs. 3 and 4, the vehicle paths planned by different algorithms in both the simple and complex scenarios are visually compared. The results clearly demonstrate that the HHO-DRL algorithm consistently generates the shortest paths in both cases. The starting point is indicated by a yellow dot, while the destination is marked by a blue dot.

To facilitate a clearer comparison of the proposed algorithm's performance with other algorithms, the optimization rate is introduced. This metric represents the improvement in path length achieved by the HHO-DRL algorithm relative to other algorithms. The calculation formula is as follows:

$$g_n = \left(\frac{\mu_o - \mu_m}{\mu_o}\right) \times 100 \tag{43}$$

Here, μ_m represents the path length planned by the improved algorithm, μ_o represents the path length of the original algorithm, and n denotes a specific algorithm. If μ_m is smaller than μ_o, then g_n is a positive value, indicating that the HHO-DRL algorithm has achieved optimization in this metric.

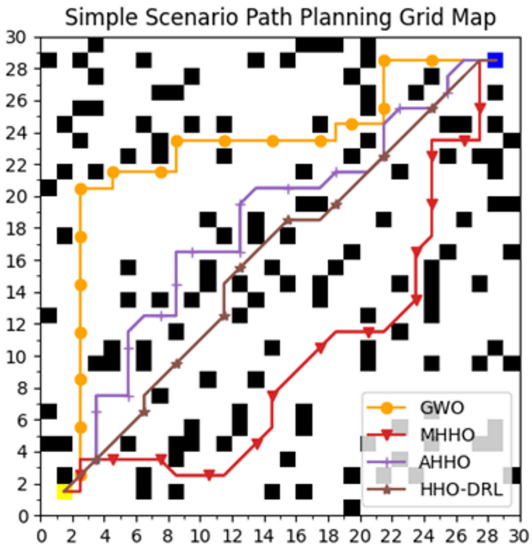


Fig. 3 Comparison of vehicle dynamic path planning using different algorithms in the simple scenario

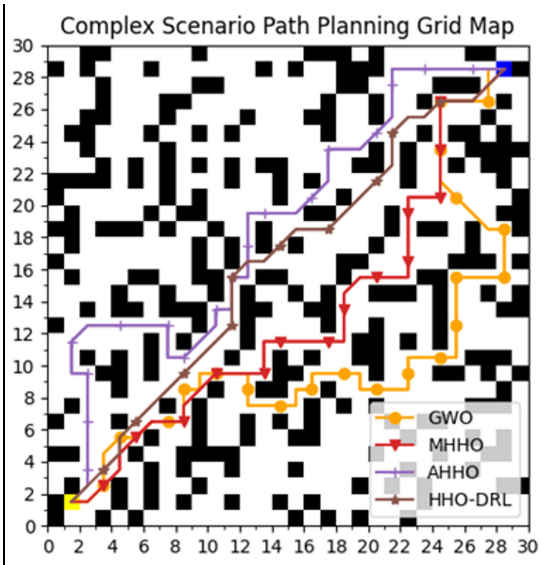


Fig. 4 Comparison of vehicle dynamic path planning using different algorithms in the complex scenario

Additionally, the study introduces the efficiency ratio to quantify the improvement multiplier of the HHO-DRL algorithm's optimization effect compared to other algorithms, providing a more comprehensive assessment of its performance. The calculation formula is as follows:

$$E_n = \frac{\mu_o}{\mu_m} \tag{44}$$

This formula indicates how many times better the improved performance is compared to the original performance. If $E_n > 1$, it signifies that the HHO-DRL algorithm outperforms the comparison algorithm in terms of path length.

As shown in Table 5, in the simple scenario, all algorithms successfully planned feasible paths. However, the HHO-DRL algorithm achieved the shortest path length (39.9411), outperforming GWO, MHHO, and AHHO by 26.04 %, 19.40 %, and 18.03 %, respectively. This result highlights the significant advantage of HHO-DRL in fine-tuning path optimization, effectively reducing travel distance and enhancing both the economic efficiency and safety of path planning.

Additionally, the efficiency ratio data further reinforces this conclusion. The efficiency ratios of HHO-DRL relative to GWO, MHHO, and AHHO are all greater than 1, specifically 1.35, 1.24, and 1.22, respectively. This indicates that HHO-DRL significantly improves path planning efficiency in simple environments.

