

Article

Not peer-reviewed version

The New Mushroom–Weed Hybrid Reproduction Optimization Algorithm and Its Application to Tourist Route Optimization

[Domagaj Palinic](#), [Rea Aladrovic](#), [Marina Ivasic-Kos](#)^{*}, [Jonatan Lerga](#)

Posted Date: 14 February 2026

doi: 10.20944/preprints202602.1159.v1

Keywords: metaheuristic algorithms; nature-inspired algorithms; ACO; MRO; IWO; hybrid algorithm; multi-objective optimization; route optimization; tourist routing



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

The New Mushroom–Weed Hybrid Reproduction Optimization Algorithm and Its Application to Tourist Route Optimization

Domagoj Palinic ¹, Rea Aladrovic ², Marina Ivasic-Kos ^{3,*} and Jonatan Lerga ⁴

¹ Faculty of Engineering, and Faculty of Informatics and Digital Technologies, University of Rijeka, 51000 Rijeka, Croatia

² Faculty of Informatics and Digital Technologies, University of Rijeka, 51000 Rijeka, Croatia

³ Centre for Artificial Intelligence, Faculty of Informatics and Digital Technologies, University of Rijeka, 51000 Rijeka, Croatia

⁴ Centre for Artificial Intelligence, Faculty of Engineering, University of Rijeka, 51000 Rijeka, Croatia

* Correspondence: marinai@uniri.hr

Abstract

Nature-inspired metaheuristic algorithms are commonly applied to complex combinatorial optimization problems where exact methods are computationally impractical. Tourist route optimization is a representative multi-objective problem characterized by realistic constraints such as travel time, cost, opening hours, and transportation modes. Although Mushroom Reproduction Optimization (MRO) is computationally efficient, it often suffers from premature convergence in complex search spaces. This paper proposes a novel hybrid algorithm, Mushroom–Weed Hybrid Reproduction Optimization (MWHRO), which integrates the colony-based local search of MRO with the fitness-proportional reproduction and competitive elimination mechanisms of Invasive Weed Optimization (IWO). Hybridization enhances population diversity and global exploration while preserving fast convergence. The proposed algorithm is evaluated on a realistic tourist route optimization problem using real-world data from Zagreb, Croatia, across multiple transportation modes and objective-weight scenarios. Performance is compared against Ant Colony Optimization (ACO), IWO, and standard MRO under equal evaluation budgets. Experimental results demonstrate that MWHRO consistently achieves high-quality solutions with significantly lower execution times, particularly in constrained and multimodal scenarios. Statistical analysis confirms the robustness and practical suitability of the proposed approach for real-world route optimization.

Keywords: metaheuristic algorithms; nature-inspired algorithms; ACO; MRO; IWO; hybrid algorithm; multi-objective optimization; route optimization; tourist routing

1. Introduction

Optimization is the process of identifying solutions that best satisfy a predefined objective under a set of constraints. Depending on the nature of the problem, this may involve minimizing a cost or loss function or maximizing a reward or utility function [1]. As the number of decision variables increases, the size of the solution space often grows exponentially due to combinatorial explosions, leading to a dramatic rise in computational complexity [2]. Many such problems belong to the class of NP-hard or NP-complete problems, for which exact solution methods become impractical for large instances. Consequently, a wide range of real-world problems that can be expressed through objective functions and constraints are commonly addressed using optimization techniques rather than exhaustive search methods [3].

Tourist route optimization represents a prominent example of a constrained combinatorial optimization problem. The task involves determining an optimal sequence of visiting attractions

while accounting for multiple, often conflicting objectives such as minimizing travel distance, time, and cost, or maximizing visitor satisfaction. In practical settings, the problem is further complicated by constraints including attraction opening hours, user preferences, transportation modes, traffic conditions, and time budgets. These characteristics make tourist route optimization computationally challenging and well suited for solution approaches based on advanced optimization algorithms.

Metaheuristic algorithms have proven particularly effective in solving such complex optimization problems. Unlike classical deterministic methods, metaheuristics operate at a higher level of abstraction and are designed to explore large solution spaces efficiently without guaranteeing exact optimality. Instead, they aim to identify high-quality solutions that are close to the global optimum within reasonable computational time. Many metaheuristic algorithms are inspired by natural phenomena, such as biological evolution or collective animal behavior, and emphasize a balance between exploration of the search space and exploitation of promising regions. In real-world applications, the ability of metaheuristics to generate a diverse set of feasible solutions is often more valuable than producing a single optimal solution, as it allows flexibility in adapting to changing constraints or preferences [4].

Among population-based metaheuristics, Ant Colony Optimization (ACO) [5], Invasive Weed Optimization (IWO) [6], and Mushroom Reproduction Optimization (MRO) [7] have been successfully applied to routing and combinatorial optimization problems. ACO relies on collective learning and indirect communication to construct solutions iteratively, making it particularly suitable for path-finding tasks. IWO models competitive reproduction and elimination mechanisms, enabling strong global exploration through fitness-proportional population growth. MRO, a more recent fungal-inspired algorithm, emphasizes rapid convergence and computational efficiency through colony-based local search strategies.

Despite its favorable performance characteristics, the standard MRO algorithm exhibits a tendency to prematurely converge local optima, particularly in complex or large-scale problem instances [8]. This limitation reduces its robustness when applied to highly constrained, multi-objective optimization problems such as tourist route planning.

Addressing this shortcoming motivated us to develop a hybrid metaheuristic algorithm that combines complementary search mechanisms from different algorithms. The algorithm integrates the colony-based structure of MRO and the local search efficiency with the IWO reproduction strategy and allows scaling the number of offspring within a fungal colony according to the agent's fitness. The proposed hybrid algorithm is named Mushroom-Weed Hybrid Reproduction Optimization (MWHRO). The design goal was to increase population diversity and strengthen global exploration while maintaining the computational efficiency of the original MRO framework to overcome the stagnation problems of standard MRO and achieve improvements in finding global solutions in state-proportion search and increase robustness.

The primary objective of this research is to evaluate whether the proposed MWHRO algorithm can overcome the stagnation issues of standard MRO and achieve improved convergence behavior, solution quality, and robustness. The algorithm is assessed in the context of tourist route optimization under realistic constraints and compared against MRO, IWO, and ACO. Additionally, the study examines how hybridization affects the exploration–exploitation balance and computational efficiency across problem instances of varying size and complexity.

The main contributions of this paper are as follows:

- A new hybrid metaheuristic algorithm Mushroom-Weed Hybrid Reproduction Optimization (MWHRO) is proposed, which combines the local exploitation efficiency of MRO with the fitness-driven reproduction and elimination mechanisms of IWO to reduce fast convergence,
- The proposed MWHRO algorithm is tested on realistic tourist route optimization problems involving real geographical data, dynamic determination of location distances with respect to different modes of transportation, real costs, and different types of constraints and consistently produces near-optimal or optimal solutions in a short execution time which is particularly important for real-time or interactive routing applications,

- Comprehensive performance comparison of the proposed MWHRO algorithm with ACO, IWO, and standard MRO in terms of solution quality, convergence, execution time, and statistical significance and MWHRO demonstrated high stability, achieving high-quality solutions with significantly lower execution times across all evaluated scenarios,
- A reproducible experimental setup based on real travel time data and multiple optimization scenarios, which enables quantitative evaluation of metaheuristic algorithms in practical routing problems.

