Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Beyond Pheromones: Exploiting Edge Frequency and Quality for Intelligent TSP Optimization

Adel Saha
Ferhat Abbas University
Fouzi Semchedine
Ferhat Abbas University
Mira Lefkir
Mohamed El Bachir El Ibrahimi University
Abdelouahab Attia
abdelouahab.attia@univ-bba.dz

Mohamed El Bachir El Ibrahimi University

Research Article

Keywords: Traveling Salesman Problem (TSP), Ant Colony Optimization (ACO), 2OPT heuristic, heuristics, metaheuristics, hybridization, BEFRA, BEQRA

Posted Date: August 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10657441/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Beyond Pheromones: Exploiting Edge Frequency and Quality for Intelligent TSP Optimization

Adel Saha[1], Fouzi Semchedine[1,2], Mira Lefkir[3,4], Abdelouahab Attia[3,4*]

[1]Departement of computer science, Ferhat Abbas University Setif 1, Setif, 19000, Algeria.
[2]Mechatronics Laboratory(LMETR), Optics and Precision Mechanics Institute, University of Sétif 1, Setif, 19000, Algeria.
[3]Department of Computer Science, Mohamed El Bachir El Ibrahimi University, Bordj BouArreridj 34000, Algeria.
[4]Intelligent Systems and Cognitive Computing Laboratory (ISCC), Bordj Bou Arreridj, 34000, Algeria.

*Corresponding author(s). E-mail(s): [abdelouahab.attia@univ-bba.dz];
Contributing authors: [adel.saha@univ-bba.dz];
[fouzi.semcheddine@univ-setif.dz]; [mira.lefkir@univ-bba.dz];

Abstract

In order to improve Ant Colony Optimization's (ACO) performance in solving the Symmetric Traveling Salesman Problem (TSP), this paper suggests four new heuristic techniques. The suggested techniques take advantage of edge frequency and edge quality, two underutilized information sources in population-based algorithms. The quality and frequency of edges in population-based TSP solutions have not yet been thoroughly studied. Prior research has not been able to establish a direct connection between these two elements and how they affect the improvement of solutions. In this regard, the current study investigates the impact of integrating edge frequency and edge quality into the process of building solutions for the first time. BEFRA (Based Edge Frequency Ranking Algorithm) and BEQRA (Based Edge Quality Ranking Algorithm) are two useful heuristics that are presented. Additionally, a local search is performed. Extensive experiments conducted on standard TSPLIB benchmarks demonstrate that the proposed hybrid approaches significantly outperform classical ACO and several well-known heuristics and metaheuristics in terms of solution quality and robustness. The findings provide new insights into how edge-level knowledge can be

effectively exploited, and they make an important contribution to combinatorial optimization problem solving.

Keywords: Traveling Salesman Problem (TSP), Ant Colony Optimization (ACO), 2OPT heuristic, heuristics, metaheuristics, hybridization ,BEFRA, BEQRA .

1 Introduction

The Traveling Salesman Problem (TSP) is a fundamental and extensively studied problem in combinatorial optimization, primarily due to its NP-hardness and its wide range of practical applications in domains such as logistics, scheduling, transportation, routing, and manufacturing systems [1, 2]. Formally, given a set of cities and the pairwise distances between them, the objective is to determine the minimum-cost Hamiltonian tour that visits each city exactly once and returns to the starting city.

The factorial growth of the search space implies that the Traveling Salesman Problem (TSP) is classified as an NP-hard problem [3]. As a result, exact optimization methods become computationally intractable for large-scale instances. To address this challenge, a wide range of heuristic and metaheuristic approaches have been proposed to obtain high-quality approximate solutions within reasonable computational time. Among the most prominent methods are Genetic Algorithms (GA) [4], Particle Swarm Optimization (PSO) [5], Grey Wolf Optimizer (GWO) [6], Artificial Bee Colony (ABC) [7], Black Hole Algorithm (BHA) [8], and Ant Colony Optimization (ACO) [9].

Among these methods, Ant Colony Optimization has attracted considerable attention for solving combinatorial optimization problems, particularly the TSP. Inspired by the foraging behavior of real ants, ACO constructs solutions using pheromone trails and heuristic visibility information [10]. In the literature, the hybridization of ACO with local search heuristics has been extensively investigated, as it significantly improves solution quality by balancing exploration and exploitation [11]. Nevertheless, most existing approaches primarily rely on pheromone information and heuristic distance measures while neglecting potentially valuable structural knowledge embedded in the generated solutions.

To date, the frequency and quality of edges generated during the search process have not been sufficiently explored in population-based optimization methods. Existing studies rarely investigate the relationship between edge occurrence frequency and edge quality, nor their potential contribution to improving solution construction and convergence behavior.