In the complex scenario, where the obstacle ratio increases to 30%, the demands on path planning algorithms become significantly higher. The HHO-DRL algorithm successfully plans the shortest path (41.6984), outperforming GWO, MHHO, and AHHO by 35.34 %, 19.16 %, and 24.61 %, respectively. Notably, in high-obstacle environments, HHO-DRL achieves an optimization rate of 35.34 %, significantly surpassing the performance of other algorithms.

Table 5 Performance comparison of different algorithms in vehicle dynamic path planning

Environment	Algorithm	Path Length	g_n (%)	E_n (Multiplier)
Simple	GWO	54.0	26.04	1.35
	MHHO	49.5563	19.40	1.24
	AHHO	48.7279	18.03	1.22
	HHO-DRL	39.9411	/	/
Complex	GWO	64.4852	35.34	1.55
	MHHO	51.5785	19.16	1.23
	AHHO	55.3137	24.61	1.33
	HHO-DRL	41.6984	/	/

Additionally, the efficiency ratio results reveal that HHO-DRL achieves values greater than 1 in the complex scenario, specifically 1.55, 1.23, and 1.33 relative to GWO, MHHO, and AHHO, respectively. This further validates its superior performance and effectiveness in high-complexity environments.

The outstanding performance of the HHO-DRL algorithm in vehicle dynamic path planning can be primarily attributed to its core innovative mechanisms: the dynamic weight fusion and bidirectional feedback strategies. The specific advantages of these mechanisms are reflected in the following aspects:

Dynamic Weight Fusion Mechanism

- Adaptive Balance Between Global and Local Search: HHO-DRL dynamically adjusts the balance between HHO and DRL during the search process, enabling the algorithm to flexibly shift the proportion of global exploration and local refinement based on the current search state. In the initial phase of path planning, the algorithm emphasizes DRL's global exploration capabilities to efficiently cover a wide search space. As it nears the optimal path, the algorithm dynamically increases HHO's local search intensity to fine-tune and further optimize the path.
- Avoidance of Local Optima Traps: Dynamic weight adjustment allows the algorithm to intensify global exploration when it becomes trapped in local optima, enabling it to escape these traps and enhance overall optimization performance.

Bidirectional Feedback Mechanism

- Experience Sharing and Policy Updating: Through the bidirectional feedback mechanism, HHO-DRL integrates the high-quality solution sequences identified by HHO into DRL's experience pool with weighted importance, accelerating DRL's policy updates. Simultaneously, the high-quality solutions discovered by DRL are fed back to HHO, directing the next search phase toward more promising regions. This efficient information exchange fosters the co-evolution of both components, significantly enhancing the algorithm's adaptability and optimization efficiency.
- Co-evolution and Information Loop: The bidirectional feedback mechanism creates a continuous information loop, ensuring seamless collaboration between HHO and DRL throughout the optimization process. This synergy enhances the algorithm's adaptability and significantly improves optimization precision in dynamic environments.

5. Conclusion

The bidirectional feedback mechanism creates a continuous information loop, ensuring seamless collaboration between HHO and DRL throughout the optimization process. This synergy enhances the algorithm's adaptability and significantly improves optimization precision in dynamic environments.

Implementation Challenges. While HHO-DRL shows promising performance in simulation and benchmark tests, several practical challenges must be addressed for deployment in real-world autonomous vehicle systems. First, the hybrid framework's computational load—stemming from serial HHO population updates with complexity $O(ND)$ and DRL policy training with complexity $O(BP)$—can strain on-board processors and hamper real-time responsiveness. Second, bridging the "sim-to-real" gap requires robust handling of unmodeled dynamics, sensor noise, and decision latency; the algorithm must tolerate discrepancies between simulated state transitions and actual vehicle behavior. Third, memory management for the DRL experience buffer and HHO population data becomes critical under continuous operation, demanding efficient sample curation and pruning strategies. Finally, safety-critical applications impose stringent verification and validation procedures—any instability due to parameter misconfiguration or unexpected environment changes could lead to unsafe maneuvers. Addressing these deployment issues will be essential to translate HHO-DRL's theoretical advantages into dependable, real-time autonomous driving solutions.

The dynamic weight fusion mechanism dynamically adjusts the contributions of HHO and DRL in real-time during the search process based on fitness improvement rates and population diversity. This enables the algorithm to achieve an adaptive balance between global exploration and

local exploitation, significantly enhancing its adaptability and optimization efficiency in high-dimensional, multimodal, and nonlinear problems.