The remainder of this paper is organized as follows. Section 2 reviews related work on tourist route optimization and nature-inspired metaheuristic algorithms, with particular emphasis on fungal- and population-based approaches. Section 3 formulates the tourist route optimization problem and describes the objective function and normalization procedure with baseline algorithms ACO, IWO, and MRO algorithms. Section 4 introduces the proposed Mushroom–Weed Hybrid Reproduction Optimization (MWHRO) algorithm in detail. Section 5 describes the experimental setup, datasets, and evaluation methodology. Section 6 reports and discusses the experimental results, including statistical analysis and comparative performance evaluation. Paper ends with a conclusion.

2. Related Work

Tourist route optimization has been extensively investigated as a variant of classical combinatorial optimization problems, most notably the Traveling Salesman Problem (TSP), extended to accommodate real-world constraints and multiple objectives [9]. Unlike the classical TSP, tourism-oriented routing problems typically incorporate factors such as tourist preferences, attraction capacities, congestion levels, contextual information, and time constraints, significantly increasing problem complexity.

A substantial body of research has focused on applying metaheuristic algorithms to tourism route planning. Ant Colony Optimization (ACO) and its variants have been particularly popular due to their effectiveness in path-finding problems. Li et al. (2021) proposed a hybrid ACO approach combined with bacterial foraging and knowledge-based strategies to maximize tourist satisfaction and attraction revenue while considering user attributes and capacity constraints [10]. Liang et al. (2021) further extended ACO by incorporating contextual factors such as weather conditions and crowd density into pheromone updates, improving route comfort, and mitigating premature convergence [11]. Similarly, He (2023) introduced an improved ACO framework for post-epidemic tourism management that accounts for visitor flow and destination carrying capacity, demonstrating enhanced scalability through GPU-based implementation [12].

Beyond ACO-based methods, genetic algorithms and other evolutionary approaches have also been widely applied. Cao (2022) proposed an improved genetic algorithm for round-trip tourism route planning, enhancing crossover operators, adaptive mechanisms, and penalty functions to outperform standard genetic algorithms and neural network models [13].

Yang (2024) presented a preference-aware routing method using weighted directed graphs and multi-criteria decision-making techniques, such as fuzzy analytic hierarchy process, to generate personalized tourist routes efficiently [14]. These studies demonstrate the effectiveness of evolutionary and hybrid approaches in handling multi-objective and constraint-rich tourism optimization problems.

Metaheuristic algorithms have also been successfully applied in related domains such as logistics, routing, and scheduling, where similar combinatorial structures and constraints arise. Xin et al. (2022) addressed logistics distribution optimization using an enhanced genetic algorithm that considers delivery weight, time, customer value, and operational efficiency, achieving significantly faster convergence than traditional approaches [15]. Saleh et al. (2023) proposed a hybrid discrete invasive weed optimization and greedy algorithm for large-scale home care scheduling and routing problems, showing superior performance compared to genetic algorithms and exact mixed-integer linear programming models in complex, constraint-intensive scenarios [16]. These results highlight

the potential of hybrid metaheuristics, particularly those incorporating IWO mechanisms, to improve scalability and solution quality in practical applications.

In recent years, nature-inspired algorithms based on fungal and mushroom behaviors have attracted growing interest due to their distinctive population dynamics and balance between exploration and exploitation. Bidar et al. (2018) introduced Mushroom Reproduction Optimization (MRO), inspired by fungal spore reproduction and colony growth mechanisms. Benchmark evaluations demonstrated that MRO is computationally efficient and competitive with established metaheuristic algorithms across a range of optimization and engineering problems [7]. More recently, Sabet-Rasekh et al. (2025) proposed the bioluminescent fungi optimization algorithm (BFOA), which models fungal spore dispersal through direct release and insect-assisted transport. By dynamically adjusting exploration and exploitation based on fitness and population diversity, BFOA achieved top-ranked performance on benchmark problems and strong results on classical engineering design tasks [17].

Despite these advances, existing fungal-inspired algorithms, including MRO, often face challenges related to premature convergence and limited global exploration, particularly in large-scale or highly constrained problems. While hybrid metaheuristic approaches have shown promise in logistics and routing applications, limited research has explored hybridization strategies that specifically enhance fungal-inspired algorithms for tourist route optimization. In particular, the integration of fitness-proportional reproduction mechanisms, such as those employed in invasive weed optimization, into mushroom-based optimization frameworks remains underexplored.

This research addresses this gap by proposing a novel hybrid algorithm that combines the computational efficiency of MRO with the diversity-enhancing reproduction strategy of IWO.

3. Materials and Methods

3.1. Problem Formulation

The tourist route planning problem is being represented so it can be processed computationally in order to apply nature-inspired optimization algorithms. Every tourist route is modeled as a permutation of the set of chosen tourist locations, with each location appearing exactly once in the sequence, with the order of elements representing the visit order of attractions. For example, for a set of locations {A, B, C, D}, a possible route can be represented as [C, A, D, B], meaning that the location C will be visited first, followed by A, D, and B.

3.2. Input Data

The input data for the algorithm includes information about tourist locations and travel parameters such as:

1. List of locations: predefined n attractions in the selected city (e.g. museums, parks, churches)
2. Location coordinates: latitude and longitude of each location
3. Distance and travel time matrices between each two attractions for the selected mode of transport (e.g. according to the Google Routes)
4. Opening hours of each attraction (start and end of opening hours)
5. Duration of stay at each location (average visit time)
6. Ticket prices and transportation prices, including conditions of use of public transport tickets (ticket validity period)
7. Mode of transport: walking, driving or public transport
8. Determine a fixed starting location

3.3. Objective Function and Normalization

In this task, we employ a multi-criteria objective function, integrating distance between attractions, both travel time and time spent at attractions, total cost, comprising entrance and transportation fees, and penalties for arriving before or after the opening hours of an attraction.

The combined objective function is computed as a weighted sum of all listed components, according to the following expression (1):

$$score = w_d \cdot distance + w_t \cdot time + w_c \cdot cost + penalties \quad (1)$$

where w_d is the weight factor for distance, w_t is the weight factor for time and w_c is the weight factor for cost.

The goal of the optimization algorithm is to minimize this function, i.e., to find a route that achieves the lowest overall score according to the defined criteria and constraints.

