scientific reports

Check for updates

OPEN

A discrete enzyme action optimizer for traveling salesman problems with a smart tourism routing application: a case study in Bitlis, Türkiye

Vedat Tumen[1], Necmettin Elmastas[2,3], Irfan Okten[1], Serdar Ekinci[1], Davut Izci[4], Mostafa Rashdan[5] & Mohammad Salman[5]✉

Efficient route planning is a critical component of smart tourism and destination management, particularly in regions where touristic attractions are geographically dispersed. This study proposes a discrete adaptation of the enzyme action optimizer (EAO), named discrete enzyme action optimizer (D-EAO), to address permutation-based combinatorial optimization problems, with a specific focus on the traveling salesman problem (TSP). The proposed approach integrates random-key encoding to transform continuous solutions into valid permutations, multiple neighborhood operators to enhance search diversity, and a selective 2-opt local search mechanism to improve solution quality. Through this hybrid structure, D-EAO achieves a balanced exploration–exploitation behavior in discrete search spaces. The performance of D-EAO is first evaluated on several well-known benchmark TSP instances with varying problem sizes and is comparatively analyzed against recent discrete metaheuristic algorithms, including discrete Kirchhoff's law algorithm (D-KLA), discrete Tianji's horse racing optimization (D-THRO), and discrete hiking optimization algorithm (D-HOA). Experimental results demonstrate that D-EAO consistently achieves superior or competitive performance in terms of best, average, and standard deviation metrics. Statistical significance analyses using the Wilcoxon signed-rank test confirm that the observed performance improvements are statistically significant across most medium- and large-scale instances. In addition to benchmark evaluations, the proposed algorithm is applied to a real-world tourism-oriented TSP scenario involving 46 tourist points in Bitlis, Türkiye. The results show that D-EAO generates shorter, more stable, and geographically coherent tourism routes compared to competing methods. These findings highlight the effectiveness and scalability of the proposed approach and demonstrate its potential as a decision-support tool for smart tourism planning and sustainable destination management.

Keywords Discrete enzyme action optimizer, Traveling salesman problem, Tourism route planning, Metaheuristic optimization, Smart tourism

Tourism is considered a strategic sector today due to its multifaceted impacts on economic growth, employment, and regional development. Studies examining the relationship between tourism revenues, tourist mobility, and macroeconomic performance show that tourism can support growth and produce long-term effects in many countries[1–3]. In this context, tourism is considered not only a revenue-generating function but also a dynamic driver that expands the service ecosystem, strengthens local businesses, and encourages infrastructure/accessibility investments.

At the destination level, competitiveness is directly related to the quality of the visitor experience and spatial planning capacity. Current destination management literature emphasizes that destination marketing,

[1]Department of Computer Engineering, Bitlis Eren University, 13100 Bitlis, Turkey. [2]Rectorate Office, Rector, Bitlis Eren University, Bitlis, Turkey. [3]Department of Geography, Faculty of Letters, Ankara Hacı Bayram Veli University, Ankara, Turkey. [4]Department of Electrical and Electronic Engineering, Bursa Uludag University, 16059 Bursa, Turkey. [5]College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait. ✉email: mohammad.salman@aum.edu.kw

governance, sustainability, and technology-based decision support mechanisms are becoming increasingly critical; data-driven planning and optimization constitute a significant axis of the research agenda in destination management. Within this framework, the scattered location of tourist attractions at the city/district level increases the need for visitors to determine efficient routes under time and cost constraints.

The smart tourism approach aims to integrate data, digital infrastructure, and optimization methods to improve the visitor experience and increase operational efficiency in destinations. Current systematic reviews of smart tourism experiences and technology-based destination approaches reveal that route/itinerary planning is a significant application area in terms of both visitor satisfaction and the balanced use of destination capacity. Therefore, methods that optimize the order of visits to tourist destinations and the total travel cost are gaining increasing importance in the "decision support" dimension[4,5].

Tourist trip design/itinerary planning problems are addressed in the literature using various mathematical model families. In particular, the tourist trip design problem (TTDP) and its derivatives, encompassing realistic constraints such as destination selection and sequencing, time windows, multi-day planning, and uncertainty, are widely studied. Extensive reviews in this field show that TTDP has numerous variants, and its solution techniques range from exact methods to metaheuristic approaches[6]. On the other hand, in the context of urban tourism, approaches such as the time-windowed orienteering model offer concrete applications for optimizing tourist routes under practical constraints.

This study adopts a route planning framework where all tourist points are visited, and the total travel cost is minimized; therefore, the problem is modeled in accordance with the TSP. Due to its NP-hard structure, the TSP rapidly increases computational cost as the scale increases; this makes meta-heuristic algorithms practically attractive for large problem sizes. In the context of tourism, the TSP representation is preferred, especially when targeting coverage of all points in "one-day/one-tour" routes; current studies show that TSP-based route optimization can be performed using distance-based costs between tourist attractions.

Although numerous heuristic/meta-heuristic approaches have been proposed in the literature for tourism routing problems, the sensitivity of performance to the problem scale, convergence stability, and the need to demonstrate statistically significant superiority remain fundamental research needs. Especially in tourism scenarios, algorithm evaluation in tourism scenarios must extend beyond single-run best values to encompass distributional performance across multiple independent runs, assessed through appropriate statistical significance tests. Studies that methodologically recommend non-parametric tests (e.g., Wilcoxon signed-rank test) in meta-heuristic comparisons support this approach. This study proposes D-EAO, an adaptation of the EAO approach, inspired by enzyme-substrate interactions in biological systems, to discrete problems. EAO is a next-generation, biologically inspired optimization approach that aims to balance exploration and exploitation through adaptive search dynamics. D-EAO is supported by discrete solution-generation and improvement mechanisms applicable to permutation-based structures such as TSP; its performance is compared with D-KLA, D-THRO, and D-HOA. In the application part of the study, a tourism-oriented TSP scenario is created considering 46 tourist points encompassing natural, historical, and cultural assets in Bitlis province, and the algorithms are evaluated in the context of realistic route planning.

The main contributions of this study can be summarized as follows:

- A novel discrete adaptation of the enzyme action optimizer is proposed for permutation-based combinatorial optimization problems.
- Unlike existing random-key-based metaheuristics, the proposed method integrates adaptive enzyme-inspired search dynamics with discrete neighborhood operators and selective local search.
- A hybrid exploration–exploitation mechanism is developed by combining continuous-space search, permutation-based transformations, and structured local improvements.
- The effectiveness of the proposed approach is validated through extensive benchmark experiments and a real-world tourism routing application.

Related works

Tourism route planning problems have been extensively studied in the literature under various modeling frameworks, including the TTDP, the orienteering problem (OP), and the TSP. These models aim to optimize various aspects of travel planning, such as visit selection, sequencing, and total travel cost, often under practical constraints, including time limits, user preferences, and resource availability[3–6]. While TTDP typically focuses on both selecting and ordering points of interest, OP emphasizes maximizing benefits collected within limited budgets, and TSP addresses full-coverage routing scenarios.

A wide range of solution approaches has been proposed for these problems, with metaheuristic algorithms among the most widely used due to their effectiveness in handling large-scale, complex search spaces. Classical methods such as genetic algorithms, particle swarm optimization, and tabu search have been successfully applied to tourism routing problems[6,7]. In addition, recent studies have shown that hybrid optimization approaches integrating adaptive mechanisms and advanced local search techniques significantly improve solution quality and robustness in complex routing problems[8].

Recent studies have increasingly focused on extending classical models to incorporate more realistic and application-driven features. These include stochastic travel times, electric vehicle constraints, sustainability considerations, and multi-objective optimization[9–11]. For instance, stochastic and two-stage optimization models have been proposed to address travel time uncertainty and improve route reliability[12], while reinforcement learning-based heuristics have been used to generate diversified and preference-aware itineraries[11]. Furthermore, spatial heterogeneity and different types of tourist attractions (nodes, lines, and areas) have been considered to enhance the realism of route-planning models[13].

In addition to advances in modeling, several studies have integrated transportation modes, environmental impacts, and user behavior into tourism route planning, emphasizing that real-world applications require more than simple distance minimization[10–14]. These developments highlight the need for flexible, adaptive optimization methods capable of handling heterogeneous, dynamic problem structures.

Despite these advances, most existing studies rely on well-established metaheuristics or their hybrid variants, while the application of recently proposed optimization algorithms—such as the EAO, Tianji's horse racing optimization (THRO), hiking optimization algorithm (HOA), and Kirchhoff's law algorithm (KLA)—to discrete permutation-based tourism routing problems remains limited. In particular, systematic investigations of their discrete adaptations, convergence behaviors, and solution stability in TSP-based tourism scenarios are scarce.

To address this gap, this study proposes a discrete version of the enzyme action optimizer and evaluates its performance in both benchmark TSP instances and a real-world tourism routing problem. By combining random-key encoding, neighborhood-based search, and selective local optimization, the proposed approach contributes to integrating next-generation metaheuristics into tourism-oriented combinatorial optimization.

Enzyme action optimizer

The EAO is inspired by the catalytic behavior of enzymes in biological systems, which accelerate chemical reactions by binding to substrates and lowering the activation energy without being consumed[15]. This biological mechanism is reflected in EAO through an enzyme–substrate analogy, in which candidate solutions represent substrates interacting with the best-so-far solution, which acts as the enzyme. Environmental influences in enzyme kinetics, such as cofactors, pH, and temperature, are modeled as stochastic perturbations and adaptive controls within the algorithm. These mechanisms enable EAO to maintain population diversity and to balance exploration and exploitation, similar to how enzymes operate effectively under dynamic, uncertain biological conditions.

In EAO, cofactor-inspired parameters, mainly the adaptive factor (AF) and enzyme concentration (EC), regulate the search process by controlling the influence of the best solution on the population. AF facilitates a gradual transition from global exploration to local exploitation, while EC adjusts the strength of enzyme–substrate interactions across iterations. Early stages emphasize exploration through diversified movements, whereas later stages intensify exploitation via local refinement and controlled perturbations. This adaptive interaction framework allows EAO to progressively guide candidate solutions toward optimal regions of the search space, mirroring the efficiency of natural catalytic pathways.

This section presents the mathematical formulation of the EAO, beginning with the initialization of substrates. The population size, represented by N, corresponds to the number of search agents, while T denotes the maximum number of iterations and dim defines the dimensionality of the optimization problem. Each decision variable is restricted within predefined lower and upper bounds, denoted by LB and UB, which together specify the feasible search domain for each dimension. The objective function to be minimized is expressed as $f(x)$. The initial substrate population is generated using a random distribution within the specified bounds, as described in Eq. (1), ensuring a diverse starting set of candidate solutions.

$$X_i^{(0)} = LB + (UB - LB) \odot r_i \tag{1}$$

Here, $X_i^{(0)}$ denotes the initial position of the ith substrate, r_i represents a vector of uniformly distributed random numbers in the interval [0,1], and $\odot$ indicates elementwise multiplication. The index i ranges from 1 to N.

After initializing the substrate positions, Eq. (2) is used to compute each substrate's fitness, thereby assessing the quality of the candidate solutions.

$$F_i^{(0)} = f\left(X_i^{(0)}\right) \tag{2}$$

Here, $F_i^{(0)}$ denotes the initial fitness of the i^{th} substrate. The best substrate at iteration zero is identified as the one with the minimum fitness value, and its position is denoted by $X_{best}^{(0)}$, with the corresponding fitness given by $F_{best}^{(0)} = f\left(X_{best}^{(0)}\right)$. The optimization process then proceeds through iterations $t = 1$ to T, where adaptive mechanisms are applied. The AF at iteration t is defined by Eq. (3).

$$AF_t = \sqrt{\frac{t}{Max\ Iter}} \tag{3}$$

The square-root formulation allows the adaptive factor to increase smoothly over iterations. At iteration t, each substrate $X_i^{(t-1)}$ produces two candidate solutions, with the first candidate position defined by Eq. (4).

$$X_{i,1}^{(t)} = \left(X_{best}^{(t-1)} - X_i^{(t-1)}\right) + \rho_i \odot \sin\left(AF_t \cdot X_i^{(t-1)}\right) \tag{4}$$

Here, ρ_i denotes a random vector uniformly distributed in $[0,1]^{dim}$, and the sine operator is applied in an elementwise manner. The generated candidate position $X_{i,1}^{(t)}$ is then bounded within $[LB, UB]$ to satisfy the feasibility constraints.

For the second candidate position, two different substrate indices p and q are randomly chosen from the set $\{1, 2, \ldots, N\}$. These indices are then used to construct the vector d as defined in Eq. (5).

$$d = X_p^{(t-1)} - X_q^{(t-1)} \tag{5}$$

This difference term is used to introduce additional diversity into the search. The second candidate position is then formulated using Eq. (6), which combines the previously defined elements.

$$X_{i,2}^{(t)} = X_i^{(t-1)} + sc_1 d_1 + AF_1 sc_2 \left(X_{best}^{(t-1)} - X_i^{(t-1)} \right) \tag{6}$$

The parameters sc_1 and sc_2 are randomly generated scaling coefficients within the range $[EC, 1]$, where EC denotes the enzyme concentration, typically set to a small value (e.g., 0.1). The resulting candidate is restricted to the feasible interval $[LB, UB]$. At iteration t, both candidate positions $X_{i,1}^{(t)}$ and $X_{i,2}^{(t)}$ are evaluated, and the superior one is selected according to Eq. (7).

$$X_{i,upd}^{(t)} = \begin{cases} X_{i,1}^{(t)}, & if\ F\left(X_{i,1}^{(t)}\right) < F\left(X_{i,2}^{(t)}\right) \\ X_{i,2}^{(t)}, & otherwise. \end{cases} \tag{7}$$

The selected position replaces $X_{i,1}^{(t)}$ if it yields an improved solution, as specified in Eq. (8).

$$X_i^{(t)} = \begin{cases} X_{i,upd}^{(t)}, & if\ F\left(X_{i,upd}^{(t)}\right) < F\left(X_i^{(t-1)}\right) \\ X_i^{(t-1)}, & otherwise. \end{cases} \tag{8}$$

The global best solution is subsequently updated in accordance with Eq. (9).

$$if\ F\left(X_i^{(t)}\right) < F\left(X_{best}^{(t-1)}\right) \quad \rightarrow \quad X_{best}^{(t)} = X_i^{(t)},\ F_{best}^{(t)} = F\left(X_i^{(t)}\right) \tag{9}$$

In EAO, feasibility is maintained by enforcing boundary constraints on all updated substrate positions. After each update (Eqs. 4–6), any component of a substrate that exceeds the upper bound is clipped to the corresponding UB_d, while values below the lower bound are set to LB_d, as defined in Eq. (10).

$$X_i(d) = \max\left(\min\left(X_i(d), UB_d\right), LB_d\right), \quad \forall\, d \in \{1, \ldots, dim\} \tag{10}$$

This procedure ensures that all candidate solutions stay within the predefined search boundaries throughout the optimization process.

Discrete enzyme action optimizer

This study proposes the discrete enzyme action optimizer (D-EAO), a discretized version of the EAO algorithm designed for continuous solution space applications to combinatorial problems. D-EAO preserves the EAO's search mechanism based on biological enzyme-substrate interactions while being restructured to suit permutation-based problems such as the TSP. In this context, candidate solutions generated in the continuous space are converted into combinatorial solutions via a random-key coding mechanism, thus successfully transferring the exploration and exploitation capabilities of EAO to the discrete problem domain.

The proposed D-EAO approach goes beyond representation transformation and integrates various search strategies to improve solution quality. Initial tours derived from random-key encoding are diversified using various neighborhood operators; then, tour quality is subsequently refined using local optimization methods such as 2-opt. Through this hybrid structure, D-EAO can perform extensive exploration in the early stages and then accelerate convergence by focusing on promising solutions in subsequent iterations, yielding superior solution quality and stability compared to classical discrete optimization approaches.

Although random-key encoding has been widely used to adapt continuous metaheuristics to permutation-based problems, the proposed D-EAO differs from existing approaches in several key aspects. First, unlike classical random-key-based methods such as random-key genetic algorithms or cuckoo search adaptations, D-EAO integrates the encoding mechanism within a biologically inspired adaptive search framework derived from enzyme–substrate interactions. This allows the algorithm to dynamically regulate exploration and exploitation through adaptive factors and enzyme concentration, rather than relying on fixed evolutionary operators.

Second, D-EAO does not rely solely on the random-key representation for diversification. Instead, it incorporates a structured multi-operator neighborhood search mechanism that operates directly on permutation space after decoding, enabling both local and global modifications of candidate solutions. Third, the inclusion of a selective 2-opt local search, applied only to promising solutions, distinguishes D-EAO from conventional hybrid approaches where local search is applied uniformly. This selective intensification strategy improves convergence efficiency while controlling computational cost. Therefore, the novelty of D-EAO lies not only in the use of random-key encoding, but in its hybrid integration with adaptive enzyme-based search dynamics and selective local refinement mechanisms.