Moreover, the bidirectional feedback mechanism fosters co-evolution between HHO and DRL, enabling deep collaboration throughout the optimization process. The high-quality solutions identified by DRL during global exploration are fed back to HHO, guiding subsequent iterations to focus on promising regions. Simultaneously, the high-quality solution sequences discovered by HHO are incorporated into DRL's experience pool with weighted importance, accelerating DRL's policy updates. This bidirectional information exchange not only enhances the algorithm's overall optimization performance but also improves its robustness and stability in dynamic environments.

Experimental results demonstrate that HHO-DRL exhibits exceptional optimization capabilities across various standard test functions, including international benchmark functions and CEC-2014 complex functions, as well as in practical vehicle path planning tasks. Compared to standalone HHO, DRL, and other classical intelligent optimization algorithms (e.g., PSO, GWO, WOA, MHHO, AHHO), HHO-DRL shows significant advantages in avoiding local optima, enhancing search efficiency, and adapting to dynamic environments. Particularly in complex multimodal and high-dimensional nonlinear problems, HHO-DRL leverages dynamic weight fusion and bidirectional feedback mechanisms to achieve superior global exploration and local refinement, significantly improving the economic efficiency and safety of path planning.

Although the HHO-DRL algorithm has demonstrated significant performance improvements in the current study, several directions warrant further exploration and optimization: Multi-objective Optimization Extension: The current algorithm primarily focuses on optimizing single-objective functions. Future work could extend HHO-DRL to address multi-objective optimization problems, balancing multiple objectives such as path length, energy consumption, and safety to better meet the demands of more complex real-world applications.

Real-time and computational efficiency optimization: Real-time performance is critical for path planning in high-dimensional and dynamic environments. Future research could focus on improving the computational efficiency of HHO-DRL through techniques such as parallel computing, distributed optimization, and other acceleration strategies to reduce runtime and satisfy real-time path planning requirements.

Adaptive parameter adjustment: While the dynamic weight fusion mechanism provides a degree of adaptive regulation, further optimization of the parameter adjustment strategies—such as integrating adaptive learning rates or automated parameter tuning methods—could improve the algorithm's generalization capability and adaptability across diverse environments.

Future studies will aim to broaden its application scope, refine its algorithmic structure, and explore its theoretical properties in greater depth, further advancing the development and application of intelligent optimization algorithms to tackle increasingly complex real-world problems.

Acknowledgement

This project is supported by [funding of Visual Computing and Virtual Reality Key Laboratory of Sichuan Province] under Grant [SCVCVR2024.09VS]. We appreciate their support very much.

References

[1] Zhu, L. (2024). Traditional and advanced technologies in intelligent transportation systems, *Highlights in Science, Engineering and Technology*, Vol. 83, 696-702, doi: 10.54097/fv1m0f60.

[2] Dui, H., Zhang, S., Liu, M., Dong, X., Bai, G. (2024). IoT-enabled real-time traffic monitoring and control management for intelligent transportation systems, *IEEE Internet of Things Journal*, Vol. 11, No. 9, 15842-15854, doi: 10.1109/JIOT.2024.3351908.

[3] Dai, P., Xu, J. (2019). Establishing a network planning model of urban-rural logistics based on hub-and-spoke network, *Tehnički Vjesnik – Technical Gazette*, Vol. 26, No. 5, 1383-1391, doi: 10.17559/TV-20190704095301.

[4] Wang, D.L., Ding, A., Chen, G.L., Zhang, L. (2023). A combined genetic algorithm and A* search algorithm for the electric vehicle routing problem with time windows, *Advances in Production Engineering & Management*, Vol. 18, No. 4, 403-416, doi: 10.14743/apem2023.4.481.