To prevent any single component of the objective function (time, cost, or distance) from dominating the optimization process, a normalization procedure is applied prior to aggregation. The normalization ranges are determined automatically through statistical analysis of a set of randomly generated routes. Specifically, the lower and upper bounds (low and high) are defined as the 5th and 95th percentiles of the observed values, respectively. Each component is normalized using the following expression (2):

$$\frac{v - low}{high - low} \quad (2)$$

where v denotes the original value of a given component, and low and $high$ represent the lower and upper normalization bounds.

The final route cost is then computed as a weighted sum of the normalized values for time, cost, and distance, with an additional penalty applied for temporal constraint violations.

3.4. Output of Algorithm

The algorithm returns:

1. Best route (order of attractions)
2. Objective function value (score)
3. Number of evaluations
4. History of best solutions over iterations

3.5. Baseline Algorithms

3.5.1. Invasive Weed Optimization (IWO)

Proposed by Mehrabian and Lucas (2006) [6], Invasive Weed Optimization (IWO) is an algorithm inspired by the spreading behavior and remarkable adaptability of weeds. In nature, weeds are known for their ability to rapidly occupy space, resist adverse conditions, and outperform other plants through high reproductive rates. The theoretical basis of the IWO algorithm simulates these biological principles. Seeds are dispersed and grow into mature plants, where individuals that exhibit better environmental adaptability grow faster and produce a larger number of new seeds. This cycle repeats until the population reaches its natural limit, at which point of competition ensures that only the fittest plants survive and reproduce. By applying Darwin's principle of "survival of the fittest," the algorithm iteratively navigates the search space to find optimal solutions. Based on this biological process, the concept of invasive weed growth can be summarized through the following phases: initialization, reproduction, spatial distribution, and competitive elimination.

The initial population consists of a set of candidate solutions (weeds) that are randomly scattered across the search space. Each seed develops into a plant with a certain fitness value. Accordingly, the number of new plants produced by each plant depends on its fitness value relative to all other plants in the population. The produced seeds are distributed around the parent plant, and this distribution

depends exclusively on a radius around the parent plant, which scales inversely with the number of algorithm iterations. In other words, the spreading radius decreases as the number of iterations increases. As a result, IWO initially favors exploration and gradually shifts its focus toward exploitation of the discovered solutions as the number of iterations increases. When the population reaches a predefined capacity, selection occurs.

3.5.2. Mushroom Reproduction Optimization (MRO)

Mushroom Reproduction Optimization (MRO) is an algorithm inspired by the natural reproduction of mushrooms and their ability to migrate toward more favorable habitats rich in nutrients, moisture, and light. In nature, mushrooms in their reproductive phase produce spores that germinate in favorable environments. Some spores fall near the parent's mushroom, while others are carried by the wind to more distant locations.

MRO models mushroom reproduction to locate optimal solutions in the search space. Each solution represents a mushroom (agent) and new solutions are generated as spores. Mushrooms are divided into colonies that represent regions of high-quality solutions. Artificial wind disperses spores across the search space. Only a small fraction of land in fertile regions, where colonies form with higher probability.

Initially, M parent mushrooms are randomly distributed and they represent colonies. Each parent produces exactly N spores. Within each colony, one mushroom is selected as the new parent and produces new spores that can fall nearby and grow into mature mushrooms if conditions are favorable, enabling local expansion. Local spreading places N spores within a radius r . Parent selection strategies include direct selection of the best mushroom and roulette wheel selection based on fitness. In poor environments, spores remain inactive until carried by wind to new locations. Only the best spores are retained. Inferior colonies are removed, and new colonies form from the best spores.

3.5.3. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO), introduced by Dorigo in 1992, is a population-based metaheuristic inspired by the foraging behavior of real ant colonies. In nature, ants initially explore their environment randomly. When an ant finds food, it deposits pheromones along its path, which can be sensed by other ants. Over time, paths with higher pheromone concentration have become increasingly attractive, leading the colony to collectively converge toward shorter and more efficient routes. This process is governed by positive feedback and indirect communication, known as stigmergy. Due to its suitability for routing and path-planning problems, ACO is commonly applied to combinatorial optimization tasks such as tourist route optimization [18].

ACO is based on indirect communication between agents (ants). The general procedure consists of three main stages: initialization, solution construction, and pheromone updating. Initially, pheromone values are set, and a population of ants is placed randomly in the search space. Each ant then constructs a solution by moving through the problem graph according to probabilistic rules influenced by pheromones and heuristic visibility. After all ants complete their tours, pheromone levels are updated through evaporation and reinforcement, emphasizing high-quality solutions. This process is repeated until the stopping criterion is satisfied [18].

4. Mushroom-Weed Hybrid Reproduction Optimization (MWHRO)

4.1. Fundamental Similarities and Differences Between the MRO and IWO Algorithms

When we compare the initialization of agents in the MRO and IWO algorithms, we can see that the algorithms are most similar in this step. Both algorithms release a predefined number of agents into the problem space. In the case of MRO, these are referred to as "parent mushrooms," while in IWO they are referred to as "weed seeds."

In the MRO algorithm, initialization is followed by the step of adding spores to the initial parent mushrooms. In the IWO algorithm, the process begins by adding seeds that later grow into mature plants. In both cases, initialization defines the initial population of agents in the problem space.

The first significant difference becomes apparent in reproduction. In MRO, only one spore per parent mushroom, out of a predefined number of generated spores, is allowed to pass into the next generation. Additionally, for each colony, only one mushroom is selected as the parent, based on the value of the fitness function that corresponds to the solution represented by that position.

In the IWO algorithm, reproduction is different— all plants are allowed to release seeds, with the number of seeds being proportional to the value of the fitness function. Plants with higher fitness values produce a larger number of seeds. In both cases, reproduction is closely linked to the fitness function value of each agent.

MRO introduces wind as a separate mechanism for spore transport in the problem space, or in other words, as a global search mechanism. In this way, MRO favors exploitation, while exploration depends on the wind factor and the number of unfavorable spores.

On the other hand, although IWO mentions that seeds are dispersed by wind, the spatial distribution mechanism does not take wind as a factor. Spatial distribution depends exclusively on the radius around the parent plant, which scales inversely with the number of algorithm iterations. In other words, the spreading radius decreases as the number of iterations increases. As a result, IWO initially favors exploration and gradually shifts its focus toward exploitation of the discovered solutions as the number of iterations increases.

Since MRO takes only one new spore per colony during each reproduction, the growth in the number of agents is linear with respect to the number of initial colonies. Because of this, no specific mechanism for eliminating existing agents is defined, as it is assumed that the algorithm will converge near the optimal solution long before the entire problem space becomes populated with agents.

In the IWO algorithm, each agent is allowed to create new agents in its neighborhood, which shrinks over time. The growth in the number of agents thus follows a natural logarithmic curve. To prevent saturation of the problem of space with agents, a mechanism of competitive elimination is introduced, which removes from the overall population those agents whose fitness values are low relative to the values of other agents.