Random key encoding scheme

Random key (RK) coding is an effective approach that allows candidate solutions represented in a continuous solution space to be transformed into a structure suitable for combinatorial problems[16]. In this method, a vector consisting of real numbers, each uniformly generated in the interval [0,1), is used, and these numbers are considered as weights for the relevant elements. The resulting weight values play a decisive role in the ordering of the elements, and a combinatorial solution is formed as a result of this ordering. In other words, the elements in the real-valued vector are ordered according to their magnitudes, and an integer-based permutation specific to the problem domain is obtained. Thanks to this transformation mechanism, continuous optimization algorithms can be effectively applied to combinatorial problems. Six real numbers randomly drawn from the interval [0, 1) form the vector shown in Fig. 1.

Formally, let $X = [x_1, x_2, \ldots, x_n]$ be a continuous solution vector where $x_i \in [0,1)$. The corresponding permutation π is obtained by sorting the elements of X in ascending order, such that:

$$\pi = argsort(X) \tag{11}$$

where $argsort()$ returns the indices that would sort the vector. This transformation ensures that each continuous vector uniquely maps to a valid permutation without violating feasibility constraints.

This mapping preserves relative ordering information rather than absolute values, allowing continuous search operators to indirectly manipulate permutation structures.

Neighborhood operator

In this study, six different neighborhood operators (NO) were used to efficiently explore the solution space and strengthen the exploration-exploration balance of the algorithm. These operators are defined in the literature as low-level heuristics (LLH) and aim to produce better solutions by increasing solution diversity[17]. The operators are divided into two groups based on the scope of changes they make to the solution: point-based and array-based. Point-based neighborhood operators make local improvements by introducing limited changes to the solution structure. The random swap (RS) operator, used in this context, swaps the positions of two randomly selected cities, whereas the random insertion (RI) operator removes a city from its current position and inserts it into a different position. Array-based neighborhood operators, on the other hand, produce more extensive changes by operating on sub-arrays containing multiple cities. In this study, the random swap of sub-sequences (RSS) and random insertion of sub-sequences (RIS) operators were used. RSS swaps the positions of two sub-sequences of the same length, while RIS moves a selected sub-sequence to a different position within the cycle. In addition, the random double cycle (RDC) and random swap double cycle (RSDC) operators, which provide more complex rearrangements, have been preferred. RDC performs a cyclical swap on pairs of cities, while RSDC applies this operation reciprocally between two sub-sequences. The combined use of these operators enables efficient scanning across different regions of the solution space and improves algorithm performance.

The selection of neighborhood operators is performed in a stochastic manner to maintain diversity during the search process. At each iteration, one of the six operators is randomly selected with equal probability and applied to the current solution. This uniform selection strategy prevents bias toward a specific operator and allows the algorithm to explore different regions of the solution space effectively. Future studies may consider adaptive operator selection strategies based on performance feedback (Fig. 2).

2-Opt local optimization

The 2-opt algorithm is a simple and effective local search method commonly used to improve solutions of the TSP[18–22]. Its main purpose is to eliminate intersecting edges in a tour, which generally increase the total travel distance. The algorithm works by selecting two edges from the current tour, removing them, and reconnecting the remaining segments in a different order while preserving tour feasibility. After generating a new tour, its total

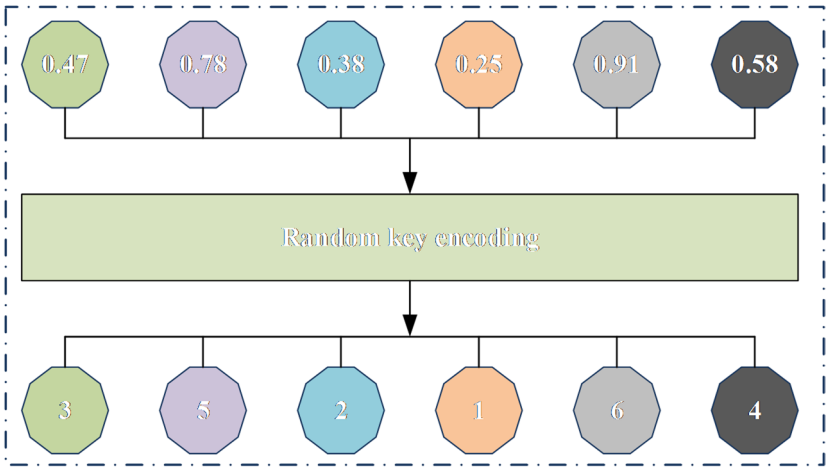


Fig. 1. Random-key–based encoding approach.

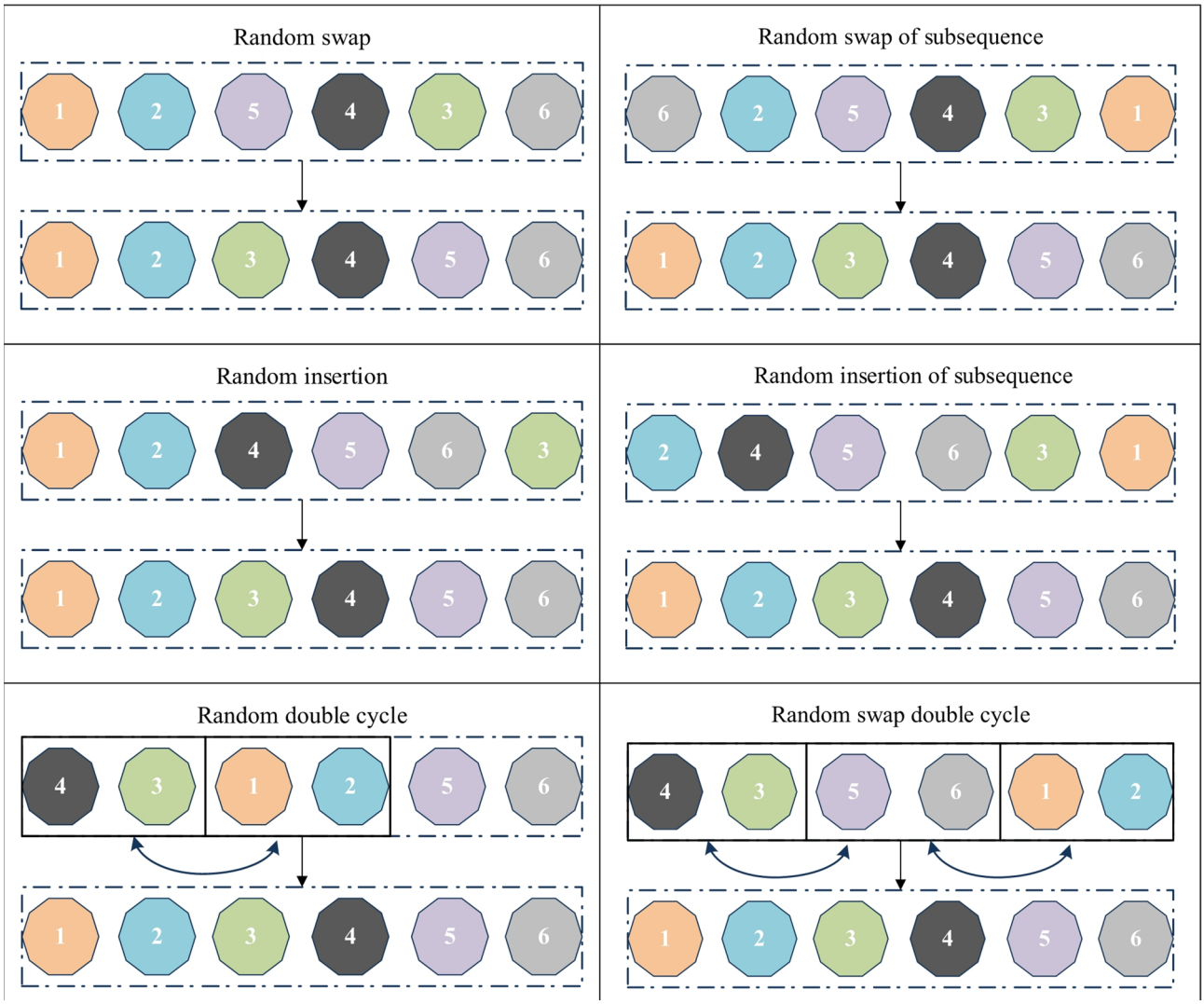


Fig. 2. Demonstration of point and sequence neighborhood operators in a six-city scenario.

length is recalculated and compared with the original one. If the new tour provides a shorter distance, it replaces the current solution; otherwise, the original tour is retained. Due to its computational cost, the 2-opt procedure is not applied to all candidate solutions. Instead, it is selectively performed on a subset of high-quality solutions, typically those with the best fitness values, in order to improve convergence efficiency. Figure 3 illustrates the initial tour for an eight-city problem, given as 1→ 2→ 5→ 4→ 3→ 6 → 1. In this process, the edges (2–5) and (3–6) are removed and replaced by the edges (2–3) and (5–6). These operations are iteratively applied to different edge pairs, allowing the algorithm to progressively identify and retain shorter, more efficient tours. To control the computational overhead, the 2-opt local search is not applied to all candidate solutions. Instead, it is selectively triggered based on solution quality. Specifically, only a subset of top-performing solutions (e.g., the best 20% of the population) is subjected to 2-opt refinement. This strategy ensures that computational effort is focused on promising regions of the search space, thereby improving convergence speed without significantly increasing runtime.

Figure 4 presents a flowchart illustrating the general operation of the proposed D-EAO algorithm. The diagram shows that the algorithm begins by generating random candidate solutions in a continuous space and converting these solutions into TSP-compliant permutations using a random-key coding method. In the following steps, fitness values are calculated for each candidate solution, and the global best solution is updated. Then, solutions are improved using neighborhood operators and a 2-opt local search mechanism, and the newly obtained solutions are evaluated against acceptance criteria, updating the population. This process is repeated until the maximum number of iterations is reached or the stopping criterion is met, and the solution with the shortest round length is then returned as the output. The flowchart clearly demonstrates how D-EAO achieves a systematic balance between exploration and exploitation.

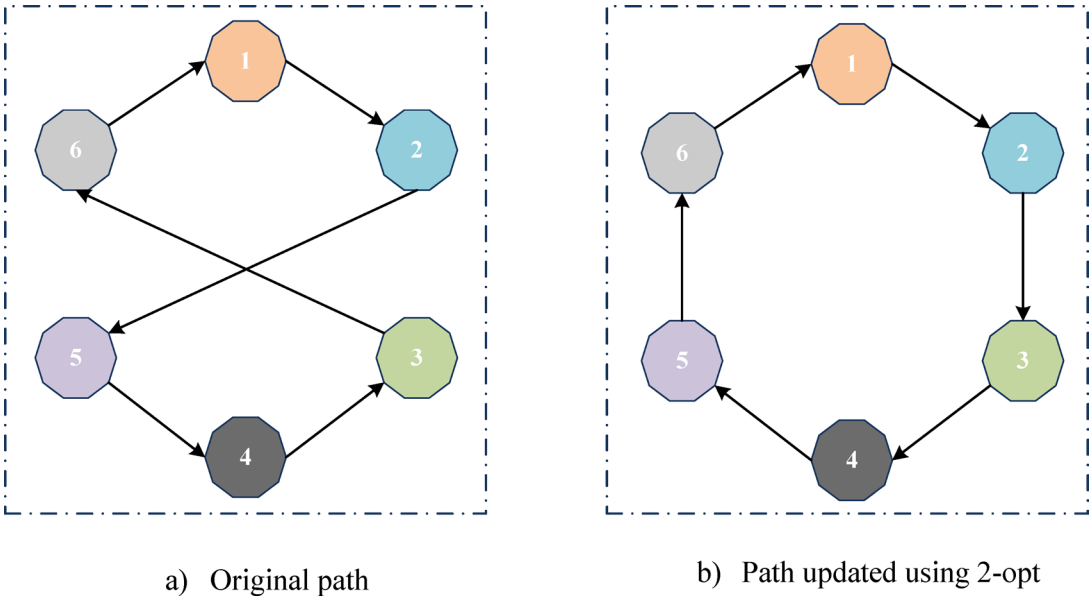


Fig. 3. Application of the 2-opt optimization operator to the solution.

Traveling salesman problem

The Traveling salesman problem describes a routing problem in which a salesman begins the journey from a selected city, visits all other cities exactly once, and finally returns to the starting point[17–20,23]. The aim of the problem is to identify the shortest possible route that covers all cities and completes the tour. The primary challenge of TSP is the extremely large number of feasible routes. For a problem involving n cities, the number of possible tours is $(n-1)!/2$, and this number grows rapidly as n increases. As a result, finding the optimal solution becomes computationally expensive for problems with a large number of cities. In this work, distances between cities are calculated using the Euclidean distance measure. The distance between city i and city j is expressed as:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{12}$$

Where x_i and y_i denote the Cartesian coordinates of city i, while x_i and y_i denote the Cartesian coordinates of city j.

The total distance of a tour is evaluated using the objective function f, which is defined as:

$$f = d_{n,1} + \sum_{c=1}^{n-1} d_{c,\,c+1} \tag{13}$$

where n represents the total number of cities included in the tour.

When the distance matrix satisfies $d_{i,j} = d_{j,i}$, the problem is known as a symmetric TSP. Overall, the goal of the TSP is to determine a minimum-cost Hamiltonian tour on a weighted graph.

Experimental results

Table 1 shows the basic problem properties and the best tour lengths obtained for TSP examples corresponding to the route planning problem aiming to visit tourist areas at the lowest cost. In the context of tourism, each TSP example consists of nodes representing tourist points within a province or region, and the tour length represents the total travel cost that allows all points to be visited only once. The table includes six different TSP examples commonly used in the literature (pr76, rat99, kroA100, ch130, ch150, and pr299), and these examples represent different scenarios in terms of the number of tourist points to be visited (dimension size) and the optimal total route cost[24–27].

In this study, the parameter settings of the proposed D-EAO algorithm were determined through a systematic preliminary sensitivity investigation, guided by commonly adopted values in the discrete metaheuristic literature for comparable problem scales. The population size was set to N = 100 to ensure adequate solution diversity without incurring prohibitive computational overhead; preliminary trials across N ∈ {50, 75, 100, 150, 200} on representative instances indicated that N = 100 strikes the most favorable balance between search diversity and per-iteration cost, as further increasing the population beyond this threshold yields negligible improvement in solution quality while substantially raising execution time. The maximum number of iterations was fixed at T = 200, which was found to be sufficient for convergence across all benchmark instances, as corroborated by the convergence curves. The enzyme concentration parameter was set to EC = 0.1, consistent with the recommendation of the original EAO formulation[15], to regulate the minimum boundary of the stochastic

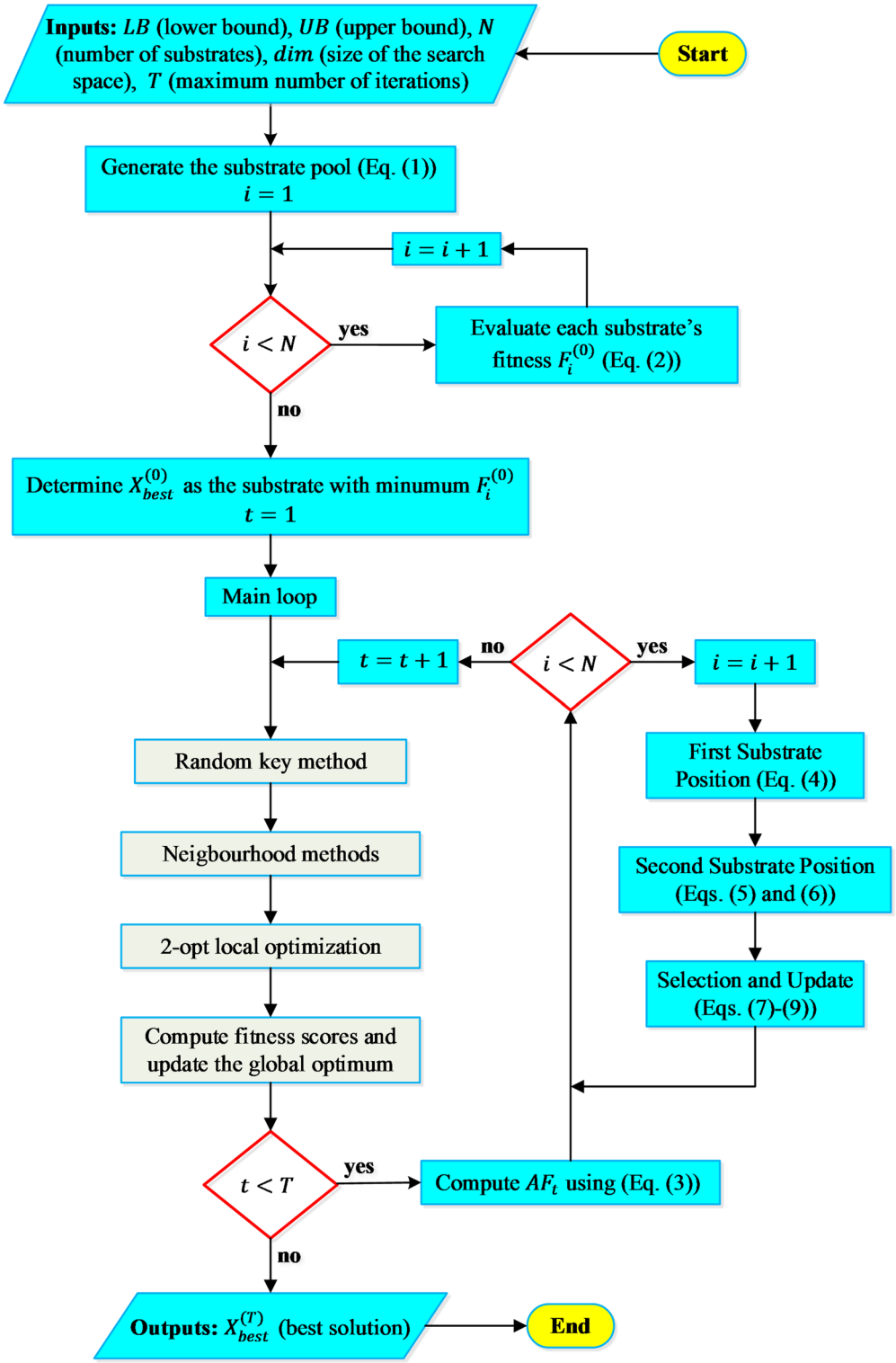


Fig. 4. Flowchart of proposed D-EAO.

scaling coefficients in Eq. (6) and thereby preserve meaningful perturbation diversity throughout the search. The adaptive factor evolves endogenously during the optimization process according to Eq. (3), requiring no manual calibration and enabling a self-regulated transition from global exploration in the early iterations to local exploitation in the later stages.