[5] Ezugwu, A.E., Shukla, A.K., Nath, R., Akinyelu, A.A., Agushaka, J.O., Chiroma, H., Muhuri, P.K. (2021). Metaheuristics: A comprehensive overview and classification along with bibliometric analysis, *Artificial Intelligence Review*, Vol. 54, 4237-4316, doi: 10.1007/s10462-020-09952-0.
[6] Alabool, H.M., Alarabiat, D., Abualigah, L., Heidari, A.A. (2021). Harris hawks optimization: A comprehensive review of recent variants and applications, *Neural Computing and Applications*, Vol. 33, 8939-8980, doi: 10.1007/s00521-021-05720-5.
[7] Sun, H., Zhao, S. (2021). Fault diagnosis for bearing based on 1DCNN and LSTM, *Shock and Vibration*, Article No. 1221462, doi: 10.1155/2021/1221462.
[8] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. (2019). Harris hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, Vol. 97, 849-872, doi: 10.1016/j.future.2019.02.028.
[9] Piao, L. (2024). Remora optimization algorithm-based adaptive fusion via ant colony optimization for traveling salesman problem, *Computer Science and Information Systems*, Vol. 21, No. 4, 1651-1672, doi: 10.2298/CSIS240314052P.
[10] Ouyang, C., Liao, C., Zhu, D., Zheng, Y., Zhou, C., Li, T. (2024). Integrated improved Harris hawks optimization for global and engineering optimization, *Scientific Reports*, Vol. 14, No. 1, Article No. 7445, doi: 10.1038/s41598-024-58029-3.
[11] Zhong, C., Li, G., Meng, Z., He, W. (2023). Opposition-based learning equilibrium optimizer with Lévy flight for high-dimensional global optimization problems, *Expert Systems with Applications*, Vol. 215, Article No. 119303, doi: 10.1016/j.eswa.2022.119303.
[12] Zhou, P., Li, N., Huang, J., Zhang, S., Zhou, X., Liu, G. (2024). Deep reinforcement learning-based decision making for six degree of freedom UCAV close range air combat, In: Fu, S. (ed.), *2023 Asia-Pacific International Symposium on Aerospace Technology (APISAT 2023) Proceedings*, APISAT 2023, *Lecture Notes in Electrical Engineering*, Vol. 1051, Springer, Singapore, 320-334, doi: 10.1007/978-981-97-4010-9_24.
[13] Ai, T., Huang, L., Song, R.J., Huang, H.F., Jiao, F., Ma, W.G. (2023). An improved deep reinforcement learning approach: A case study for optimisation of berth and yard scheduling for bulk cargo terminal, *Advances in Production Engineering & Management*, Vol. 18, No. 3, 303-316, doi: 10.14743/apem2023.3.474.
[14] Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., Dai, B., Miao, Q. (2024). Deep reinforcement learning: A survey, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 4, 5064-5078, doi: 10.1109/TNNLS.2022.3207346.
[15] Wang, G., Wu, F., Zhang, X., Guo, N., Zheng, Z. (2024). Adaptive trajectory-constrained exploration strategy for deep reinforcement learning, *Knowledge-Based Systems*, Vol. 285, Article No. 111334, doi: 10.1016/j.knosys.2023.111334.
[16] Huo, L., Mao, J., San, H., Zhang, S., Li, R., Fu, L. (2024). Efficient and stable deep reinforcement learning: Selective priority timing entropy, *Applied Intelligence*, Vol. 54, 10224-10241, doi: 10.1007/s10489-024-05705-6.
[17] De Almeida, V.N., Alegre, L.N., Bazzan, A.L.C. (2024). Knowledge transfer in multi-objective multi-agent reinforcement learning via generalized policy improvement, *Computer Science and Information Systems*, Vol. 21, No. 1, 335-362, doi: 10.2298/CSIS221210071A.
[18] Dong, C., Li, D. (2024). Adaptive evolutionary reinforcement learning with policy direction, *Neural Processing Letters*, Vol. 56, Article No. 69, doi: 10.1007/s11063-024-11548-6.
[19] Majid, A.Y., Saaybi, S., Francois-Lavet, V., Venkatesha Prasad, R., Verhoeven, C. (2024). Deep reinforcement learning versus evolution strategies: A comparative survey, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 9, 11939-11957, doi: 10.1109/TNNLS.2023.3264540.
[20] Yu, Y., Zhang, P., Zhao, K., Zheng, Y., Hao, J. (2023). Accelerating deep reinforcement learning via knowledge-guided policy network, *Autonomous Agents and Multi-Agent Systems*, Vol. 37, Article No. 17, doi: 10.1007/s10458-023-09600-1.
[21] Alweshah, M., Almiani, M., Almansour, N., Al Khalaileh, S., Aldabbas, H., Alomoush, W., Alshareef, A. (2022). Vehicle routing problems based on Harris Hawks optimization, *Journal of Big Data*, Vol. 9, Article No. 42, doi: 10.1186/s40537-022-00593-4.
[22] Xie, B., Wang, L., Li, H., Huo, H., Cui, C., Sun, B., Ma, Y., Wang, J., Yin, G., Zuo, P. (2021). An interface-reinforced rhombohedral Prussian blue analogue in semi-solid state electrolyte for sodium-ion battery, *Energy Storage Materials*, Vol. 36, 99-107, doi: 10.1016/j.ensm.2020.12.008.
[23] Wang, J., Zhang, Y., Hu, Y., Yin, B. (2024). Large-scale traffic prediction with hierarchical hypergraph message passing networks, *IEEE Transactions on Computational Social Systems*, Vol. 11, No. 6, 7103-7113, doi: 10.1109/TCSS.2024.3419008.
[24] Hou, D.N., Liu, S.C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 315-332, doi: 10.14743/apem2024.3.509.
[25] Pu, Z., Wang, H., Liu, Z., Yi, J., Wu, S. (2023). Attention enhanced reinforcement learning for multi agent cooperation, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 34, No. 11, 8235-8249, doi: 10.1109/TNNLS.2022.3146858.
[26] Wang, C., Chen, X., Li, C., Song, R., Li, Y., Meng, M.Q.-H. (2023). Chase and track: Toward safe and smooth trajectory planning for robotic navigation in dynamic environments, *IEEE Transactions on Industrial Electronics*, Vol. 70, No. 1, 604-613, doi: 10.1109/TIE.2022.3148753.
[27] Huang, F., Cao, M., Wang, L. (2024). Approximate policy iteration for robust stochastic control of multi-agent Markov decision processes, *IEEE Transactions on Automatic Control*, Vol. 70, No. 6, 3587-3602, doi: 10.1109/TAC.2024.3510596.