4.2. Modification of Reproduction Modeled After the IWO Algorithm

The basic idea behind modifying the MRO algorithm in the spirit of the IWO algorithm arises from the fundamental difference in reproduction between these two algorithms. Specifically, MRO favors exploitation of local solutions and, as such, has a greater tendency to become “stuck” in a local optimum. If the wind factor is not favorable for global search, the potential for finding the global optimum in the presence of multiple local optima is reduced.

For this reason, in proposed MWHRO algorithm we modify the reproduction mechanism of the MRO algorithm by allowing a larger number of mushrooms to release spores. In addition, we scale the number of spores accepted into the next generation according to the fitness function value of each agent.

From an implementation perspective, we use an approach similar to probabilistic parent selection. For each colony, we define the three best mushrooms—the agents with the highest fitness function values—and scale the number of spores they release according to their fitness values.

The procedure in proposed MWHRO is described with the following pseudocode:

```

For each colony in the list of colonies:
  Define a list of fitness values
  For each mushroom in the colony
    Add the mushroom's fitness function value to the list of values
  Find the three maximum values in the list and return the list of
  parent mushrooms
  For each mushroom in the list of parents:
    If the mushroom is in first place:
      Create three new spores
    If the mushroom is in second place:
      Create two new spores
    If the mushroom is in third place:
      Create one new spore

```

4.3. Competitive Elimination Modeled After the IWO Algorithm

Additionally, we introduce an elimination mechanism, modeled after the IWO algorithm. The reason for this is the potential tendency of the MRO algorithm to favor exploitation over exploration. By increasing the number of agents during reproduction, we may increase the number of unfavorable solutions and thus cause the algorithm to become “stuck” in local optima.

The elimination mechanism we introduce therefore removes the same number of agents from the entire problem space for each increase in the number of agents during reproduction. Agents are selected for elimination based on the value of the fitness function. All agents whose fitness function value falls below a certain threshold are eliminated during each iteration of the algorithm. In this way, we ensure that both the time required to execute the algorithm and the computational resources used remain within relatively acceptable limits.

5. Experimental Setup

The experiments were designed to evaluate and compare the performance of the IWO, MRO, ACO and proposed MWHRO algorithm in the context of tourist route optimization. The data consists of detailed information about tourist attractions and relevant travel parameters, which were used to model and solve the tourist route optimization problem.

5.1. Parameter Initialization, Data Collection and Preprocessing

The dataset was constructed to realistically represent urban tourism conditions in the city of Zagreb in Croatia. A predefined set of 15 tourist attractions was selected (Figure 1):

```

LOCATIONS = [
  "Trg bana Josipa Jelačića, Zagreb",
  "Maksimir park, Zagreb",
  "Crkva sv. Marka, Zagreb",
  "Zagrebačka katedrala, Zagreb",
  "Tržnica Dolac, Zagreb",
  "Park Zrinjevac, Zagreb",
  "Hrvatsko narodno kazalište, Zagreb",
  "Muzej iluzija, Zagreb",
  "Muzej prekinutih veza, Zagreb",
  "Tehnički muzej Nikola Tesla, Zagreb",
  "Muzej suvremene umjetnosti, Zagreb",
  "Kula Lotrščak, Zagreb",
  "Strossmayerovo šetalište, Zagreb",
  "Jarun jezero, Zagreb",
  "Trg kralja Tomislava, Zagreb",
]

```

Figure 1. List of attraction locations in Zagreb for which the tourist route is optimized.

The starting location of each route was fixed to the Ban Josip Jelačić Square, Zagreb, serving as the departure point of all generated routes. All locations are represented in Figure 2. Each attraction is marked by its geographical coordinates (longitude and latitude).

Attractions are further divided into predefined types of attractions: museums, churches, parks, and general tourist attractions (other).

For each type of attraction, the duration of stay, the entrance fee and opening hours are predefined according to data that are usually valid in Zagreb (Table 1). If data on opening hours, working hours, tickets are available for an attraction, the predefined data from Table 1 are replaced with actual data.

Table 1. Predefined data for each type of attraction.

Attraction Type	Stay time (hours)	Price (EUR)	Opening days	Opening hours
Museum	1.5	5	Mon-Fri	9-16
Church	0.3	0	All	7-19
Park	0.5	0	All	0-24
Other	1	2		

Additionally, three modes of travel were considered for visiting attractions: walking (mark as **Walk**), using a car or personal vehicle (mark as **Drive**), and using public transport (mark as **Public**), which allows for a comparative assessment of routes in different mobility scenarios.

In the case of car travel, the cost is calculated based on the distance traveled and the price per kilometer (set at 0.08 €/km in experiment). For public transport, a fixed cost based on the ticket validity periods and fare policies is considered. In this experiment, the price is set to a one-way tram ticket of 0.53 € for every 30 minutes of uninterrupted travel. For walking, the cost of transportation is not calculated.

For the selected mode of transportation, two matrices are created, being a distance matrix and a time matrix for every pair of attractions. The matrices were obtained using Google Routes API [19], ensuring realistic travel estimates that account for actual road networks and transportation conditions (Figure 2).

```

"DRIVE": {
  "5_7": {
    "distance": 1.592,
    "duration": 6.416666666666667
  },
  "8_2": {
    "distance": 0.79,
    "duration": 2.5833333333333335
  },
  "6_9": {
    "distance": 0.986,
    "duration": 4.1833333333333334
  },
}

```

Figure 2. Example of distance and time matrix for Drive travel mode.

All data in this study were collected on January 22, 2026, at 09:40 AM. This timestamp defines the reference state of the transportation network and travel conditions under which experiments were conducted.

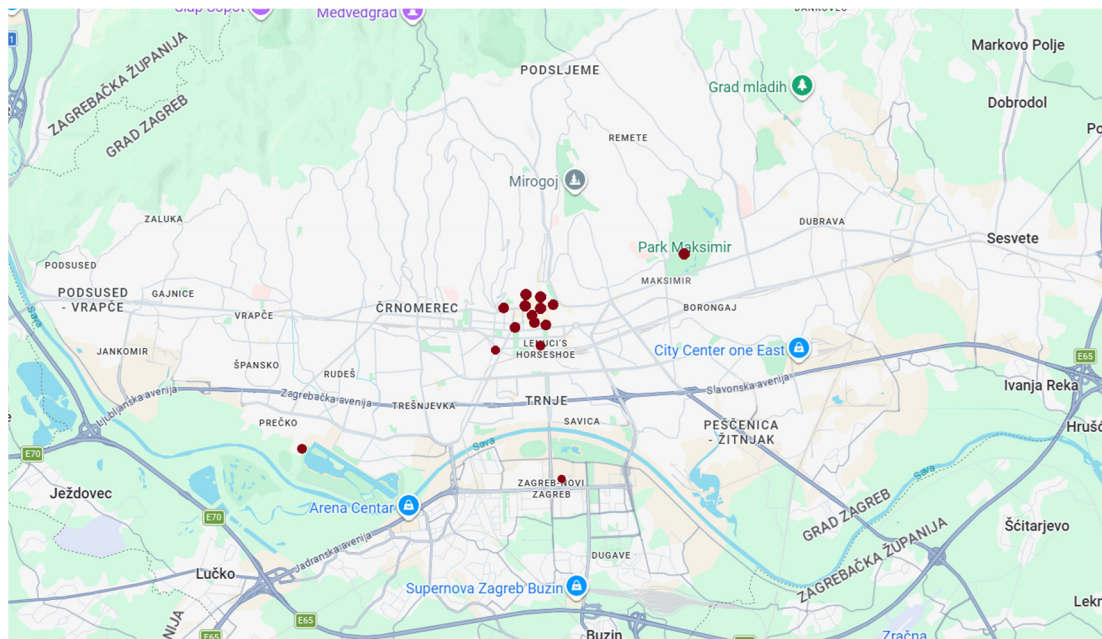


Figure 3. Map of Zagreb, Croatia with marked locations to be visited during single tour.