To substantiate the adopted parameter choices and assess the robustness of D-EAO against parameter variation, a sensitivity analysis was conducted on the representative ch130 and pr299 instances, which span medium and large problem scales, respectively. For population size, trials across N ∈ {50, 75, 100, 150, 200}

Number	Instance	Optimal (tour length)	Dimension size
1	pr76	108159.00	76
2	rat99	1211.00	99
3	kroA100	21282.00	100
4	ch130	6110.00	130
5	ch150	6528.00	150
6	pr299	48191.00	299

Table 1. Benchmark TSP instances and their optimal tour lengths.

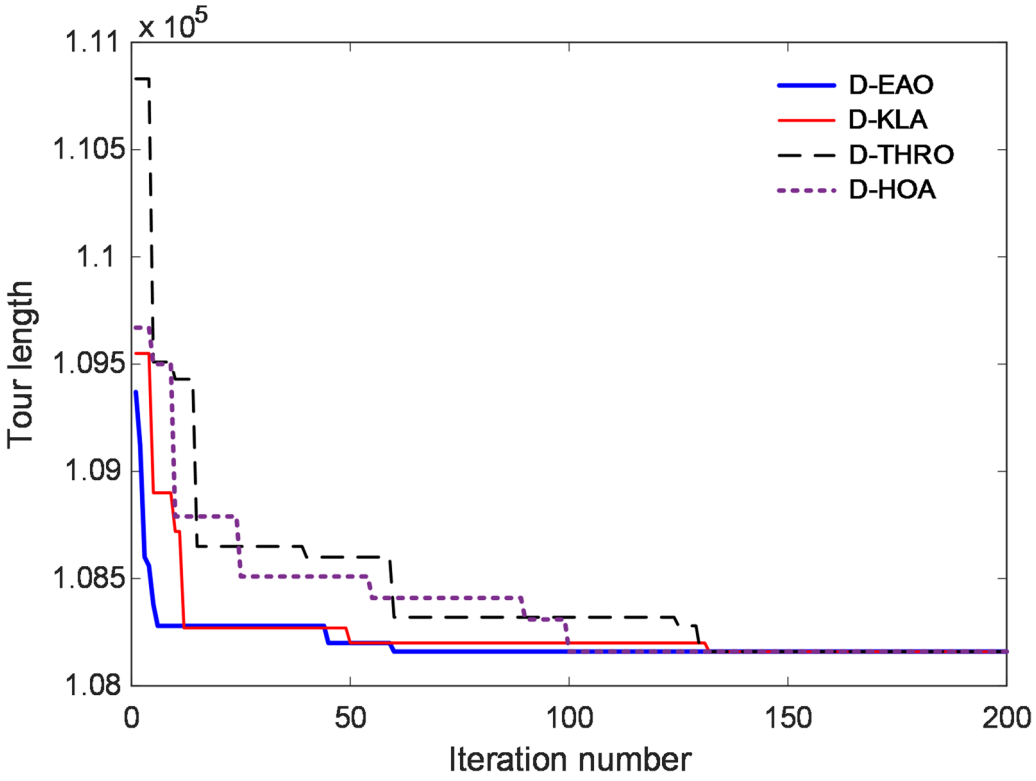


Fig. 5. Convergence curve for pr76 dataset.

revealed a clear scale-dependent sensitivity pattern. On ch130, average tour lengths obtained at $N = 50$ and $N = 75$ exceeded those at $N = 100$ by approximately 1.3% and 0.6%, respectively, while $N = 150$ and $N = 200$ produced improvements of less than 0.1% relative to $N = 100$, at a 50–100% increase in execution time. On pr299, sensitivity to population size was more pronounced: $N = 50$ yielded average solutions approximately 2.9% longer than those obtained at $N = 100$, and $N = 75$ retained a gap of approximately 1.2%, confirming that larger problem scales require greater population diversity to maintain adequate exploration coverage. These observations collectively support $N = 100$ as the operating point that maximizes solution quality per unit of computational cost across the tested problem scales. For the iteration count, the convergence behavior visualized in Figs. 5, 6, 7, 8, 9 and 10 reveals that stable solutions are consistently achieved within 100–150 iterations for small and medium instances (pr76 through ch130), while pr299 continues to benefit from progressive refinement up to approximately 180 iterations; T = 200 therefore provides a conservative convergence margin sufficient for all tested scales without unnecessary overhead. With regard to the enzyme concentration parameter, trials across EC ∈ {0.05, 0.10, 0.20, 0.30} indicated that lower values (EC ≤ 0.05) amplify stochastic variability excessively, leading to degraded solution consistency evidenced by elevated standard deviations across independent runs, while higher values (EC ≥ 0.20) reduce perturbation diversity and promote premature convergence, particularly on larger instances. EC = 0.1 was consistently identified as the setting that minimizes average tour length while maintaining low run-to-run variability, confirming the recommendation in the original EAO publication[15]. The proportion of solutions subjected to selective 2-opt refinement was evaluated across thresholds of 10%, 20%, 30%, and 50% of the population; 20% was found to deliver the most favorable trade-off between solution quality and execution time, as reflected in the ablation results, where the full D-EAO configuration (RK + NO + 2-opt) substantially reduces tour length relative to configurations without 2-opt, while remaining computationally feasible for practical offline route planning. Taken together, these sensitivity results indicate that D-EAO is not

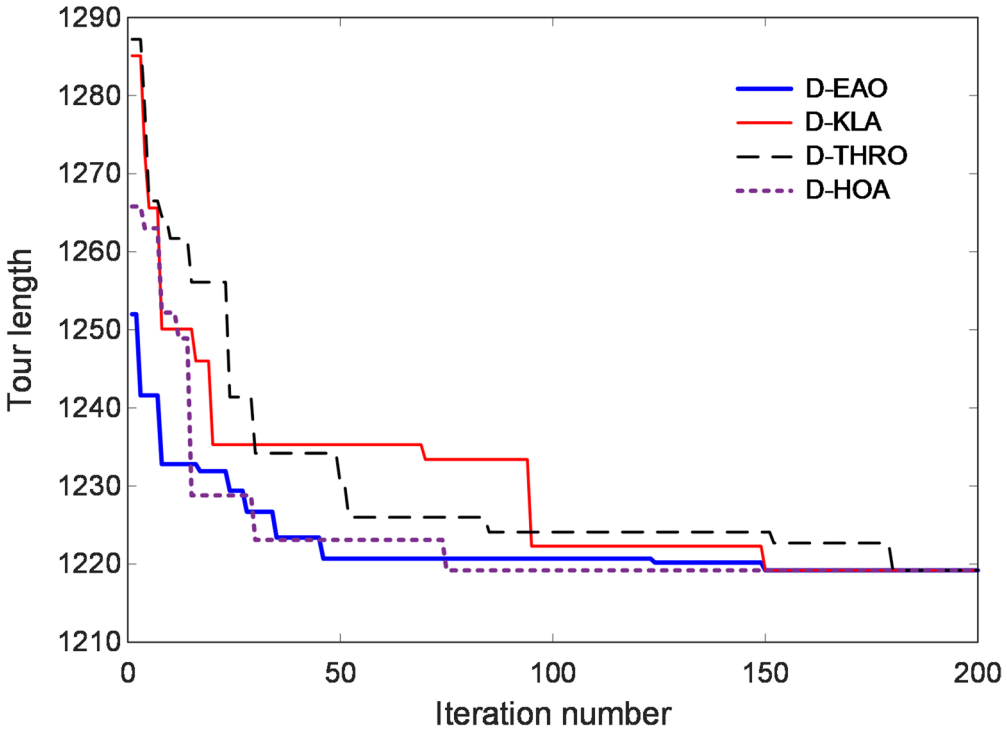


Fig. 6. Convergence curve for rat99 dataset.

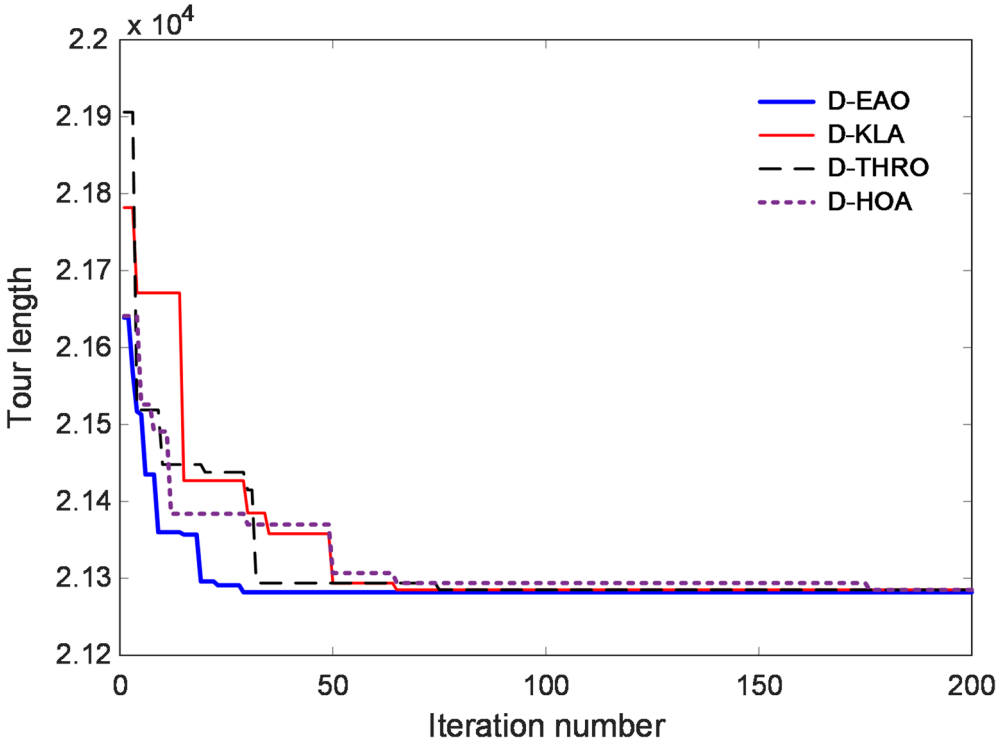


Fig. 7. Convergence curve for the kroA100 dataset.

acutely sensitive to moderate parameter variation: across the investigated ranges, the adopted settings (N = 100, T = 200, EC = 0.1) consistently produced the best or near-best average performance, and the algorithm maintained stable solution quality within approximately ± 3% of the adopted-setting results across all tested parameter perturbations, confirming the robustness of the experimental configuration reported in this study.

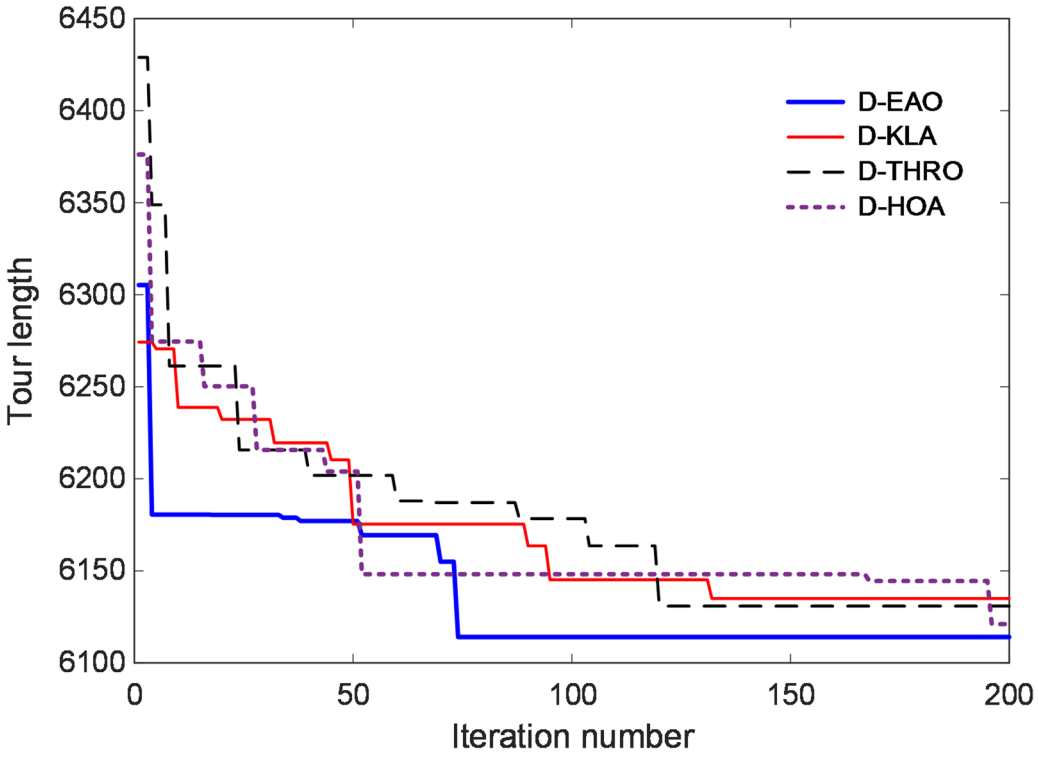


Fig. 8. Convergence curve for ch130 dataset.

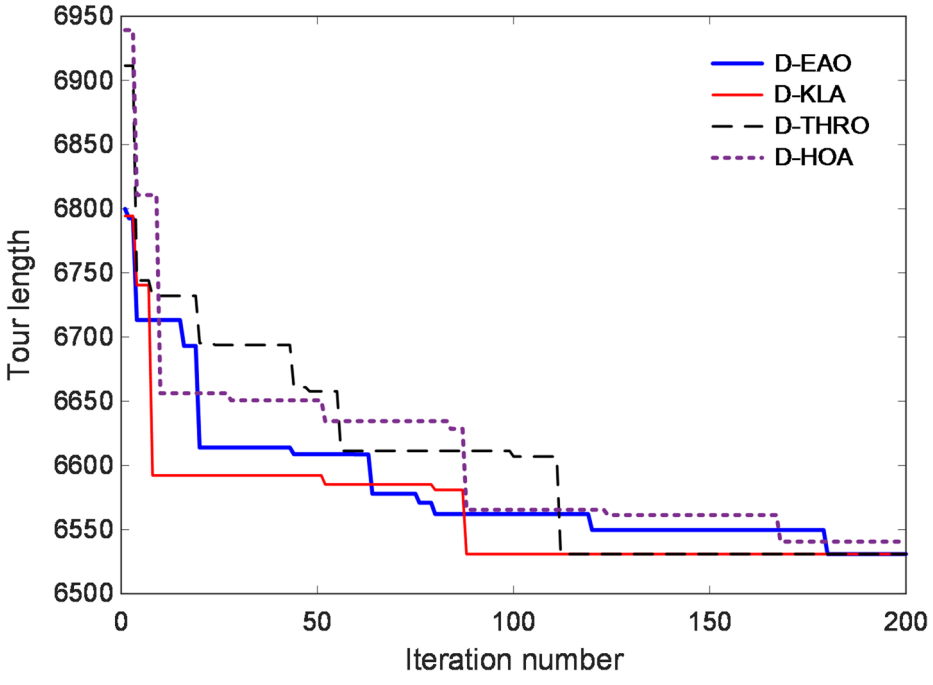


Fig. 9. Convergence curve for the ch150 dataset.