[28] Wen, X., Yu, X., Yang, R., Chen, H., Bai, C., Wang, Z. (2024). Towards robust offline-to-online reinforcement learning via uncertainty and smoothness, *Journal of Artificial Intelligence Research*, Vol. 81, 481-509, doi: 10.1613/jair.1.16457.
[29] Iancu, D.-T., Florea, A.-M. (2023). An improved vehicle trajectory prediction model based on video generation, *Studies in Informatics and Control*, Vol. 32, No. 1, 25-36, doi: 10.24846/v32i1y202303.
[30] Ouyang, C., Liao, C., Zhu, D., Zheng, Y., Zhou, C., Zou, C. (2024). Compound improved Harris hawks optimization for global and engineering optimization, *Cluster Computing*, Vol. 27, 9509-9568, doi: 10.1007/s10586-024-04348-z.
[31] Fayti, M., Mjahed, M., Ayad, H., El Kari, A. (2023). Recent metaheuristic-based optimization for system modeling and PID controllers tuning, *Studies in Informatics and Control*, Vol. 32, No. 1, 57-67, doi: 10.24846/v32i1y202306.
[32] Gezici, H., Livatyali, H. (2022). An improved Harris Hawks Optimization algorithm for continuous and discrete optimization problems, *Engineering Applications of Artificial Intelligence*, Vol. 113, Article No. 104952, doi: 10.1016/j.engappai.2022.104952.
[33] Banerjee, C., Chen, Z., Noman, N., Zamani, M. (2022). Optimal actor-critic policy with optimized training datasets, *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 6, No. 6, 1324-1334, doi: 10.1109/TETCI.2022.3140375.
[34] Pan, W., Sun, Z., Sang, H., Wang, Z. (2023). Encrypted data learning and prediction using a BFV-based cryptographic convolutional neural network, *Studies in Informatics and Control*, Vol. 32, No. 1, 37-48, doi: 10.24846/v32i1y202304.
[35] Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S. (2013). CHOMP: Covariant Hamiltonian optimization for motion planning, *The International Journal of Robotics Research*, Vol. 32, No. 9-10, 1164-1193, doi: 10.1177/0278364913488805.
[36] Magyar, B., Tsiogkas, N., Brito, B., Patel, M., Lane, D., Wang, S. (2019). Guided stochastic optimization for motion planning, *Frontiers in Robotics and AI*, Vol. 6, Article No. 105, doi: 10.3389/frobt.2019.00105.
[37] Gobbi, H.U., Santos, G.D.D., Bazzan, A.L.C. (2024). Comparing reinforcement learning algorithms for a trip building task: A multi-objective approach using non-local information, *Computer Science and Information Systems*, Vol. 21, No. 1, 291-308, doi: 10.2298/CSIS221210072G.