The effectiveness of the algorithms was analyzed under two different weight coefficient scenarios for the objective function: balanced weights and scenarios favoring exclusively distance (Table 2).

Table 2. Weights values for each scenario: Balanced and Distance.

Scenario	wd (Distance)	wt (Time)	wc (Cost)	pen	Focus
Balanced	0.33	0.33	0.33	0.05	Multi-criteria optimization
Distance-oriented	1.0	0.0	0.0	0.05	Shortest path optimization

In all scenarios, the penalty coefficient was fixed to $pen = 0.05$ to consistently penalize violations of constraints such as visiting attractions outside their opening hours. To ensure an objective and fair evaluation, each algorithm was executed 30 independent times per scenario, with a fixed budget of 10000 objective function evaluations per run. Each algorithm was executed using two population sizes, 5 and 20 agents, to evaluate performance under both resource-constrained and more exploratory settings. To ensure statistical reliability, each configuration was independently executed 30 times.

5.2. Execution Procedure

The following procedure was applied in experimental evaluation for each scenario:

1. **Running all algorithms** (IWO, MRO, ACO, MWHRO) with predefined parameters (maximum number of evaluations, seed number and history tracking)
2. **Saving and evaluating the results for each run:**
 - a. Final score (objective function value achieved at the end of execution)
 - b. Execution time to find a solution (in seconds)
 - c. Number of evaluations
 - d. Convergence of results (series of best objective function values over time)
3. **Re-running the algorithms** multiple times for stability analysis (without fixed seed)
4. **Comparison of results** obtained by the used algorithms (IWO, MRO, ACO, MWHRO)
5. **Statistical analysis** and assessment of the significance of the difference in the obtained solutions

5.3. Performance Evaluation

The comparison of the algorithms is evaluated using the final objective function value (score), where lower values (minimum) indicate better solutions. For each experiment, both the best (minimum) value obtained after all evaluations and the average score across the 30 independent runs are calculated along with the descriptive statistics (mean and standard deviation).

In addition to the quality of the solution, computational efficiency (runtime) is measured as the average time measured in seconds required for each algorithm to obtain a solution. Also, convergence behavior is analyzed through convergence plots, which visually illustrate how the best-found solution evolves with respect to the number of evaluations, thereby indicating the convergence speed of each algorithm.

To assess the statistical significance of the observed performance differences, the Wilcoxon signed-rank test [20] was applied at a significance level of 0.05. Effect sizes of differences in results were quantified using Cliff's delta [21]. Spearman's rank correlation coefficient [22] was used to assess the similarity of attraction orderings in routes generated by selected algorithms.

These non-parametric methods were selected due to the stochastic nature of the evaluated algorithms and the absence of assumptions regarding normality.

5.4. Implementation of Optimization Algorithms

All four optimization algorithms, IWO, MRO, ACO, and proposed MWHRO were implemented in Python and adapted to solve the tourist route optimization problem. Each algorithm operates on permutations of attractions and evaluates candidate routes using the objective function described in the previous section. To ensure fair comparison, all implementations support different transportation modes (walk, drive, public) and fixed limits on the number of objective function evaluations (10 000).

The IWO algorithm captures the main components of the method: population initialization, seed reproduction, and spatial distribution (mutation), and competitive elimination based on fitness values. The algorithm operates with a decreasing mutation range controlled by a standard deviation parameter ($\sigma_{start} = 0.5, \sigma_{end} = 0.1$). Each individual produces between 3 and 8 offspring, proportional to its fitness. At each iteration, offspring is generated via mutation of parent solutions, evaluated, and merged with the current population. Only the best solutions are retained, ensuring competitive elimination and gradual improvement.

The MRO algorithm was implemented to model the reproduction behavior of mushrooms through local spore generation and competitive selection. The method operates with a population of individuals, where each individual represents a candidate tourist route. In each iteration, every colony selects its best-performing individual as a parent and generates a fixed number (5) of offspring (spores) through local mutations implemented as swaps between two attractions. The offspring is evaluated and merged with the current population, after which only the best solutions are retained. This elitist selection mechanism ensures controlled population size, fast convergence, and computational efficiency while maintaining effective local exploration of the solution space.

The proposed MWHRO algorithm extends the standard MRO approach by introducing a fitness-proportional reproduction strategy to enhance diversity and exploration. Instead of selecting only the single best individual per colony, multiple top-ranked individuals (five) are chosen as parents, with each producing a different number of offspring proportional to its fitness. Specifically, the best-ranked individual generates five spores, while progressively lower-ranked individuals generate fewer spores, down to one spore for the weakest among the selected parents. The offspring are generated via local mutations, evaluated, and combined with the existing population, after which only the best solutions are preserved. This hybrid strategy improves the balance between exploration and exploitation and reduces the risk of premature convergence, leading to more robust search behavior in complex optimization landscapes.

The ACO algorithm includes pheromone matrix initialization, probabilistic construction of routes by ants, solution evaluation, and pheromone updating. Each ant incrementally builds a route based on pheromone information (τ), which represents collective experience, and heuristic visibility

(η), defined as the inverse of a combination of normalized distance and travel time. An elitist pheromone update strategy is employed in which only the best ant of the current iteration and the globally best solution deposit additional pheromones, while existing pheromones evaporate. Compared to standard ACO, this approach accelerates convergence, reduces noise from low-quality solutions, and improves search efficiency.

The algorithm is executed with a predefined number of ants, while its behavior is controlled by several parameters that regulate the balance between exploration and exploitation. The parameter $\alpha = 0.5$ determines the influence of pheromone trails, $\beta = 2.0$ controls the importance of heuristic information, $\rho = 0.1$ governs pheromone evaporation, and $Q = 1.0$ regulates the amount of pheromone deposited by the best-performing ant.

Functions of all algorithms return to the best route found, its score, the number of evaluations, and optionally the convergence history.