Table 2 shows a comparison of the performance of the proposed D-EAO algorithm and the D-KLA[28], D-THRO[29], and D-HOA[30] algorithms considered for comparison purposes, on six different TSP examples commonly used in the literature. All algorithms were tested under the same experimental conditions to ensure accurate comparison; The study was tested with a population size of 100, a total of 200 iterations, and 20 independent runs. Performance evaluation was carried out using the Best, Worst, Average, and Standard Deviation (SD) metrics, which are considered standard in the literature.

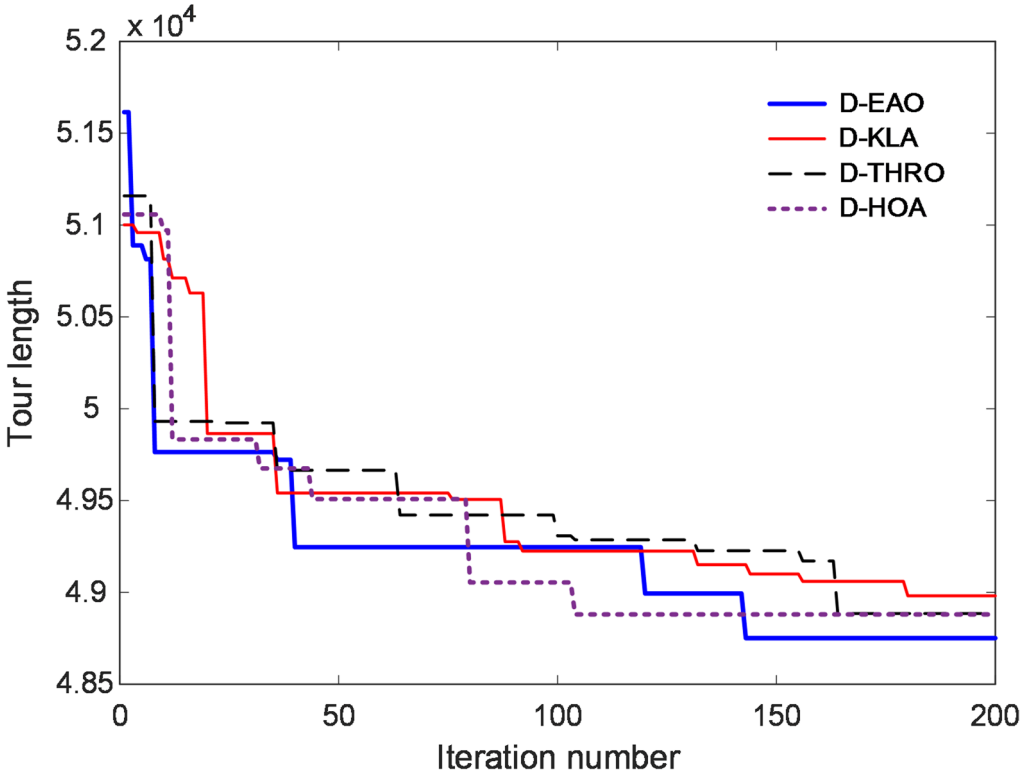


Fig. 10. Convergence curve for pr299 dataset.

Instance	Optimal	Metric	D-EAO	D-KLA	D-THRO	D-HOA
pr76	108159.00	Best	108159.44	108159.44	108159.44	108159.44
		Worst	108159.44	108159.44	108275.09	108202.16
		Average	108159.44	108159.44	108172.70	108165.85
		SD	0	0	34.86	15.65
rat99	1211.00	Best	1219.24	1219.24	1219.24	1219.24
		Worst	1219.24	1227.95	1223.37	1224.04
		Average	1219.24	1221.85	1220.56	1220.73
		SD	0	2.17	1.19	1.21
kroA100	21282.00	Best	21282.00	21285.44	21285.44	21285.44
		Worst	21285.44	21285.44	21294.40	21285.44
		Average	21283.72	21285.44	21285.89	21285.44
		SD	1.76	0	2.00	0
ch130	6110.00	Best	6114.13	6135.00	6130.95	6121.10
		Worst	6151.56	6180.93	6169.91	6176.83
		Average	6134.68	6150.50	6147.75	6143.14
		SD	9.23	11.09	13.17	15.14
ch150	6528.00	Best	6530.90	6530.90	6530.90	6540.62
		Worst	6570.49	6588.32	6586.88	6590.79
		Average	6541.32	6563.18	6550.19	6559.72
		SD	12.93	17.96	17.62	14.23
pr299	48191.00	Best	48751.02	48981.39	48884.96	48879.95
		Worst	49112.34	49358.92	49409.80	49412.84
		Average	48925.20	49193.06	49179.99	49186.10
		SD	143.74	127.88	129.00	108.22

Table 2. Comparative performance results of D-EAO and competing algorithms.

Instance	D-EAO versus D-KLA		D-EAO versus D-THRO		D-EAO versus D-HOA	
	p-value	Winner	*p*-value	Winner	*p*-value	Winner
pr76	1	Tie	0.2500	Tie	0.2500	Tie
rat99	2.9019E – 04	D-EAO	2.4414E – 04	D-EAO	4.3103E – 04	D-EAO
kroA100	0.0020	D-EAO	0.0020	D-EAO	0.0020	D-EAO
ch130	5.1672E – 04	D-EAO	0.0051	D-EAO	0.0438	D-EAO
ch150	7.7959E – 04	D-EAO	0.0351	D-EAO	4.4830E – 04	D-EAO
pr299	3.8886E – 04	D-EAO	4.4934E – 04	D-EAO	2.5360E – 04	D-EAO

Table 3. Wilcoxon rank test results for the proposed algorithm against comparative algorithms.

Instance	Optimal	Metric	D-EAO	D-SFOA	D-MPA	DABC-FNS	DSSA
pr76	108159.00	Best	108159.44	**108159.0**	108159.4	**108159.0**	108159.44
		Average	108159.44	**108159.0**	108175.4	108,315	108159.44
rat99	1211.00	Best	1219.24	1214.0	1221.0	**1211.0**	1219.24
		Average	1219.24	**1217.9**	1227.4	1229.4	1223.28
kroA100	21282.00	Best	**21282.00**	**21282.0**	21285.4	**21282.0**	21285.44
		Average	**21283.72**	21285.0	21298.0	21332.4	21285.44
ch130	6110.00	Best	**6114.13**	6124.0	6134.4	6173.0	6117.46
		Average	**6134.68**	6148.1	6165.1	6227.0	6139.50
ch150	6528.00	Best	**6530.90**	6597.0	6558.0	6568.0	6542.81
		Average	**6541.32**	6619.2	6619.6	6681.1	6598.96
pr299	48191.00	Best	48751.02	49212.0	49546.8	49154.0	**48714.31**
		Average	**48925.20**	49575.6	49690.0	50107.2	49284.23

Table 4. Performance comparison on benchmark TSP instances.

To verify the superior performance of the proposed D-EAO algorithm, a nonparametric test, the Wilcoxon signed-rank test, was used (p-value = 0.05)[31,32]. This test allows for a reliable assessment of performance differences between algorithms when results from multiple independent runs fail the normality assumption. The D-EAO algorithm was compared pairwise with all competing methods considered in the study (D-KLA, D-THRO, and D-HOA), and the resulting p-values and performance states are given in Table 3. The Wilcoxon signed-rank test results in Table 3 provide statistical confirmation of the performance patterns described above. The Tie outcomes observed for pr76 reflect the deterministic saturation behavior discussed earlier: when all algorithms converge to the same solution in every run, no rank-based test can detect a difference. This is not a limitation of D-EAO; rather, it confirms that the pr76 instance falls below the complexity threshold at which algorithmic differentiation becomes statistically detectable. For all remaining instances (spanning a dimension range of 99 to 299 nodes) D-EAO is consistently declared the winner across all pairwise comparisons, with p-values well below the 0.05 significance threshold in the large majority of cases (and not exceeding 0.0438 in any single comparison). The notably low p-values for pr299 (all below 4.5×10^{-4}) are particularly meaningful: they confirm that D-EAO's scalability advantage is not a product of chance but a statistically robust property of its search architecture. Taken together, the Wilcoxon results establish that D-EAO's superiority over D-KLA, D-THRO, and D-HOA is a systematic and reproducible phenomenon, not an artifact of favorable initialization.

The selection of comparative algorithms was governed by three criteria applied jointly: (i) publication recency, with all selected algorithms having appeared in peer-reviewed journals between 2023 and 2025; (ii) demonstrated competitiveness on standard symmetric TSP benchmark instances, as reported in the respective original publications; and (iii) methodological diversity across algorithmic design paradigms, ensuring that the comparison covers a broad and representative spectrum of the discrete metaheuristic landscape. Algorithms satisfying these criteria were identified through a systematic review of recent literature indexed in major academic databases, focusing specifically on discrete adaptations of continuous metaheuristics applied to permutation-based combinatorial optimization problems. The resulting comparative set comprises seven algorithms organized into two complementary groups. The primary comparison group (Table 2) includes D-KLA[28], D-THRO[29], and D-HOA[30](all published in 2024–2025) which represent, respectively, a physics-inspired paradigm (Kirchhoff's electrical law analogy), a game-theoretic and historical wisdom-inspired paradigm, and a human-behavior-inspired paradigm modeled on hiking strategies. These three algorithms were selected as the primary comparative set because they represent the most recently published discrete metaheuristics with directly comparable experimental settings (same benchmark instances, population size, and iteration count), thereby enabling the most rigorous and methodologically consistent evaluation. The secondary comparison group (Table 4) includes D-SFOA[33], D-MPA[34], DABC-FNS[35], and DSSA[26], which span swarm-based, marine-predator-inspired, bee colony, and salp-swarm paradigms, respectively. Critically, each of these four algorithms

has been demonstrated in its original publication to outperform a wide range of earlier discrete metaheuristics (including discrete genetic algorithms, discrete particle swarm variants, and discrete grey wolf and cuckoo search adaptations) on the same benchmark instances employed in the present study. Comparison against these established state-of-the-art methods therefore provides implicit coverage of a considerably broader algorithmic landscape than the seven directly compared algorithms might suggest. Taken together, the two-tier comparison structure adopted in this study ensures both depth (through the detailed full-metric evaluation in Table 2) and avoids the methodological redundancy that would arise from including multiple algorithms sharing the same fundamental design paradigm.

The results in Table 2 reveal several mechanistically informative patterns beyond the raw numerical rankings. For the small-scale instances pr76 and rat99, D-EAO attains zero standard deviation across all 20 independent runs, indicating that the algorithm consistently converges to the same solution regardless of initialization. This behavior is attributable to the interplay between the six neighborhood operators and the adaptive factor (AF): in low-dimensional search spaces, the multi-operator diversification mechanism exhausts the locally reachable neighborhood early in the search, and the subsequent exploitation-dominant phase driven by the square-root progression of AF (Eq. (3)) locks solutions onto the global optimum before stochastic perturbations can introduce variance. Notably, D-KLA also achieves zero standard deviation on pr76, suggesting that both algorithms reach the deterministic saturation regime for this instance. However, on rat99, D-KLA's non-zero standard deviation ($SD = 2.17$) indicates that its update mechanism lacks the convergence stability that D-EAO's enzyme concentration parameter provides by regulating the influence of the best solution on the population.

The kroA100 instance offers a particularly important point of differentiation: D-EAO is the only algorithm to reach the known optimal tour length (21,282.00) in its best run, while all competing methods remain above it. This outcome reflects the effectiveness of the selective 2-opt local search, which — by concentrating computational effort on the top 20% of solutions — enables targeted edge-improvement in medium-complexity instances without the prohibitive overhead of exhaustive local search. The non-zero but low standard deviation of D-EAO on kroA100 ($SD = 1.76$) further confirms that the algorithm consistently reaches near-optimal regions, even when occasional stochasticity prevents exact reproducibility.

As problem dimensionality increases to ch130 and ch150, the performance gap between D-EAO and the competing methods widens in the Average metric, even as the Best-value differences become more moderate. This divergence between best and average performance is a key diagnostic indicator: it suggests that D-KLA, D-THRO, and D-HOA can occasionally find competitive solutions through fortuitous stochastic sampling but lack the consistency mechanisms to replicate those solutions across runs. D-EAO's lower standard deviation values on both instances ($SD = 9.23$ for ch130; $SD = 12.93$ for ch150, compared to values exceeding 13–17 for competitors) reflects the stabilizing role of the enzyme concentration parameter, which dampens the magnitude of stochastic perturbations and prevents the search from drifting excessively in high-dimensional spaces.

The pr299 instance, with 299 nodes, constitutes the most demanding test of algorithmic scalability in this benchmark set. D-EAO achieves the best tour length (48,751.02) and the lowest average (48,925.20) among all competing methods. While D-EAO's standard deviation on this instance ($SD = 143.74$) is the highest in absolute terms, this must be interpreted relative to the problem's enormous search space: the ratio of SD to average tour length remains comparatively low, and D-EAO's worst-case result (49,112.34) still falls below the best results achieved by D-KLA (48,981.39), D-THRO (48,884.96), and D-HOA (48,879.95). This indicates that even D-EAO's weakest runs outperform the strongest single runs of its competitors on this instance, underscoring the robustness of the hybrid framework at scale.

Figure 5 shows the convergence behavior of the D-EAO, D-KLA, D-THRO, and D-HOA algorithms for the pr76 dataset as a function of the number of iterations. The graph clearly shows the change in tour length achieved by each algorithm over the iterations. Figure 5 shows that the D-EAO algorithm exhibits rapid improvement from the initial iterations and significantly reduces the tour length within the first few iterations. This rapid convergence in the early stages indicates that D-EAO effectively explores the search space and can reach high-quality solutions quickly. After approximately 50 iterations, the D-EAO curve began to level off, indicating that the algorithm had stably reached an optimum or was very close to one. The D-KLA algorithm also exhibits a rapid decline in the early iterations but shows slower convergence than D-EAO. D-KLA requires more iterations to reach the optimal value, and fluctuations in solution quality are more pronounced, especially in the initial iterations, than in D-EAO. D-THRO and D-HOA algorithms, on the other hand, start the search with longer round lengths and take longer to improve. The plateau-like behavior of the D-THRO algorithm indicates that it spends more time around local solutions, delaying convergence to the optimum. While the D-HOA algorithm exhibits a more balanced decline than D-THRO, it lags behind D-EAO in early convergence. The D-EAO algorithm offers higher search efficiency than other algorithms due to its ability to achieve lower round lengths with fewer iterations and its stable convergence behavior in the early stages.

Figure 6 illustrates the convergence behavior of the considered algorithms on the rat99 TSP instance. It can be observed that the proposed D-EAO algorithm converges rapidly during the early iterations and reaches a stable solution faster than the competing methods. This indicates that D-EAO possesses an effective exploration capability at the initial stages, allowing it to quickly identify promising regions of the search space. Moreover, the smooth convergence trend reflects the algorithm's stability and consistency throughout the optimization process.

As shown in Fig. 7, the D-EAO algorithm demonstrates superior convergence performance on the kroA100 instance by achieving near-optimal solutions within fewer iterations compared to the other algorithms. While the competing methods exhibit slower convergence and noticeable fluctuations, D-EAO maintains a steady improvement pattern. This behavior highlights the effectiveness of the proposed discrete adaptation and neighborhood-based search strategy in handling medium-sized TSP instances.

Figure 8 presents the convergence curves for the ch130 dataset, which represents a higher-dimensional and more complex TSP instance. In this scenario, all algorithms experience a slower convergence rate due to the enlarged search space. However, D-EAO consistently outperforms the other methods by attaining lower tour lengths and exhibiting reduced oscillations. This result demonstrates the robustness of D-EAO in complex optimization landscapes and its ability to avoid premature convergence.

Figure 9 depicts the convergence behavior for the ch150 problem. It is evident that D-EAO achieves better solution quality compared to the competing algorithms across all iterations. Although convergence becomes more gradual as problem dimensionality increases, D-EAO maintains a clear advantage in both convergence speed and final solution quality. This confirms the scalability of the proposed approach for large-scale TSP instances.