Calendar of events

- 18th European Congress and Exhibition on Advanced Materials and Processes (EUROMAT 2025), 14-18 September 2025, Granada, Spain.
- ASTM International Conference on Advanced Manufacturing (ICAM 2025), 6-10 October 2025, Westgate Las Vegas Resort & Casino, Las Vegas, USA.
- 52nd International Conference on Computers and Industrial Engineering (CIE52), 29-31 October 2025, INSA Lyon, Lyon, France.
- International Conference on Industry of the Future & Smart Manufacturing (ISM 2025), 12-14 November 2025, Valletta, Malta.
- 17th EAI International Conference on Simulation Tools and Techniques (SIMUTools), 1-2 December 2025, Valencia, Spain.
- 21st International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2025), 7-10 December 2025, Seattle, WA, USA.
- 7th Industrial Engineering and Industrial Management Conference (IEIM 2026), 7-9 January 2026, Milan, Italy.
- 9th International Conference on Intelligent Manufacturing and Automation (IMA 2026), 16-18 January 2026, Nanjing, China.
- 10th International Conference on Robotics, Control and Automation (ICRCA 2026), 26-28 March 2026, Osaka, Japan.

This page intentionally left blank.

Notes for contributors

General

Articles submitted to the *APEM journal* should be original and unpublished contributions and should not be under consideration for any other publication at the same time. Manuscript should be written in English. Responsibility for the contents of the paper rests upon the authors and not upon the editors or the publisher. The content from published paper in the *APEM journal* may be used under the terms of the Creative Commons Attribution 4.0 International Licence (CC BY 4.0). For most up-to-date information please see the APEM journal homepage *apem-journal.org*.

Submission of papers

A submission must include the corresponding author's complete name, affiliation, address, phone and fax numbers, and e-mail address. All papers for consideration by *Advances in Production Engineering & Management* should be submitted by e-mail to the journal Editor-in-Chief:

Miran Brezocnik, Editor-in-Chief
UNIVERSITY OF MARIBOR
Faculty of Mechanical Engineering
Chair of Production Engineering
Smetanova ulica 17, SI – 2000 Maribor
Slovenia, European Union
E-mail: apem@apem-journal.org

Manuscript preparation

Manuscript should be prepared in *Microsoft Word 2010* (or higher version) word processor. *Word .docx* format is required. Papers on A4 format, single-spaced, typed in one column, using body text font size of 11 pt, should not exceed 12 pages, including abstract, keywords, body text, figures, tables, acknowledgements (if any), references, and appendices (if any). The title of the paper, authors' names, affiliations and headings of the body text should be in *Calibri* font. Body text, figures and tables captions have to be written in *Cambria* font. Mathematical equations and expressions must be set in *Microsoft Word Equation Editor* and written in *Cambria Math* font. For detail instructions on manuscript preparation please see instruction for authors in the *APEM journal* homepage *apem-journal.org*.

The review process

Every manuscript submitted for possible publication in the *APEM journal* is first briefly reviewed by the editor for general suitability for the journal. Notification of successful submission is sent. After initial screening, and checking by a special plagiarism detection tool, the manuscript is passed on to at least two referees. A double-blind peer review process ensures the content's validity and relevance. Optionally, authors are invited to suggest up to three well-respected experts in the field discussed in the article who might act as reviewers. The review process can take up to eight weeks on average. Based on the comments of the referees, the editor will take a decision about the paper. The following decisions can be made: accepting the paper, reconsidering the paper after changes, or rejecting the paper. Accepted papers may not be offered elsewhere for publication. The editor may, in some circumstances, vary this process at his discretion.

Proofs

Proofs will be sent to the corresponding author and should be returned within 3 days of receipt. Corrections should be restricted to typesetting errors and minor changes.

Offprints

An e-offprint, i.e., a PDF version of the published article, will be sent by e-mail to the corresponding author. Additionally, one complete copy of the journal will be sent free of charge to the corresponding author of the published article.

APEM journal

Advances in Production Engineering & Management

Chair of Production Engineering (CPE)
University of Maribor
APEM homepage: apem-journal.org

Volume 20 | Number 3 | September 2025 | pp 295-418

Contents

Published by CPE, University of Maribor

ISSN 1854-6250

apem-journal.org