6. Results and Discussion

For each scenario, we are looking for the best minimum score. The experimental results include the average score, standard deviation, and best (minimum) score across 30 runs and average execution time are given in Tables 3–10, along to an assessment of statistical significance.

Table 3. Average execution time (in seconds) across different transport types in **Balanced scenario** (10000 evaluations, 5 agents). Best results are marked in bold.

Algorithm	Drive (s)	Public (s)	Walk (s)
ACO	2.37 ± 0.3	2.27 ± 0.2	2.34 ± 0.45
MRO	1.03 ± 0.44	0.85 ± 0.19	0.82 ± 0.22
MWHRO (ours)	0.79 ± 0.18	0.75 ± 0.14	0.66 ± 0.12
IWO	3.57 ± 1.43	2.6 ± 0.9	2.72 ± 1.05

Table 4. Average execution time (in seconds) across different transport types in a **Distance-oriented scenario** (10000 evaluations, 5 agents). Best results are marked in bold.

Algorithm	Drive (s)	Public (s)	Walk (s)
ACO	2.26 ± 0.2	2.31 ± 0.26	2.19 ± 0.17
MRO	0.86 ± 0.25	0.78 ± 0.08	0.82 ± 0.21
MWHRO (ours)	0.74 ± 0.08	0.72 ± 0.12	0.69 ± 0.14
IWO	3.09 ± 1.27	3.13 ± 1.28	2.72 ± 1.07

Table 5. Performance comparison in the **Public distance-oriented scenario** (10000 evaluations, 5 agents). Best results are marked in bold.

Algorithm	Avg. score	Std. dev score	Min score	Avg. time (s)
ACO	-0.672	0.0374	-0.7512	2.3142
MRO	-0.6663	0.0704	-0.7606	0.7791
MWHRO (ours)	-0.7009	0.0486	-0.7606	0.7185
IWO	-0.5829	0.112	-0.7606	3.132

Table 6. Performance comparison in the **Drive distance-oriented scenario** (10000 evaluations, 5 agents). Best performances are marked in bold, second best in italic.

Algorithm	Avg. score	Std. dev score	Min score	Avg. time (s)
ACO	-1.1173	0.006	-1.1227	2.2594
MRO	-1.0779	0.0358	-1.1227	0.8621
MWHRO (ours)	<i>-1.0945</i>	<i>0.0208</i>	-1.1227	0.7418
IWO	-1.0083	0.081	-1.1185	3.0871

Table 7. Statistical comparison of solution quality using the Wilcoxon signed-rank test ($\alpha = 0.05$, 10000 evaluations, 5 agents). Statistically significant difference with medium and large effect for MWHRO are marked in bold.

Comparison	Scenario	p-value	Cliff's δ	Effect
MWHRO vs ACO	Public – Balanced	< 0.001	-0.868	Large
MWHRO vs MRO	Public – Balanced	< 0.01	-0.386	Medium
MWHRO vs IWO	Public – Balanced	< 0.01	-0.462	Medium
MWHRO vs ACO	Drive – Distance	< 0.001	0.610	Large
MWHRO vs MRO	Drive – Distance	0.056	-0.221	Small (ns)
MWHRO vs IWO	Drive – Distance	< 0.001	-0.759	Large

Table 8. Performance comparison across **Balanced** scenarios for different population sizes (10,000 evaluations). Best performances are marked in bold.

Transport mean	Algorithm	Avg Score (5 agents)	Avg Score (20 agents)	Avg Time (5 agents)	Avg Time (20 agents)
Public	MWHRO (ours)	-0.7770	-0.7787	0.745 s	0.445 s
	MRO	-0.7238	-0.8035	0.855 s	0.540 s
	ACO	-0.6896	-0.7040	2.265 s	2.144 s
	IWO	-0.6740	-0.7571	2.600 s	0.356 s
Drive	MWHRO (ours)	-1.1118	-1.1168	0.791 s	0.452 s
	MRO	-1.0897	-1.1222	1.029 s	0.500 s
	ACO	-1.1448	-1.1427	2.370 s	2.155 s
	IWO	-1.0450	-1.0677	3.568 s	0.361 s
Walk	MWHRO (ours)	0.1708	0.1393	0.665 s	0.416 s
	MRO	0.1636	0.1294	0.820 s	0.498 s
	ACO	0.1564	0.1583	2.338 s	2.072 s
	IWO	0.2734	0.2185	2.717 s	0.320 s

Table 9. Spearman's rank correlation between the best routes obtained by MWHRO and competing algorithms (**Public balanced** scenario, 10000 evaluations, 5 agents).

Comparison	Spearman (best route)
MWHRO vs ACO	0.754
MWHRO vs MRO	1
MWHRO vs IWO	1

Table 10. Representative final routes for the **Public balanced scenario** (10000 evaluations, 5 agents). The attractions according to which the routes differ are marked with bold.

Algorithm	Final route (order of attractions)
ACO	Ban Josip Jelačić Square → Dolac Market → Zagreb Cathedral → Zrinjevac Park → Museum of Illusions → Croatian National Theatre → Nikola Tesla Technical Museum → Jarun Lake → Museum of Contemporary Art → King Tomislav Square → Lotrščak Tower → Museum of Broken Relationships → St. Mark's Church → Strossmayer Promenade → Maksimir Park
MRO	Ban Josip Jelačić Square → Zagreb Cathedral → Dolac Market → Museum of Illusions → Croatian National Theatre → Nikola Tesla Technical Museum → Jarun Lake → Museum of Contemporary Art → King Tomislav Square → Zrinjevac Park → Lotrščak Tower → Museum of Broken Relationships → St. Mark's Church → Strossmayer Promenade → Maksimir Park
MWHRO (ours)	Ban Josip Jelačić Square → Zagreb Cathedral → Dolac Market → Museum of Illusions → Croatian National Theatre → Nikola Tesla Technical Museum → Jarun Lake → Museum of Contemporary Art → King Tomislav Square → Zrinjevac Park → Lotrščak Tower → Museum of Broken Relationships → St. Mark's Church → Strossmayer Promenade → Maksimir Park

IWO	Ban Josip Jelačić Square → <i>Zagreb Cathedral</i> → <i>Dolac Market</i> → Museum of Illusions → Croatian National Theatre → Nikola Tesla Technical Museum → Jarun Lake → Museum of Contemporary Art → <i>King Tomislav Square</i> → <i>Zrinjevac Park</i> → Lotrščak Tower → Museum of Broken Relationships → St. Mark's Church → Strossmayer Promenade → Maksimir Park
-----	--

6.1. Execution Time Comparison

Table 3 compares the average execution time for balanced scenarios across Drive, Public, and Walk means of transport. Values are reported as mean \pm standard deviation over 30 independent runs.

The results clearly indicate that MWHRO is the fastest algorithm in all three types of transport, achieving average execution times of 0.79 s, 0.75 s, and 0.66 s for Drive, Public transport, and Walk scenarios, respectively. In comparison, ACO requires approximately three times longer execution time, while IWO exhibits both the highest runtime and the largest variability. These findings confirm that the hybrid design of MWHRO enables rapid convergence without introducing additional computational overhead, which is particularly important for real-time or interactive routing applications.