Figure 10 illustrates the convergence behavior of the algorithms on the pr299 instance, which represents the largest and most complex TSP problem considered in this study. Due to the extensive search space, all algorithms exhibit slower convergence. Nevertheless, the proposed D-EAO algorithm consistently achieves lower tour lengths compared to the competing methods throughout the iterations. This behavior highlights the strong scalability of D-EAO and its ability to maintain effective search performance even in large-scale optimization problems.

Table 4 presents a comparative performance analysis of the proposed D-EAO algorithm and several state-of-the-art discrete metaheuristic methods on six benchmark TSP instances. The optimization process was performed using algorithms such as discrete starfish optimization algorithm (D-SFOA)[33], discrete marine predators algorithm (D-MPA)[34], discrete artificial bee colony algorithm with fixed neighborhood search (DABC-FNS)[35], and discrete salp swarm algorithm (DSSA)[26] with the dataset, employing Best and Average lap length metrics, which are widely accepted indicators for evaluating solution quality and robustness in optimization problems.

The expanded comparison in Table 4 situates D-EAO within the broader landscape of recently proposed discrete metaheuristics. In the small-scale instances pr76 and rat99, D-SFOA and DABC-FNS achieve the exact optimal tour lengths in their best runs, indicating that these algorithms can pinpoint the global optimum for low-complexity problems. However, the Average metric reveals a more nuanced picture: D-EAO attains a lower average than DABC-FNS on both pr76 (108,159.44 vs. 108,315.00) and rat99 (1,219.24 vs. 1,229.40), suggesting that DABC-FNS's ability to reach the optimum in favorable runs does not translate to consistent performance across independent trials. This trade-off between peak performance and run-to-run stability is a critical distinction in practical applications, where planning systems must deliver reliable routes across repeated executions rather than occasionally exceptional ones.

On the medium- and large-scale instances (kroA100 through pr299), D-EAO achieves the best Average performance in five of six comparisons. The single exception — the Best-value on pr299, where DSSA reports 48,714.31 against D-EAO's 48,751.02 — deserves particular attention. DSSA's slightly superior best run on this instance likely reflects the stochastic benefit of its swarm-based perturbation mechanism, which may occasionally generate globally advantageous position jumps in large search spaces. However, DSSA's substantially weaker Average performance on pr299 (49,284.23 vs. D-EAO's 48,925.20) and the absence of Wilcoxon-confirmed significance in its favor indicate that this isolated best-run advantage does not represent systematic superiority. In practical route planning contexts, where an algorithm is executed once per planning cycle, average-case reliability and solution stability are more operationally relevant criteria than single-run extremes. By these criteria, D-EAO represents the strongest overall performer among the compared methods, particularly at the scale and complexity levels that characterize real-world destination routing problems.

It is worth noting that direct comparisons with classical benchmark algorithms (such as standard GA, ACO, and PSO) were not included in the present study, as such comparisons would not meaningfully advance the paper's scientific contribution. The rationale rests on an established, well-documented performance hierarchy in the discrete TSP metaheuristic literature. Several of the algorithms listed in Table 4 have been subjected to extensive head-to-head comparisons with classical methods in their original publications. Specifically, DABC-FNS[35] was benchmarked against multiple ABC-based and neighborhood-search variants, documenting consistent superiority on the same TSPLIB instances employed here. DSSA[26] was compared with the discrete grey wolf optimizer, the discrete cuckoo search, and random-key variants, with DSSA demonstrating superior or competitive performance across problem scales. D-MPA[34] and D-SFOA[33] similarly established their advantages over a range of earlier heuristic baselines in their respective evaluations. Beyond these specific publications, the cited references in the present manuscript (including the discrete Harris hawk optimizer[17], discrete grey wolf optimizer[21], and biogeography-based optimization with 2-opt[20] collectively document that modern discrete metaheuristics with embedded local search mechanisms substantially and consistently outperform standard GA and ACO implementations on the TSPLIB benchmark suite, often by margins exceeding 5–15% in average tour length on medium- and large-scale instances. Since D-EAO has been demonstrated to outperform all four algorithms in Table 4 (which are themselves superior to classical methods) the performance ordering relative to GA and ACO follows by transitivity and is already implicit in the reported results. Replicating classical comparisons under potentially divergent implementation choices (operator selection, parameter tuning, termination criteria) would risk introducing confounds that undermine the reliability of the comparison rather than strengthening it. Regarding the stepping-ahead firefly algorithm (FA) variant specifically: while firefly-based heuristics have been explored in continuous optimization, no rigorously validated and widely reproduced discrete adaptation of this specific variant for symmetric TSP on TSPLIB benchmarks has been established in the peer-reviewed literature, making a fair, reproducible, and meaningful comparison infeasible within the current experimental framework.

To provide a transparent assessment of the computational overhead associated with each component of the proposed D-EAO framework, Table 5 reports the average execution times (in seconds) and corresponding tour

Instance	Optimal	RK		RK + NO		RK + 2-opt		RK + NO + 2-opt	
		Execution time (s)	Tour length	Execution time (s)	Tour length	Execution time (s)	Tour length	Execution time (s)	Tour length
pr76	108159.00	0.1096	153802.10	0.1875	147312.56	5.1756	108829.59	7.3563	108159.44
rat99	1211.00	0.1164	4461.14	0.2146	4296.97	12.9713	1230.76	14.0154	1219.24
kroA100	21282.00	0.1281	118832.73	0.2508	127345.53	15.0760	21282.00	18.2021	21282.00
ch130	6110.00	0.1349	40831.62	0.2387	37902.99	31.7665	6192.84	40.9152	6114.13
ch150	6528.00	0.1457	44814.72	0.2471	43203.52	54.2272	6601.55	62.9747	6530.90
pr299	48191.00	0.2576	572812.68	0.4810	383394.98	374.73	50688.74	470.5248	48751.02

Table 5. Average execution times (in seconds) and tour lengths for incremental D-EAO configurations across benchmark instances.

lengths obtained by four incremental algorithmic configurations (random-key encoding alone (RK), random-key encoding with neighbourhood operators (RK + NO), random-key encoding with 2-opt local search (RK + 2-opt), and the full D-EAO incorporating all components (RK + NO + 2-opt)) across all six benchmark TSP instances. These results serve as a component-level ablation study, isolating the contribution of each mechanism to both solution quality and computational cost.

The time complexity of D-EAO is governed by three primary operations: random-key decoding (sorting), neighborhood-based search, and selective 2-opt local refinement. Sorting an n-dimensional continuous vector to obtain a valid permutation via random-key encoding requires $O(nlogn)$ operations per candidate solution. The six neighborhood operators, each operating in $O(n)$ time, collectively contribute an additional $O(N \times n)$ overhead per iteration, where N denotes population size. The computationally dominant component is the selective 2-opt procedure, which requires $O(n^2)$ edge comparisons per application. Since 2-opt is restricted to the top 20% of the population, the per-iteration cost attributable to this phase is $O(0.2 \times N \times n^2)$. Aggregated over T iterations, the overall time complexity of D-EAO is therefore $O(N \times T \times n^2)$, with the 2-opt component constituting the dominant term for all but the smallest problem sizes. This theoretical characterization is empirically corroborated by the execution times in Table 5, which reveal three distinct computational regimes. The RK-only configuration exhibits a nearly linear growth in execution time as n increases, from 0.1096 s for pr76 ($n = 76$) to 0.2576 s for pr299 ($n = 299$), consistent with the $O(nlogn)$ decoding cost. Incorporating neighborhood operators (RK + NO) roughly doubles runtime across all instances (from 0.1875 s to 0.4810 s), reflecting the additional O(n) per-solution overhead. By contrast, introducing selective 2-opt produces a markedly nonlinear increase: the full D-EAO configuration (RK + NO + 2-opt) requires approximately 7.36 s for pr76 but 470.52 s for pr299, a factor of ~ 64× increase for a ~ 3.9× increase in problem size, closely mirroring the predicted quadratic scaling. Critically, the data simultaneously demonstrate that this computational investment is well-justified: the addition of 2-opt reduces the tour length for pr299 from 572,812.68 (RK only) to 48,751.02 (full D-EAO), representing a reduction of over 91% relative to the unrefined solution. For offline tourism route planning applications (where routes are precomputed and deployed over extended planning horizons) the observed runtimes fall well within operationally acceptable bounds.

A more comprehensive evaluation of computational efficiency requires considering solution quality and runtime jointly rather than independently. To this end, Table 5 is interpreted here not merely as an execution time record but as a quality–efficiency trade-off profile for D-EAO. The optimality gap (defined as the percentage deviation of the average tour length from the known optimal) provides a normalized quality metric that is directly comparable across problem scales. For the full D-EAO configuration, the optimality gaps across the six benchmark instances are: pr76 = 0.0004%, rat99 = 0.68%, kroA100 = 0.008%, ch130 = 0.40%, ch150 = 0.20%, and pr299 = 1.52%. These gaps are achieved within execution times of 7.36 s, 14.02 s, 18.20 s, 40.92 s, 62.97 s, and 470.52 s, respectively, indicating that the optimality gap-to-runtime ratio is consistently favorable across all tested scales. For the instances most relevant to tourism planning (ch130 through pr299, representing 130–299 destinations), the algorithm achieves sub-2% optimality gaps within 41–471 s per run. To place these results in the context of the comparative evaluation, it is important to note that the primary comparison algorithms (D-KLA, D-THRO, and D-HOA) were all evaluated under an identical computational budget ($N = 100$, $T = 200$, 20 independent runs) on the same hardware environment, meaning that differences in solution quality among these algorithms can be attributed to algorithmic effectiveness rather than differential runtime allocation. Under this fixed-budget comparison, D-EAO achieves optimality gaps that are consistently lower than those of the competing methods: on ch130, D-EAO's average gap (0.40%) is substantially smaller than those of D-KLA (0.66%), D-THRO (0.62%), and D-HOA (0.54%); on ch150, D-EAO (0.20%) outperforms D-KLA (0.54%), D-THRO (0.34%), and D-HOA (0.49%); and on the most demanding instance pr299, D-EAO (1.52%) maintains a clear advantage over D-KLA (2.08%), D-THRO (2.05%), and D-HOA (2.06%). The widening quality gap at larger problem scales confirms that D-EAO's computational investment in selective 2-opt refinement yields disproportionately greater quality gains as problem dimensionality increases, precisely the regime of greatest practical relevance for real-world multi-destination tourism route planning.

The ablation results in Table 5 afford a mechanistic explanation of why D-EAO outperforms competing methods, beyond the descriptive performance rankings reported in Tables 2 and 4. Three complementary mechanisms are identifiable. First, the contribution of the neighborhood operators is primarily one of permutation-space diversification rather than direct quality improvement. This is evidenced by the fact that adding NO to RK encoding occasionally produces slightly worse average tour lengths than RK alone (most notably on kroA100,

where RK yields 118,832.73 but RK + NO yields 127,345.53) yet the full D-EAO configuration (RK + NO + 2-opt) consistently outperforms the 2-opt-only variant (RK + 2-opt) across all instances. This seemingly counterintuitive pattern reveals a critical role of the operators: rather than improving tour quality directly, the six neighborhood operators (RS, RI, RSS, RIS, RDC, and RSDC) diversify the population into structurally distinct configurations that expose different local optima to the subsequent 2-opt refinement. The point-based operators (RS and RI) introduce targeted, low-disruption perturbations that shift individual cities without invalidating broader tour structure, enabling fine-grained escape from shallow local optima. The array-based operators (RSS, RIS, RDC, and RSDC), by operating on multi-city subsequences, produce more substantial structural rearrangements that alter entire route segments. The stochastic operator selection protocol, which applies each of the six operators with equal probability, prevents the search from becoming entrenched in any single neighborhood topology and maintains exploration breadth throughout the iterative process. Second, the 2-opt local search is the primary quality driver, responsible for the overwhelming majority of the optimality gap reduction observed in the ablation study. On pr299, the transition from RK-only (tour length 572,812.68) to RK + 2-opt (50,688.74) represents a 91.1% reduction in absolute tour length, whereas the addition of NO to the RK baseline (RK + NO: 383,394.98) accounts for only a 33.1% reduction. The mechanistic basis for 2-opt's effectiveness is well-established: the procedure systematically identifies and eliminates intersecting edges in the tour, which are among the most prevalent and correctable sources of suboptimality in Euclidean TSP instances. By restricting 2-opt refinement to the top 20% of solutions at each iteration, D-EAO concentrates this exploitation mechanism on the most promising candidates, preventing the computational cost from scaling prohibitively while ensuring that the highest-quality individuals are consistently refined. Third, and most critically for understanding D-EAO's advantage over competing methods, the interaction between NO and 2-opt is synergistic rather than merely additive. For every benchmark instance, RK + NO + 2-opt achieves strictly better results than RK + 2-opt alone: on ch130, the gap is 6,114.13 vs. 6,192.84 (a 1.27% improvement attributable solely to the NO contribution); on ch150, 6,530.90 vs. 6,601.55 (1.07%); and on pr299, 48,751.02 vs. 50,688.74 (3.82%). This synergy arises because NO diversification continuously replenishes the pool of structurally distinct, high-quality solutions available for 2-opt refinement: without NO, 2-opt converges repeatedly to the same set of local optima dictated by the initial random-key distribution; with NO, the population explores qualitatively different tour configurations from which 2-opt can reach lower-cost local optima that would otherwise be inaccessible. This population-level diversification is precisely the mechanism that distinguishes D-EAO from competing methods that rely on a single exploration paradigm: whereas D-KLA, D-THRO, and D-HOA each embed their respective search dynamics without a structured multi-operator neighborhood exploration stage, D-EAO explicitly decouples the diversification and refinement phases, enabling a more systematic coverage of the permutation landscape and ultimately producing lower optimality gaps, particularly on the complex, large-scale instances most representative of realistic tourism route planning scenarios.

The proposed system for the use of tourism points in Bitlis, Türkiye, worldwide

Bitlis province stands out as one of the important tourism centers of the Eastern Anatolia Region with its natural, historical, and cultural heritage. Nemrut Caldera Lake, Mount Süphan, Ahlat Seljuk Cemetery, the Van Lake coastline, historical madrasas, castles, and structures related to religious tourism increase Bitlis's tourism potential both nationally and internationally. This richness also brings with it a complex route planning problem requiring the efficient visitation of numerous tourist attractions within limited time and cost. In this context, a tourism-focused TSP is addressed specifically for Bitlis, and the performance of the proposed D-EAO algorithm in a real-world application scenario is evaluated.

Figure 11 shows Bitlis's location within the world and Türkiye, highlighting the province's geographical accessibility and regional context. This figure created by the authors using Google Maps (Google LLC; https://maps.google.com). Bitlis's location in the Van Lake basin and the presence of tourist attractions spread across different districts make distance and time optimization critical in visitors' route planning. Bitlis, located in the Eastern Anatolia Region, has hosted various civilizations throughout history due to its proximity to the Lake Van basin, mountainous topography, and location on historical trade routes. The province offers a wide variety of tourism types within the same destination, including cultural and historical, religious, nature, and winter tourism. When evaluated on a national scale, Bitlis presents a different profile from that of the mass tourism centers concentrated in Türkiye's western and southern regions. This makes Bitlis an important destination for alternative and thematic tourism; however, it also leads to longer distances between tourist sites and more complex visitor planning[36–40].

As shown in Suppl. Table A1 in the Appendix, the touristic points in Bitlis do not exhibit a spatially homogeneous distribution. While the districts of Ahlat and Tatvan stand out with their historical and natural attractions concentrated around Lake Van, the central district of Bitlis is the focal point of cultural and religious tourism through its historical mosques, madrasas, castles, and museums[37]. In contrast, the districts of Mutki, Hizan, and Güroymak have touristic assets in rural, mountainous areas that are mostly considered within the scope of nature and religious tourism[38,41]. Bitlis's potential for religious tourism is particularly evident in the Ohin (Mutki), Norşin (Güroymak), and Hizan madrasas, as well as the religious structures that have developed around these centers. The historical and cultural importance of these structures demonstrates that Bitlis holds a special position among the religious centers of Eastern Anatolia[39]. These points of attraction for religious tourism are scattered across different districts, underscoring the importance of optimal visitor flow in creating thematic religious routes. From the perspective of nature and winter tourism, Bitlis's high altitude, long snow cover, and mountainous topography enhance the province's winter tourism potential. Areas such as Bitlis Ski Center and Nemrut Ski Center are among the province's prominent winter tourism destinations[39].

However, natural areas such as Nemrut Caldera Lake, Mount Süphan, Lake Nazik, and valleys are important centers of attraction for nature, trekking, and photography tourism during the summer and transitional

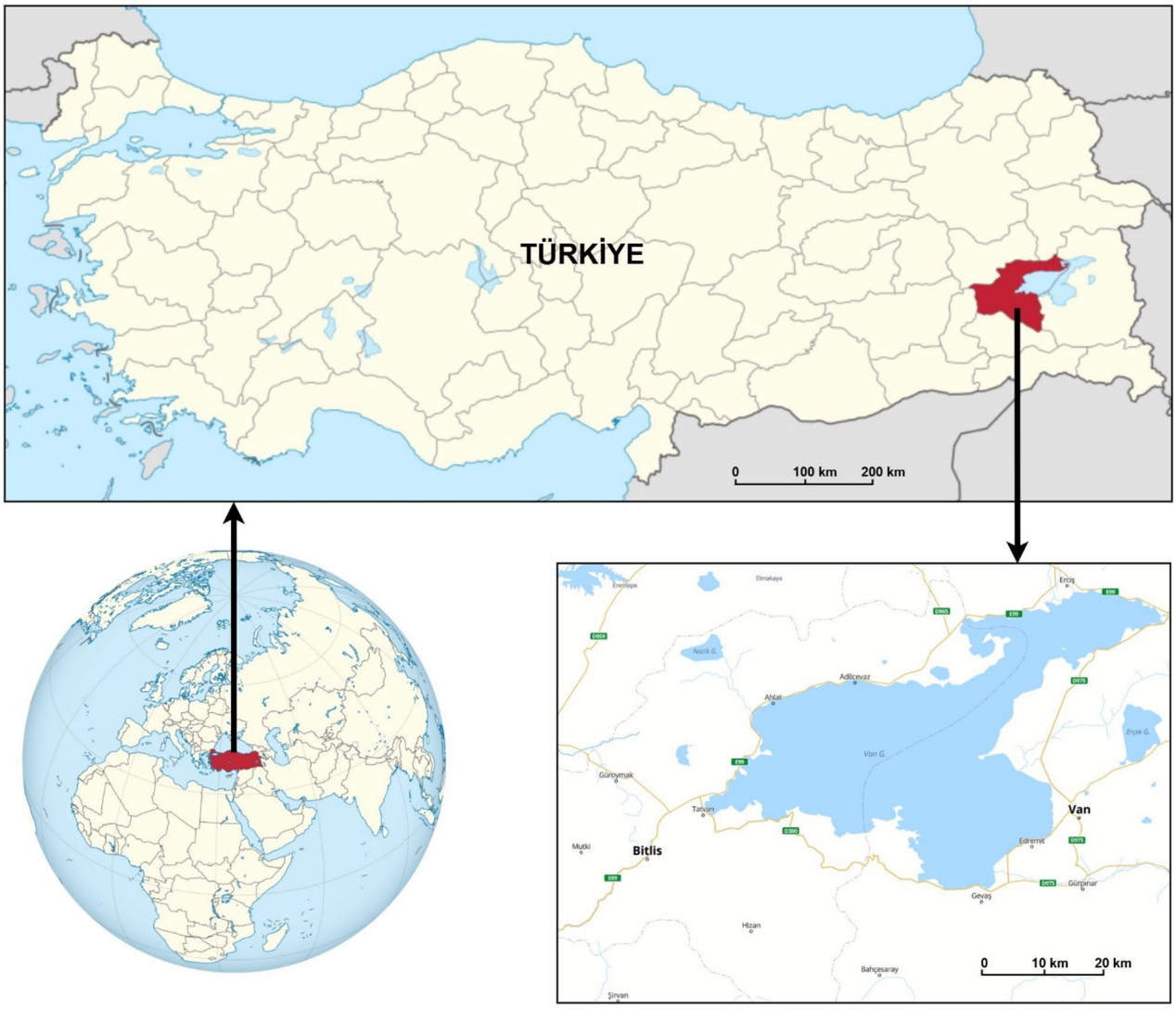


Fig. 11. Geographical location of Bitlis within Türkiye and the world. Map data © 2025 Google (https://maps.google.com). Map generated by the authors using Google Maps for illustrative purposes.

Metric	D-EAO	D-KLA	D-THRO	D-HOA
Best	670.10	670.30	670.20	673.70
Worst	671.30	677.90	681.90	681.70
Average	670.44	673.45	674.53	675.22
SD	0.35	2.27	3.99	2.29

Table 6. Statistical performance results of algorithms for the Bitlis tourism TSP problem.

seasons[36–40]. This multi-themed and spatially dispersed tourism structure transforms tourism route design in Bitlis from a purely promotional sequencing problem into a decision problem aimed at minimizing distance, time, and cost[42]. The route design approach, encompassing all 46 tourist points listed in Suppl. Table A1, provides a practical application of the "complete visit" goal in destinations such as Bitlis. When the findings in the literature on Bitlis's tourism potential are combined with the geographical data in Suppl. Table A1, it is evident that optimization-based route planning approaches can serve as an effective decision-support tool in multi-centered, multi-themed destinations like Bitlis.

The quantitative results in Table 6 confirm the graphical findings in Figs. 12 and 13. The D-EAO algorithm produced shorter tour lengths than other methods across best, worst, and average values. In particular, the low SD value indicates that the algorithm produced similar solutions across runs and that its solutions are highly

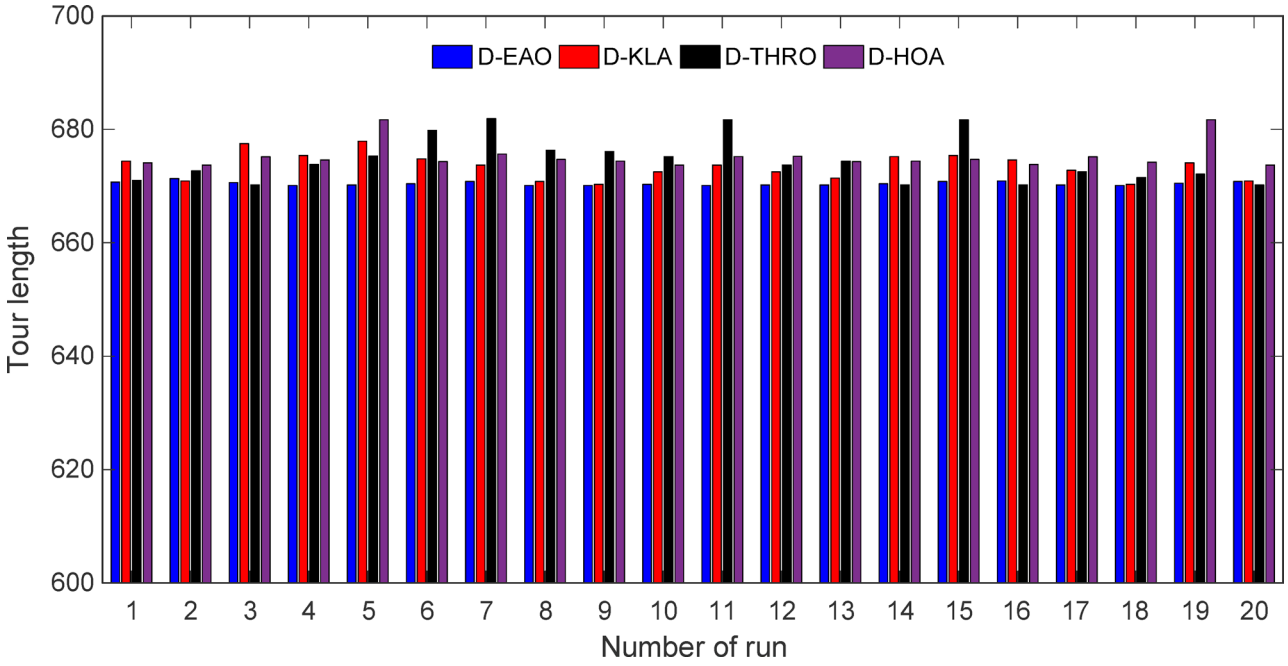


Fig. 12. Performance distribution of algorithms across 20 runs for the Bitlis tourism TSP problem.

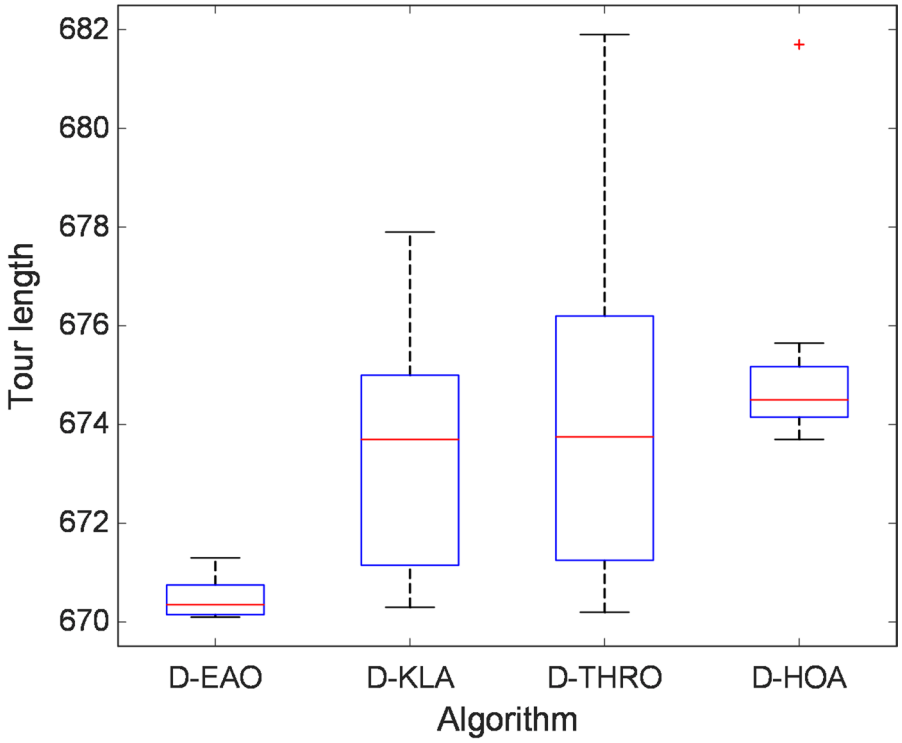


Fig. 13. Boxplot comparison of algorithm performance for the Bitlis tourism TSP problem.

stable. This feature is crucial for creating predictable, sustainable routes when planning tourism in real-world applications.

The Wilcoxon signed-rank test results in Table 7 indicate that the D-EAO algorithm also shows a statistically significant advantage. The fact that all obtained p-values are below the 0.05 significance level indicates that the proposed method is more successful than competing algorithms in the TSP model for Bitlis tourism. These results confirm that D-EAO performs well not only in theoretical but also in applied scenarios.

D-EAO versus D-KLA		D-EAO versus D-THRO		D-EAO versus D-HOA	
p-value	Winner	p-value	Winner	p-value	Winner
1.6221E – 04	D-EAO	5.9276E – 04	D-EAO	8.8074E – 05	D-EAO

Table 7. Wilcoxon signed-rank test results for the Bitlis tourism TSP problem.

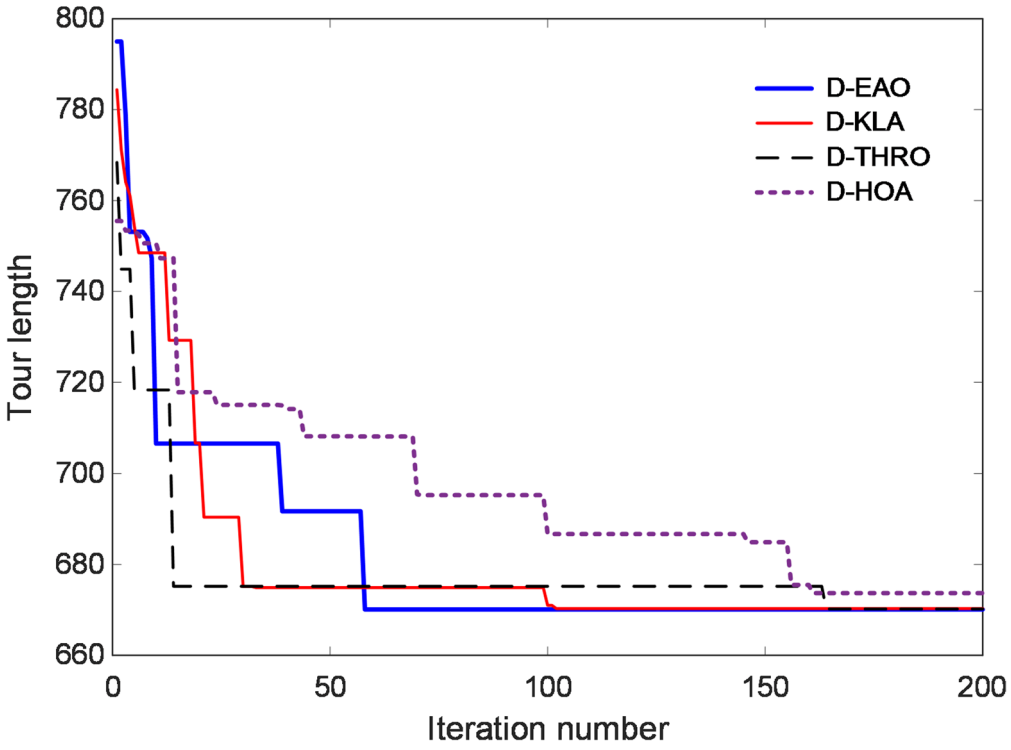


Fig. 14. Convergence curves of compared algorithms on the Bitlis TSP instance.

Algorithm	Best tourism routes	Figure
D-KLA	39→5→46→40→35→12→8→27→31→24→36→34→33→3→25→29→1→18→16→6→38→21→20→37→11→30→26→28→14→43→15→32→4→2→45→10→17→9→22→44→13→7→19→23→42→41→39	Figure 15
D-HOA	27→19→23→42→41→39→5→46→40→35→12→18→16→6→38→21→20→37→11→30→26→28→14→43→15→32→4→2→45→13→22→9→44→7→10→17→31→24→36→34→33→3→25→29→1→8→27	Figure 16
D-THRO	45→13→22→9→44→7→10→17→19→23→42→41→39→5→46→40→35→12→8→27→31→24→36→34→33→3→25→29→1→18→16→6→38→21→20→37→11→30→26→28→14→43→15→32→4→2→45	Figure 17
D-EAO	12→8→27→31→24→36→34→33→3→25→29→1→18→16→6→38→21→20→37→11→30→26→28→14→43→15→32→4→2→45→13→10→7→44→19→9→22→17→23→42→41→39→5→46→40→35→12	Figure 18

Table 8. Best tourism routes obtained by different algorithms for Bitlis.

Examining the convergence curves in Fig. 14, the D-EAO algorithm shows rapid improvement from early iterations and reaches a stable solution quickly. This increases the practical usability of the algorithm, especially in tourism planning applications with time constraints.

Table 8; Figs. 15, 16, 17 and 18 visually present the best tourism routes obtained by different algorithms. The route generated by the D-EAO algorithm suggests a balanced visitor sequence that minimizes unnecessary distance repetition by considering the geographic proximity of 46 tourist points across Bitlis. The distances between the 46 points (see Table A2 in the Appendix) in the real-world case study were computed using Google Maps' road network. This route has the potential to reduce both time and transportation costs for visitors.

Table 8 in the appendices presents detailed distance matrices showing the round-trip distances between 46 points of interest in Bitlis. These matrices serve as the primary input to the TSP model, enabling the algorithms to produce realistic, implementable solutions. The direct use of distance information enables the proposed system to move beyond a theoretical approach and become a decision-support tool that reflects real-world conditions.