Table 4 presents the average execution time for the distance-oriented scenario. The observed trend is consistent with the balanced setting reported in Table 3, as MWHRO again achieves the lowest runtime for all transport modes (0.74 s for Drive, 0.72 s for Public transport, and 0.69 s for Walk). ACO remains substantially slower, while IWO exhibits the highest execution time and variability. This consistency across both balanced and distance-oriented scenarios indicates that the proposed MWHRO approach maintains stable and efficient computational performance regardless of the adopted optimization objective, which is particularly important for real-time and interactive routing applications.

6.2. Solution Quality Comparison

The quality of the selected algorithms is quantitatively compared to proposed MWHRO within the distance-oriented scenario. The results for Public transport which represents the most complex and limited routing setting are shown in Table 5. In this scenario, MWHRO achieves the best average objective value (-0.7009), outperforming all other algorithms, followed by MRO, ACO, and the worst is IWO (-0.5829). All algorithms except ACO achieve the best minimum solution (-0.7606) so they do not differ in the minimum score, but MWHRO is the fastest (0.7185 s), significantly faster than ACO (>2.3 s) and IWO (>3.1 s). This demonstrates that MWHRO provides superior overall performance in public transport routing, because it achieves the lowest average score with the shortest average execution time, with a reasonable standard deviation.

Additionally, for a more detailed performance comparison, a convergence plot and dispersion of the results are shown in Figure 4. The convergence curves show a steep improvement of the score in the early evaluations for all methods, with ACO and MRO reaching their plateau faster. In contrast, MWHRO improves more gradually but continues refining the solution and achieves the lowest final score. The dispersion plot further shows that MWHRO and ACO achieve more consistent final scores, whereas IWO exhibits the largest variability.

The conclusions regarding the algorithm with the best performance in the Public distance-oriented scenario were not equally valid in the **Drive** distance-oriented scenario (Table 6). In this scenario, ACO achieves the best average score (-1.1173) and MWHRO is the second best (1.0945) but with the shortest runtime of 0.7418, which is significantly shorter than ACO and IWO. All algorithms except IWO are able to identify the same global minimum (-1.1227).

This result highlights the key trade-off regarding MWHRO. Although there are cases where it does not achieve the absolute best performance, it stands out with relatively good results (second best) and high speed.

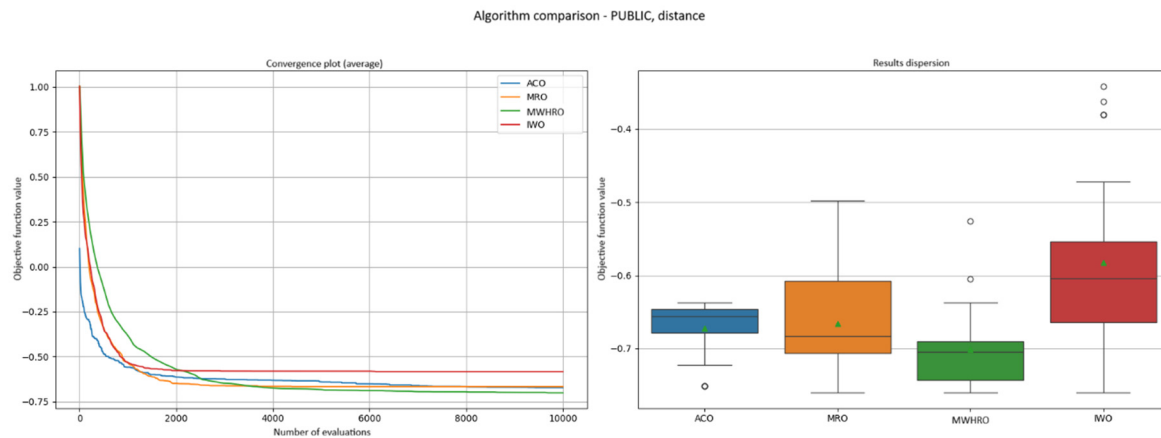


Figure 4. Convergence plot and results of dispersion for all tested algorithms, **Public distance-oriented scenario** (10000 evaluations, 5 agents).

6.3. Assessment of Statistical Significance

The statistical significance of the observed differences in solution quality with respect to p-value (significance) and Cliff's δ (effect size, the sign indicates which is better when minimized) is given in Table 7. Statistical test results are reported alongside the numerical performance values, ensuring that the observed differences are not attributable to random variation.

For the balanced **Public** transport scenario, MWHRO is statistically the best of all three competing algorithms, with a medium negative effect in the case of MRO ($\delta = -0.386$) and IWO (-0.462) to a large effect for ACO ($\delta = -0.868$) according to Cliff's delta, confirming that the observed improvements are systematic and not random.

In contrast, in the **Drive** distance scenario, the ACO algorithm achieves a statistically significantly better average solution quality than MWHRO with a large positive effect ($\delta = 0.610$), while MWHRO is clearly better than IWO with a large negative effect (-0.759). MWHRO has a small negative effect with respect to MRO ($\delta = -0.221$) but the difference is not statistically significant ($p > 0.05$).

6.4. Comparison of Sensitivity to Population Size

We have also analyzed the sensitivity of algorithms to population size (from 5 to 20). Table 8 shows the impact of population size on solution quality and computational efficiency across the balanced scenario of the **Public**, **Drive**, and **Walk** means of transport, while maintaining a constant of 10,000 evaluations.

Analysis of the data reveals that increasing the population from 5 to 20 agents generally improves the quality of the solution with the most notable gains observed for the IWO and MRO algorithms in the Public transport scenario. For all algorithms and scenarios, the convergence time with 20 agents is shorter than with 5 agents. For example, IWO's runtime decreases from 2.600 s to 0.356 s in the Public transport scenario when shifting from 5 to 20 agents. This trend can be attributed to the fixed evaluation budget, where larger populations require fewer iterations to reach the maximum number of evaluations, thereby reducing the computational overhead of the algorithms' internal control structures.

Across all configurations, MWHRO remains the most computationally efficient algorithm, consistently delivering competitive or superior scores within the shortest execution times, particularly in complex multimodal settings suggesting a high degree of robustness in its search mechanisms.

6.5. Comparison of Outputs (Best Route)

Tables 9 and 10 jointly analyze the similarity and structure of the final routes obtained for the Public balanced scenario. Table 9 reports Spearman's rank correlation between the best routes produced by MWHRO and the competing algorithms, showing perfect agreement with MRO and IWO ($\rho = 1$) and a high correlation with ACO ($\rho = 0.754$). This quantitative evidence is consistent with the representative routes presented in Table 10, where MRO, MWHRO and IWO generate identical visitation orders, while the ACO solution differs only by two minor local permutations involving the pairs Dolac Market–Zagreb Cathedral and Zrinjevac Park–King Tomislav Square. Overall, the results confirm that all algorithms converge to highly similar route structures and that the observed differences are limited to local reordering of spatially close attractions, preserving the same global organization of the tour.