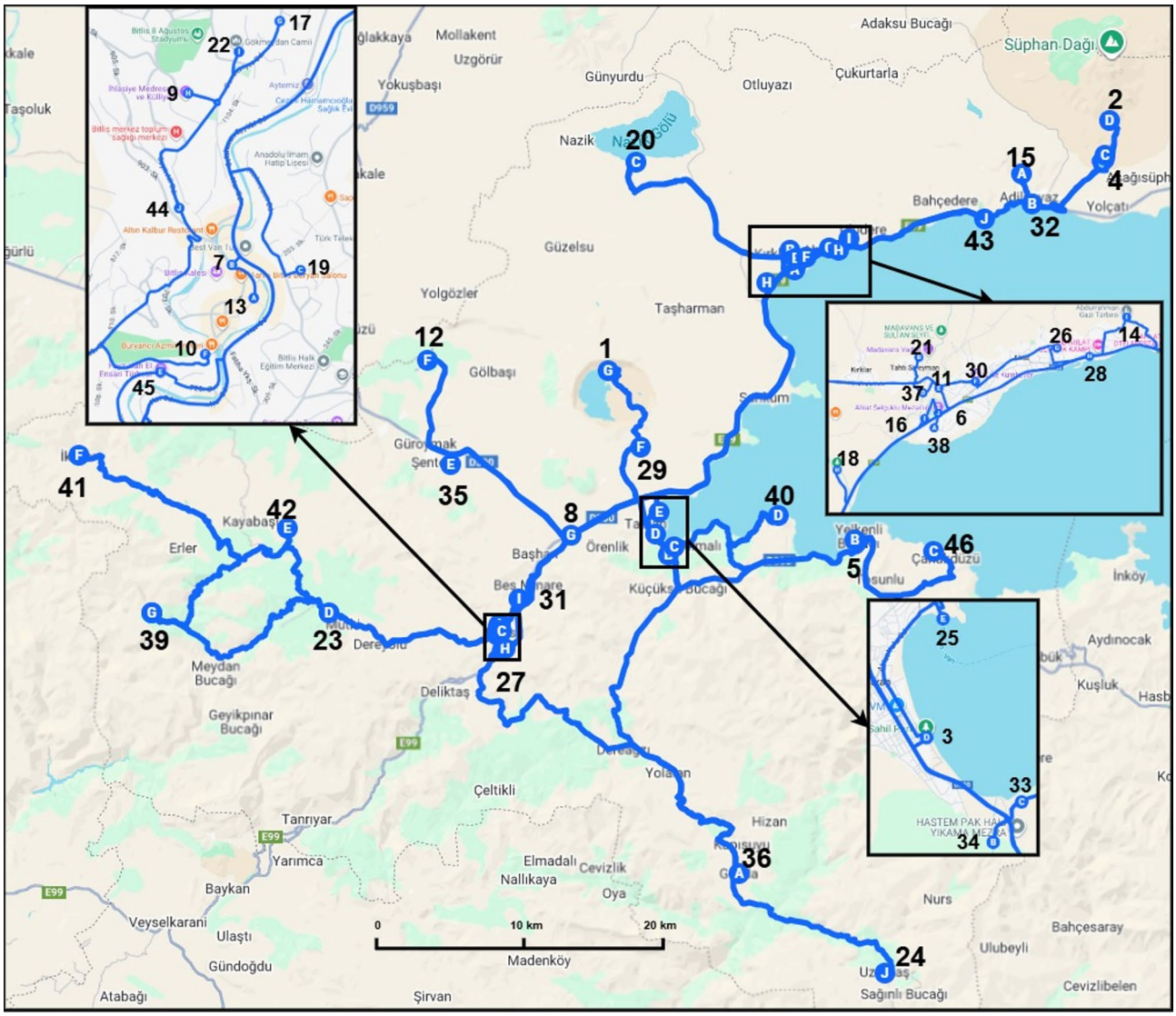


Fig. 15. Best tourism route for the D-KLA algorithm, covering all 46 tourist points in Bitlis province. The visit sequence follows the order specified in Table 8. Map data © 2025 Google (https://maps.google.com). Route visualization prepared by the authors using Google Maps based on the optimized sequence generated by D-KLA.

Figure 15 shows the shortest route generated by the D-KLA algorithm for the TSP problem, which aims to visit all 46 tourist points in Bitlis province. All tourist points are fixed in the same locations on the map, and the route is formed by the order of visits specified in Table 8. When examining the map shown in Fig. 15, it is evident that the D-KLA algorithm generally produces a holistic route that covers tourist points in Bitlis city center, the area around Lake Van, and rural areas. However, the route geometry in Fig. 15 includes transitions that do not fully preserve regional clustering, especially in densely populated cultural heritage areas around the city center and natural attractions near the lake shore.

This situation leads to consecutive visits to more distant points rather than to short-distance points, creating a structure that can increase the total tour length. Bitlis' tourism potential is shaped by the combination of historical structures, religious tourism elements, and natural landscapes. While the route generated by D-KLA encompasses this diversity, it does not utilize the principle of geographical proximity between tourist points as effectively as D-EAO. The longer, more indirect route lines, especially in transitions to mountainous areas, can create disadvantages in terms of travel time and visitor fatigue. These spatial observations are consistent with the higher mean and standard deviation values given for D-KLA in Table 6 compared to D-EAO. Therefore, Fig. 15 shows that the D-KLA algorithm offers acceptable but limited performance in terms of stability and cost minimization in destinations with heterogeneous tourism structures such as Bitlis. Figure 16 shows the shortest route to 46 tourist points in Bitlis province using the D-HOA algorithm. The map reflects the geographical projection of the D-HOA route sequence given in Table 8.

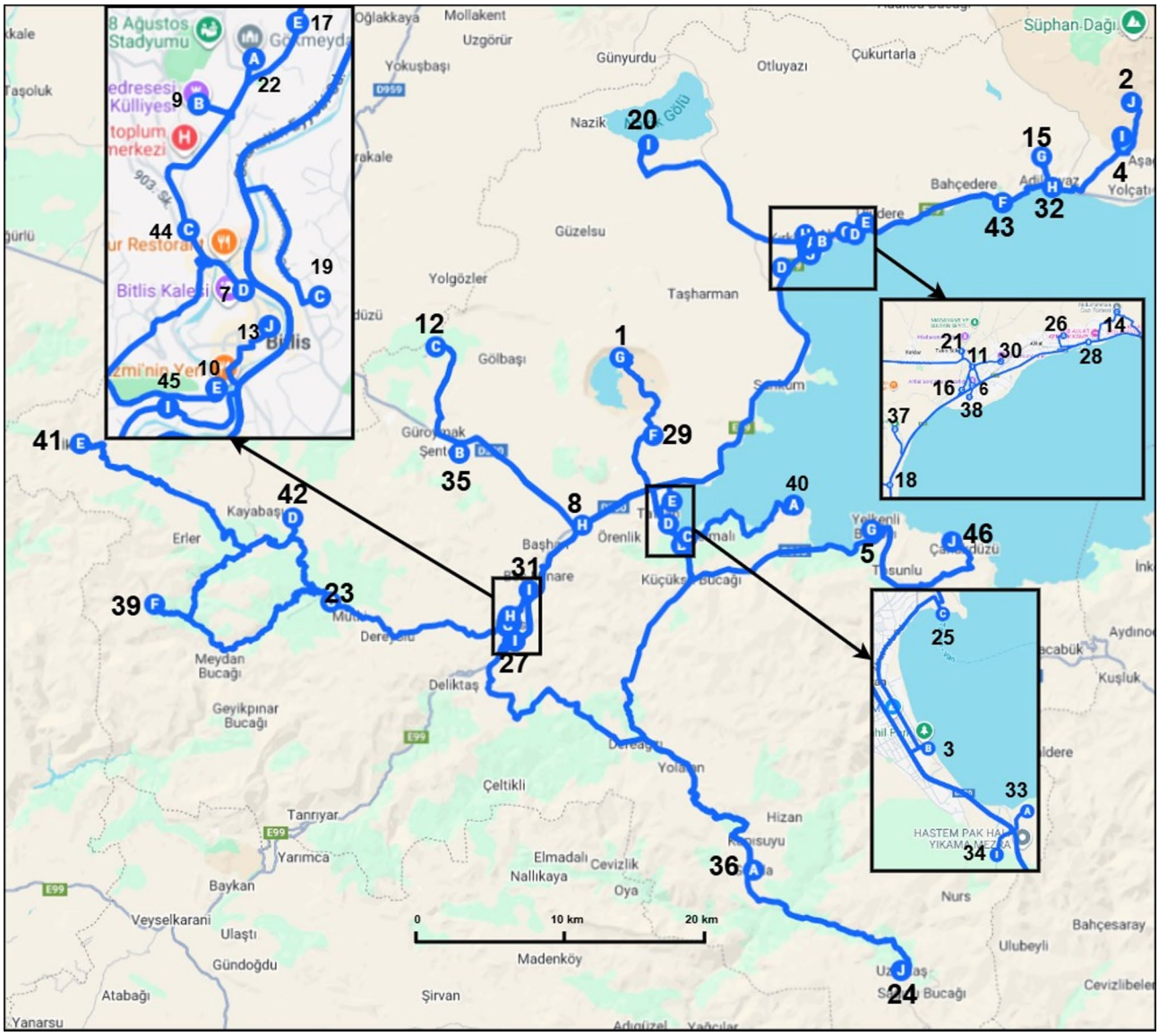


Fig. 16. Best tourism route for the D-HOA algorithm, covering all 46 tourist points in Bitlis province. The visit sequence follows the order specified in Table 8. Map data © 2025 Google (https://maps.google.com). Route visualization prepared by the authors using Google Maps based on the optimized sequence generated by D-HOA.

Examining the route in Fig. 16, it is seen that the D-HOA algorithm creates longer connections, especially in inter-regional transitions, and maintains spatial continuity relatively weakly. Considering Bitlis' mountainous topography and the fragmented settlement structure, such long jumps can increase transportation costs and lead to time losses in practical tourism planning. The transitions, particularly along the center-coast-south line on the map, show that the route geometry exhibits a more "broken" and irregular structure. While cultural heritage sites in Bitlis are generally concentrated in the center and its immediate surroundings, natural tourism points are spread over a wider area. Although the D-HOA route covers these points, it creates suboptimal sequences between cultural and natural attractions; this negatively affects route integrity. This situation is consistent with the highest average tour length values reported for D-HOA in Table 6. Consequently, Fig. 16 shows that the D-HOA algorithm can generate a viable route in the TSP solution; however, it is limited in producing the lowest-cost, most balanced route in destinations where tourist points are geographically scattered, such as Bitlis. The shortest route to the designated tourist point in Bitlis obtained using the D-THRO algorithm is shown in Fig. 17. Examining Fig. 17, it can be seen that the route generated by the D-THRO algorithm covers most of the tourist points in a logical order; however, in certain areas, it creates transitions that skip clusters rather than traversing intra-clusters. This situation leads to unnecessarily long connections, especially between natural tourism areas in the rural, mountainous parts of Bitlis. These transitions on the map stand out as the main factors increasing the route length. Since Bitlis' tourism structure offers a destination profile in which religious, cultural, lakeside, and nature tourism converge, the D-THRO route captures this multi-component structure, but the visit order

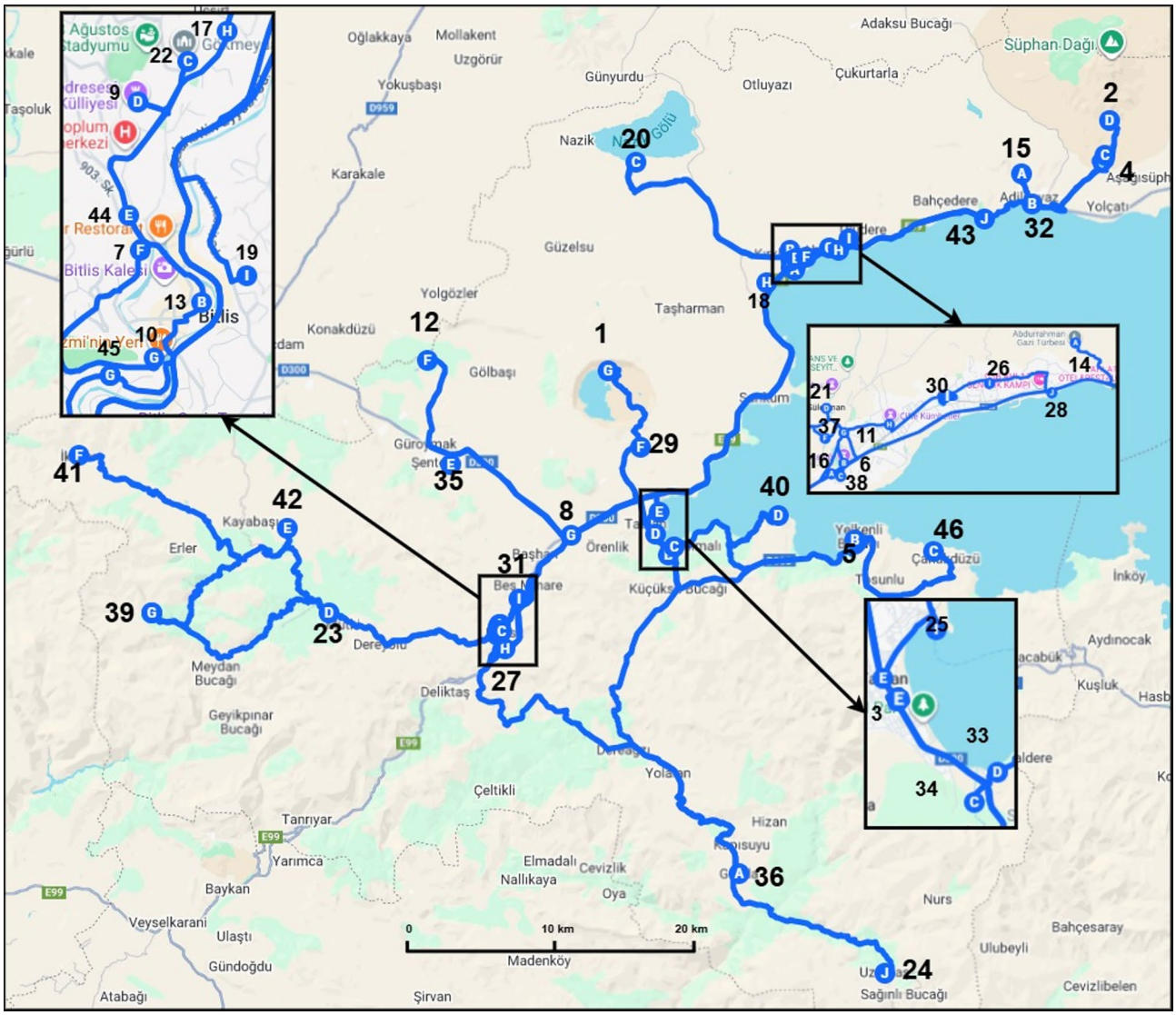


Fig. 17. Best tourism route for the D-THRO algorithm, covering all 46 tourist points in Bitlis province. The visit sequence follows the order specified in Table 8. Map data © 2025 Google (https://maps.google.com). Route visualization prepared by the authors using Google Maps based on the optimized sequence generated by D-THRO.

appears susceptible to run-to-run variability. Indeed, the high standard deviation reported for D-THRO in Table 6 indicates that the route geometry shown in Fig. 17 cannot be reproduced across all runs. In this context, Fig. 17 shows that the D-THRO algorithm can produce good solutions under certain conditions; however, it also reveals that D-EAO lags behind in terms of commitment to finding solutions in destinations such as Bitlis, where tourist spots are spatially heterogeneous.

Figure 18 shows the shortest and most stable route obtained by the D-EAO algorithm for the TSP problem, which aims to visit all 46 tourist points in Bitlis province, located in the Eastern Anatolia Region. All tourist points are fixed at the same locations on the map, and only the visit order is arranged according to the best solution found by the D-EAO algorithm. In this context, Fig. 18 reflects the spatial representation of the D-EAO route sequence given in Table 8. Examining the map in Fig. 18, it is clearly seen that the D-EAO algorithm adopted a visit sequence that maintains high geographical continuity and regional clustering when creating the route. The route generated by D-EAO. The D-EAO algorithm treats historical structures in and around Bitlis city center, natural attractions in rural areas, and tourist points around the lake as logical subsets, avoiding unnecessary transitions between them. This feature played a decisive role in minimizing the route's total length. Furthermore, while visiting closely situated cultural heritage sites sequentially, it also organized transitions to natural areas based on their geographical proximity. Thus, the route offers a balanced and feasible visit plan for both cultural and nature tourism. Examining the connection lines on the map, it is evident that the D-EAO algorithm prefers short, direct transitions, especially in mountainous and relatively difficult-to-access regions; thus, it creates a route geometry aimed at reducing travel time and transportation costs. From a tourism planning perspective, the