Taken together, the results suggest that no single algorithm dominates across all metrics and scenarios. However, the proposed MWHRO remained highly competitive, often performing best or trailing by a negligible margin while providing significantly faster execution. ACO tends to perform well in continuous and less constrained environments, whereas MWHRO excels in scenarios that require both efficient exploration and rapid convergence. Notably, MWHRO demonstrates high stability, consistently achieving near-optimal or optimal solutions while maintaining the lowest execution times across all evaluated scenarios.

These findings underline the practical relevance of the proposed hybrid approach. By combining multiple nature-inspired mechanisms, MWHRO effectively balances solution quality and computational efficiency, making it particularly suitable for real-world tourist route optimization systems where both performance and responsiveness are critical.

7. Conclusion

This paper introduces the Mushroom-Weed Hybrid Reproduction Optimization (MWHRO) algorithm. The method combines the local search strength of Mushroom Reproduction Optimization (MRO) with the fitness-based reproduction and elimination mechanisms of Invasive Weed Optimization (IWO). The main goal was to reduce premature convergence in MRO and improve performance in complex combinatorial problems.

The algorithm was evaluated on a realistic tourist route optimization problem using real data from Zagreb, Croatia, with different transportation modes and objective-weight scenarios being tested. The results show that MWHRO consistently produces high-quality solutions, and in most scenarios, it also requires less execution time than the compared algorithms. Statistical analysis confirms that in constrained and multimodal cases, especially public transport routing, MWHRO significantly outperforms MRO and IWO. In many cases, it also performs better than ACO.

ACO achieved strong results in some distance-oriented driving scenarios. However, it required considerably more computational time compared to MWHRO. In contrast, MWHRO offers a better balance between solution quality and efficiency, which makes it suitable for practical tourist route planning systems.

Future work will include problems with larger data sets (attractions) and include dynamic traffic conditions. Further improvements in parameter tuning to increase robustness and scalability, and additional optimization scenarios that favor time or cost, will be explored.

Author Contributions: Conceptualization, D.P., R.A. and M.I.K.; methodology, D.P., R.A. and M.I.K; software, D.P and R.A.; validation, M.I.K.; formal analysis, D.P., R.A. and M.I.K.; investigation, D.P. and R.A.; resources, D.P. and R.A.; data curation, D.P. and R.A.; writing—original draft preparation, D.P. and R.A.; writing—review and editing, M.I.K.; visualization, D.P. and R.A.; supervision, M.I.K. and J.L.; project administration, M.I.K.; funding acquisition, M.I.K and J.L.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by HRZZ, grant number 410-25-2 and UNIRI project (grant: uniri-iz-25-103).

Data Availability Statement: The data can be made available upon reasonable request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Optimization problem. Available online: https://en.wikipedia.org/wiki/Optimization_problem (accessed on 30. 1. 2026.).
2. Combinatorial explosion. Available online: https://en.wikipedia.org/wiki/Combinatorial_explosion (accessed on 30.1.2026.).
3. Combinatorial optimization. Available online: https://en.wikipedia.org/wiki/Combinatorial_optimization (accessed on 30.1.2026.).
4. Metaheuristic. Available online: <https://en.wikipedia.org/wiki/Metaheuristic> (accessed on 30.1.2026.).
5. Dorigo, M.; Stützle, T. *Ant Colony Optimization*; The MIT Press: Massachusetts, USA, 2004.
6. Mehrabian, A. R.; Lucas, C. A novel numerical optimization algorithm inspired from weed colonization; *Ecological Informatics* **2006**, *1*, 355-366.
7. Bidar, M.; Kanan, H. R.; Mouhoub, M.; Sadaoui, S. Mushroom Reproduction Optimization (MRO): a novel nature-inspired evolutionary algorithm; *2018 IEEE Congress on Evolutionary Computation (CEC)* **2018**, 1-10.
8. Lacković, R. Optimizacija turističke rute korištenjem prirodom inspiriranih algoritama. Graduate thesis, University of Rijeka, Croatia, September 2025.
9. Ruiz-Meza, J.; Montoya-Torres, J. R. A systematic literature review for the tourist trip design problem: Extensions, solution techniques and future research lines. *Operations Research Perspectives* **2022**, *9*, 100228.
10. Li, S.; Luo, T.; Wang, L.; Xing, L.; Ren, T. Tourism route optimization based on improved knowledge ant colony algorithm. *Complex & Intelligent Systems* **2022**, *8*, 3973-3988.
11. Liang, S.; Jiao, T.; Du, W.; Qu, S. An improved ant colony optimization algorithm based on context for tourism route planning. *PLoS One* **2021**, *16*, e0257317.
12. He, S. A novel travel route planning method based on an ant colony optimization algorithm. *Open Geosciences* **2023**, *15*, 20220541.
13. Cao, S. An Optimal Round-Trip Route Planning Method for Tourism Based on Improved Genetic Algorithm. *Computational Intelligence and Neuroscience* **2022**, *2022*, 7665874.
14. Yang, Y. An optimization method for tour routes of characteristic tourist attractions considering tourists preferences. *Proceedings of the 2024 International Conference on Computer and Multimedia Technology* **2024**, 447-451.
15. Xin, L.; Xu, P.; Manyi, G. Logistics distribution route optimization based on genetic algorithm. *Computational Intelligence and Neuroscience* **2022**, *2022*, 8468438.
16. Saleh, M. B.; Grunder, O.; El Hassani, A. H. Discrete invasive weed optimization and greedy hybridization algorithm for home care multi-days assignment scheduling and routing problems. *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)* **2023**, 638-643.
17. Sabet-Rasekh, S.; Beheshti, Z. Bioluminescent fungi optimization algorithm and its applications to solve engineering optimization problems. *Iran Journal Computer Science* **2025**, *8*, 2307-2343.
18. Gandomi, A. H.; Yang, X.-S.; Talatahari, S.; Alavi, A. H. Metaheuristic Algorithms in Modeling and Optimization. In *Metaheuristic Applications in Structures and Infrastructures*; Gandomi, A. H.; Yang, X.-S.; Talatahari, S.; Alavi, A. H., Eds.; Elsevier, Waltham, MA, USA, 2013; pp. 1-24.
19. Routes API. Available online: <https://developers.google.com/maps/documentation/routes/reference/rest> (accessed on 29. 1. 2026.).
20. Wilcoxon signed-rank test. Available online: https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test (accessed on 30. 1. 2026.).
21. Cliff's Delta Effect Size. Available online: https://tkimhofer.github.io/metabom8/reference/es_cdelta.html (accessed on 30. 1. 2026.).
22. Spearman's rank correlation coefficient. Available online: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient (accessed on 30. 1. 2026.).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.