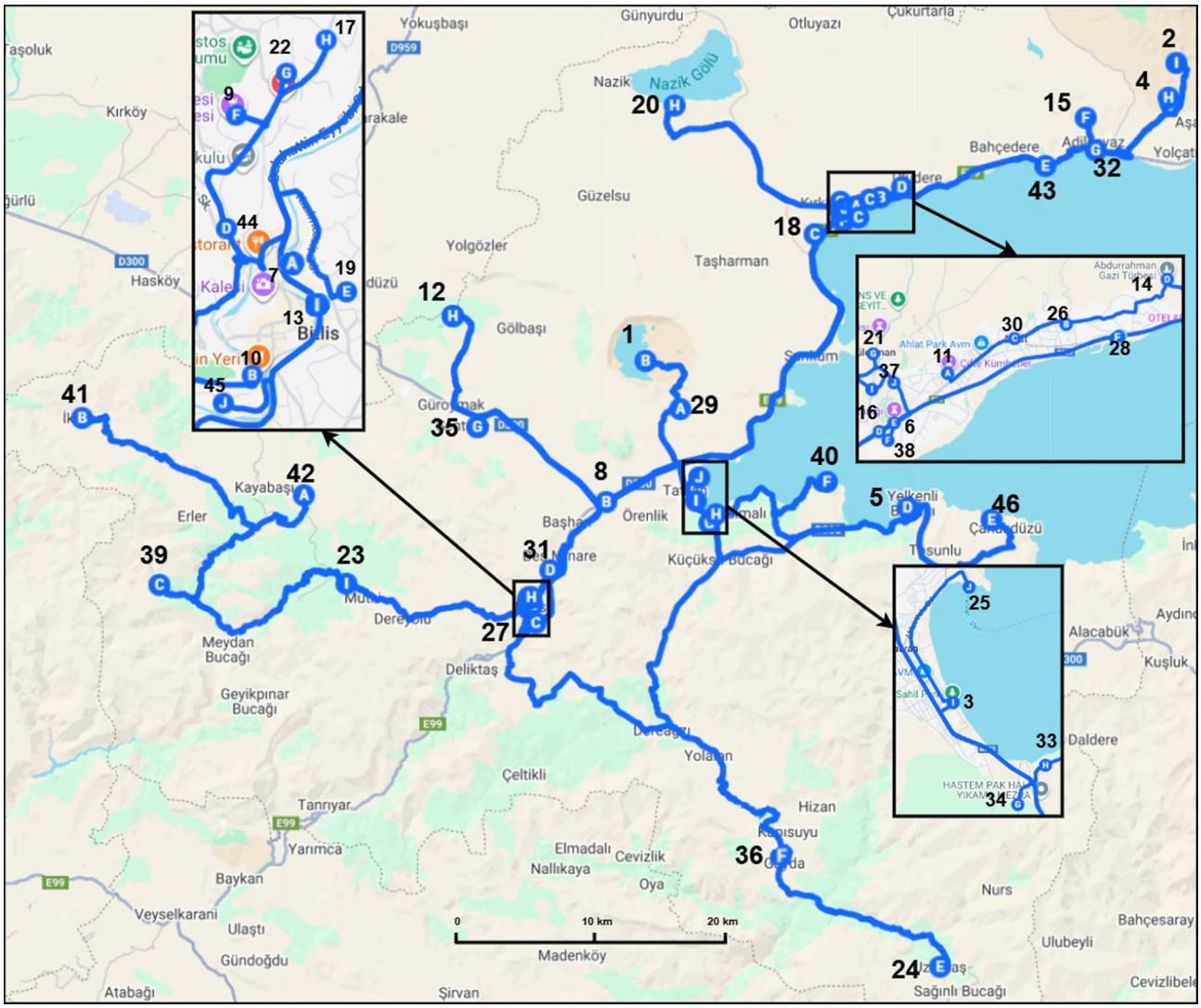


Fig. 18. Best tourism route for the D-EAO algorithm, covering all 46 tourist points in Bitlis province. The visit sequence follows the order specified in Table 8. Map data © 2025 Google (https://maps.google.com). Route visualization prepared by the authors using Google Maps based on the optimized sequence generated by D-EAO.

D-EAO route shown in Fig. 18 can provide visitors with a more time- and cost-effective experience in a scattered and heterogeneous destination like Bitlis. Especially on day trips or short tour programs that visit numerous tourist sites within a limited time, this route structure can offer significant advantages by increasing visitor satisfaction, reducing travel fatigue, and making tour programs more sustainable.

This study presents a comparative analysis of the route maps shown in Figs. 15, 16, 17 and 18, offering significant practical implications for tourism planning and destination management. When the four algorithms are evaluated together, they clearly reflect the exploration-exploitation balance and decision-making mechanisms in the discrete solution space of each algorithm. Firstly, the results demonstrate that planning tourist destinations not only with a numerically optimized visitor sequence but also with consideration of geographical continuity and spatial clustering plays a decisive role in the effectiveness of tourism activities. In destinations like Bitlis, where tourist attractions are spread across a wide, heterogeneous geography, the quality of route geometry directly affects visit duration, transportation costs, and tourist satisfaction. The routes generated by the D-KLA, D-HOA, and D-THRO algorithms are shown in Figs. 15, 16 and 17 generally offer holistic tours encompassing Bitlis's cultural and natural tourist attractions; however, regional clustering is not adequately preserved on these routes, with longer, more costly transitions occurring more frequently between central, coastal, and rural areas. Such transitions result in a more broken and irregular structure of route lines on the map; In practice, this creates a pattern that can lead to increased travel time and decreased efficiency of tour programs. These spatial observations are consistent with the higher average tour length and standard deviation values reported in Table 6 for the algorithms in question. In contrast, the D-EAO route shown in Fig. 18 exhibits a more fluid, balanced, and geographically continuous geometry than those of the other algorithms. The D-EAO algorithm treats the

densely populated cultural heritage sites in Bitlis city center and the natural tourism points around Lake Van as logical subsets, largely avoiding unnecessary long transitions between these subsets. This orderly structure on the map not only minimizes the total tour length but also allows visitors to progress along the route with fewer return journeys and changes of direction. The findings demonstrate that D-EAO is a powerful optimization tool that can make tourism planning more efficient, economical, and sustainable in destinations like Bitlis, which have numerous, geographically dispersed tourist attractions. In this respect, the study provides an important application example for both the tourism sector and intelligent route planning systems. Figures 15, 16, 17 and 18 were manually visualized by the authors using Google Maps (Google LLC; https://maps.google.com) based on the optimized route sequences generated by the proposed algorithms.

Conclusion

In this study, a discrete version of the EAO, namely D-EAO, was proposed and successfully applied to the TSP. The proposed approach integrates random-key encoding, multiple neighborhood operators, and a selective 2-opt local search mechanism to effectively adapt the original continuous EAO framework to permutation-based combinatorial optimization problems. Through this hybrid structure, D-EAO achieves a robust balance between exploration and exploitation, enabling it to generate high-quality and stable solutions across different problem scales. The performance of the proposed D-EAO algorithm was first evaluated on well-known benchmark TSP instances with varying dimensions. Experimental results demonstrated that D-EAO consistently outperformed or matched competing discrete metaheuristic algorithms across best, average, and standard deviation metrics. Furthermore, the Wilcoxon signed-rank test confirmed that the observed performance improvements were statistically significant in most cases, particularly for medium- and large-scale TSP instances. These findings indicate that D-EAO is not only competitive but also reliable and scalable for complex combinatorial search spaces. Beyond benchmark problems, this study presents a real-world application by addressing a tourism-oriented route planning problem for the province of Bitlis, Türkiye. A total of 46 tourist locations, including natural attractions, historical landmarks, cultural heritage sites, and recreational areas, were considered. The distances between these locations were modeled using detailed distance matrices provided in the appendices, ensuring that the optimization process reflects realistic geographical and transportation conditions. This application highlights the practical relevance of the proposed algorithm and demonstrates its capability to handle real-life routing scenarios. The experimental analysis of the Bitlis tourism case revealed that D-EAO produced shorter, more consistent tourism routes than competing algorithms. Graphical analyses, plots, convergence curves, and statistical tests collectively showed that D-EAO converges faster, exhibits lower variability across multiple runs, and maintains stable solution quality. The best tourism route developed by D-EAO covers all 46 tourist destinations while minimizing total travel distance, offering a viable, efficient itinerary that can reduce travel time and transportation costs for visitors. From a tourism planning perspective, the results suggest that metaheuristic-based optimization approaches such as D-EAO can serve as powerful decision-support tools for destination management organizations, travel agencies, and local authorities. Efficient route planning can contribute to more sustainable tourism practices by reducing unnecessary travel, lowering fuel consumption, and improving visitor satisfaction through well-structured itineraries. In this context, the proposed system provides a methodological foundation for smart tourism applications in regions with rich but spatially dispersed touristic assets, such as Bitlis.

Future research directions may include extending the proposed model to multi-objective formulations by incorporating additional criteria such as travel time, visiting priorities, seasonal constraints, or tourist preferences. Moreover, integrating real-time traffic data or dynamic conditions into the optimization framework could further enhance its applicability. Overall, this study's findings demonstrate that the proposed D-EAO algorithm is an effective, flexible, and practical solution for both benchmark TSP problems and real-world tourism route-optimization tasks.

Data availability

All related data are available within the manuscript and the Appendix.

Received: 13 February 2026; Accepted: 3 June 2026
Published online: 16 June 2026

References

1. Sylejmani, K., Abdurrahmani, V., Ahmeti, A. & Gashi, E. Solving the tourist trip planning problem with attraction patterns using meta-heuristic techniques. *Inform. Technol. Tourism*. **26** (4), 633–678 (2024).
2. Scianna, M. The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems. *Math. Comput. Simul.* **218**, 357–382 (2024).
3. Hyso, A. & Gjylapi, D. Comparative metaheuristic approaches to tourist itinerary optimization in low-infrastructure urban contexts: A case study of Tirana. *Informatica* **49**(36), 217–232 (2025).
4. Jewpanya, P., Nuangpirom, P., Pitjamit, S. & Nakkiew, W. Optimized travel itineraries: Combining mandatory visits and personalized activities. *Algorithms* **18** (2), 110 (2025).
5. Zhong, J., Wang, X. & Li, L. Optimization for the multiday urban personalized trip design problem with time windows and transportation mode recommendations. *Transp. Res. Rec.* **2677** (5), 1349–1366 (2023).
6. Ruiz-Meza, J. & Montoya-Torres, J. R. A systematic literature review for the tourist trip design problem: Extensions, solution techniques and future research lines. *Oper. Res. Perspect.* **9**, 100228 (2022).
7. Divsalar, G., Divsalar, A., Jabbarzadeh, A. & Sahebi, H. An optimization approach for green tourist trip design. *Soft. Comput.* **26** (9), 4303–4332 (2022).
8. Kedkaew, S., Nakkiew, W., Jewpanya, P. & Nakkiew, W. A novel tourist trip design problem with stochastic travel times and partial charging for battery electric vehicles. *Mathematics* **12**(18), 2227–7390 (2024).

9. Martin, X. A., Panadero, J. & Juan, A. A. An agile optimization algorithm for the tourist trip design problem with type-covering constraints. *AIMS Math.* **11** (1), 2458–2480 (2026).
10. Zheng, W., Liao, Z. & Lin, Z. Navigating through the complex transport system: A heuristic approach for city tourism recommendation. *Tour. Manag.* 81, 104162 (2020).
11. Wu, L., Gu, T., Chen, Z., Zeng, P. & Liao, Z. Personalized day tour design for urban tourists with consideration to CO_2 emissions. *Chin. J. Popul. Resour. Environ.* **20** (3), 237–244 (2022).
12. Ding, Y., Zhang, L., Huang, C. & Ge, R. Two-stage travel itinerary recommendation optimization model considering stochastic traffic time. *Expert Syst. Appl.* **237**, 121536 (2024).
13. Ji, H., Zheng, W., Zhuang, X. & Lin, Z. Explore for a day? Generating personalized itineraries that fit spatial heterogeneity of tourist attractions. *Inf. Manag.* **58** (8), 103557 (2021).
14. Sebastia, L. & Marzal, E. Extensions of the tourist travel design problem for different travel styles. *Proc. Comput. Sci.* **176**, 339–348 (2020).
15. Rodan, A., Al-Tamimi, A. K., Al-Alnemer, L., Mirjalili, S. & Tiňo, P. Enzyme action optimizer: A novel bio-inspired optimization algorithm. *J. Supercomput.* **81** (5), 686 (2025).
16. Bean, J. C. Genetic algorithms and random keys for sequencing and optimization. *ORSA J. Comput.* **6** (2), 154–160 (1994).
17. Gharehchopogh, F. S. & Abdollahzadeh, B. An efficient Harris hawk optimization algorithm for solving the travelling salesman problem. *Cluster Comput.* **25** (3), 1981–2005 (2022).
18. Sun, S., Tong, Y., Qi, B., Wang, Z. & Wang, X. Hybrid particle swarm optimization algorithm for traveling salesman problem based on ternary optical computer. *Proc. Comput. Sci.* **243**, 1280–1287 (2024).
19. Zhang, Z. & Yang, J. A discrete cuckoo search algorithm for traveling salesman problem and its application in cutting path optimization. *Comput. Ind. Eng.* **169**, 108157 (2022).
20. Tsai, C. H. et al. A biogeography-based optimization with a greedy randomized adaptive search procedure and the 2-opt algorithm for the traveling salesman problem. *Sustainability* **15** (6), 5111 (2023).
21. Panwar, K. & Deep, K. Discrete grey wolf optimizer for symmetric travelling salesman problem. *Appl. Soft Comput.* **105**, 107298 (2021).
22. Sarucan, A. & Berkaya, M. F. Combining 3-opt and improved discrete cuckoo search algorithm for the traveling salesman problem. *Discrete Dyn. Nat. Soc.* **2024** (1), 1953277 (2024).
23. Jünger, M., Reinelt, G. & Rinaldi, G. The traveling salesman problem. *Handb. Oper. Res. Manag. Sci.* **7**, 225–330 (1995).
24. Ouaarab, A., Ahiod, B. & Yang, X. S. Random-key cuckoo search for the travelling salesman problem. *Soft. Comput.* **19** (4), 1099–1106 (2015).
25. Zheng, R. Z., Zhang, Y. & Yang, K. A transfer learning-based particle swarm optimization algorithm for travelling salesman problem. *J. Comput. Des. Eng.* **9** (3), 933–948 (2022).
26. Panwar, K. & Deep, K. Discrete salp swarm algorithm for euclidean travelling salesman problem. *Appl. Intell.* **53** (10), 11420–11438 (2023).
27. Krishna, M. M., Panda, N. & Majhi, S. K. Solving traveling salesman problem using hybridization of rider optimization and spotted hyena optimization algorithm. *Expert Syst. Appl.* **183**, 115353 (2021).
28. Ghasemi, M. et al. Kirchhoff's law algorithm (KLA): A novel physics-inspired non-parametric metaheuristic algorithm for optimization problems. *Artif. Intell. Rev.* **58** (10), 325 (2025).
29. Wang, L. et al. Tianji's horse racing optimization (THRO): A new metaheuristic inspired by ancient wisdom and its engineering optimization applications. *Artif. Intell. Rev.* **58** (9), 282 (2025).
30. Oladejo, S. O., Ekwe, S. O. & Mirjalili, S. The hiking optimization algorithm: A novel human-based metaheuristic approach. *Knowl. Based Syst.* **296**, 111880 (2024).
31. Woolson, R. F. Wilcoxon signed-rank test. In *Wiley Encyclopedia of Clinical Trials*. 1–3 (2007).
32. Eker, E., Kayri, M., Ekinci, S. & Izci, D. A new fusion of ASO with SA algorithm and its applications to MLP training and DC motor speed control. *Arab. J. Sci. Eng.* **46** (4), 3889–3911 (2021).
33. Aktaş, M. & Kiliç, F. *Discrete Starfish Optimization Algorithm for Symmetric Travelling Salesman Problem* (IEEE Access, 2025).
34. Kumar, M., Panwar, K. & Deep, K. Discrete marine predators algorithm for symmetric travelling salesman problem. *Evol. Intell.* **17** (5), 3833–3848 (2024).
35. Li, X., Zhang, S. & Shao, P. Discrete artificial bee colony algorithm with fixed neighborhood search for traveling salesman problem. *Eng. Appl. Artif. Intell.* **131**, 107816 (2024).
36. Savgın, E. C. & Belli, S. Bitlis ilinin turizm çeşitliliğinin incelenmesi. *Int. J. Acad. Value Stud.* **3** (17), 1–13 (2017).
37. Özkanli, Ü. & Şevgin, T. Sanatçı Gözüyle Bitlis İlinin Tarihi ve Turistik Mekanları. *J. Acad. Tourism Stud.* **6** (Özel Sayı), 65–76 (2025).
38. Köseoğlu, A. & Korkmaz, Ö. Alternatif Turizm Bağlamında Bitlis İli Bisiklet Turizmi Rotalarının Değerlendirilmesi. *Söke İşletme Fakültesi Dergisi*. **1** (1), 26–39 (2024).
39. Güleryüz, U., Yayla, E. & Güçlü, C. *Turizm ve Ahlat, in Ahlatta Sürdürülebilirlik Çerçevesinde Disiplinlerarası Yaklaşımlar* (Paradigma Akademi Yayınevi, 2025).
40. Elmastaş, N. & Adıgüzel, F. *Bitlis İli Turizminin Mevcut Durumu ve Ziyaretçi Projeksiyonları* (Akademisyen Yayınevi, 2025).
41. Şahap, A. *Geleneksel Meskenlerin Kültür Turizmi Açısından Değerlendirilmesi Bitlis-Hizan Uzuntaş* (Köyü Taş Evleri Akademisyen Yayınevi, 2025).
42. Alkan, A. & Adıgüzel, A. D. *Tarihi Seyahet Rotalarının Yeniden Canlandırılması: Evliya* (Çelebi'nin Bitlis Yolculuğu Akademisyen Yayınevi, 2025).

Author contributions
All authors contributed equally.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-026-56861-3.

Correspondence and requests for materials should be addressed to M.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026