The key research question addressed in this work is therefore whether the frequency and quality of edges can effectively guide the search process and enhance the quality of TSP solutions.

To address this limitation, this paper proposes novel heuristic strategies that explicitly exploit edge-level knowledge extracted from ACO-generated solutions. For the first time, the present study investigates the integration of edge frequency and edge quality into constructive heuristics for TSP solving.

The main contributions of this work are summarized as follows:

- Proposing two novel heuristics, namely BEFRA and BEQRA, based on edge frequency and edge quality analysis;
- Introducing hybrid models combined with the 2-OPT local search heuristic to improve solution refinement;
- Demonstrating the effectiveness of edge-based learning mechanisms for enhancing population-based optimization methods.

The findings of this study contribute to the field of combinatorial optimization by introducing intelligent search strategies based on structured search experience. Furthermore, the proposed framework opens new research directions for the design of advanced hybrid metaheuristics.

Recently, numerous nature-inspired metaheuristic algorithms that mimic biological and social behaviors observed in nature have been successfully applied to optimization problems. These include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gray Wolf Optimizer (GWO), Artificial Bee Colony (ABC), Black Hole Algorithm (BHA), and Ant Colony Optimization (ACO), among others [12].

1.1 Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) is a combinatorial optimization problem in which a salesman is required to visit n cities exactly once and return to the starting city, while either minimizing the total travel distance or maximizing the overall profit, depending on the defined objective function [13]. The resulting solution corresponds to a Hamiltonian tour.

For a TSP instance containing n cities, the number of feasible Hamiltonian tours is given by:

$$\frac{(n-1)!}{2} \tag{1}$$

which illustrates the combinatorial complexity of the problem. Consequently, the TSP is classified as an NP-hard optimization problem [3].

Mathematically, the TSP can be represented as a weighted undirected complete graph $G = (V, E)$, where V denotes the set of vertices (cities) and E represents the set of weighted edges corresponding to distances between cities. The objective is to identify the optimal Hamiltonian cycle among all feasible solutions.

More formally, let d_{ij} denote the distance between city i and city j. The objective function can be written as:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{2}$$

subject to constraints ensuring that each city is visited exactly once and that the resulting solution forms a single closed tour.

The TSP can be classified as symmetric or asymmetric depending on whether $d_{ij} = d_{ji}$ holds for all pairs of cities. In the Euclidean TSP, distances are computed using geometric coordinates in a plane, making it widely used in real-world routing, logistics, and network optimization applications.

The following figure illustrates a typical Traveling Salesman Problem instance with a feasible Hamiltonian tour:

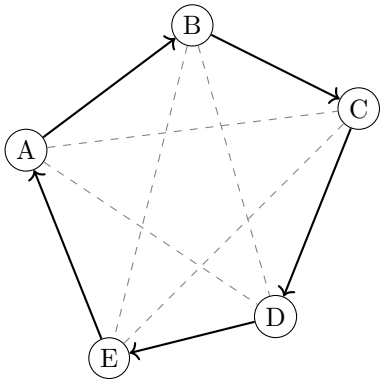


Fig. 1 Illustration of a Traveling Salesman Problem (TSP) showing a Hamiltonian tour visiting each city exactly once and returning to the starting point.

Figure 1 depicts a Hamiltonian cycle where each city is visited exactly once before returning to the origin, forming a closed loop that represents a feasible TSP solution.

1.2 Paper Organization

The remainder of this paper is organized as follows. Section II reviews the related literature on heuristic and metaheuristic approaches for solving the TSP. Section III presents the resolution methods adopted in this work. Section IV introduces the Ant Colony Optimization (ACO) metaheuristic. Section V describes the proposed methods in detail, with emphasis on the novel edge-based strategies. Section VI presents the experimental results obtained on benchmark instances and compares the proposed methods with state-of-the-art approaches, followed by a comprehensive discussion and performance analysis. Finally, Section VII concludes the paper and outlines future research directions.

2 Critical Literature Review: Heuristics and Metaheuristics for Traveling Salesman Problem Solving

The Traveling Salesman Problem (TSP) remains one of the most extensively studied combinatorial optimization problems because of its theoretical complexity and broad applicability in logistics, scheduling, transportation systems, circuit design, and manufacturing optimization [2, 13]. Since the TSP belongs to the class of NP-hard problems, exact algorithms become computationally prohibitive as the number of cities increases [3]. Consequently, researchers have increasingly focused on heuristic and metaheuristic approaches capable of producing near-optimal solutions within acceptable computational time.

Over the past two decades, the field has evolved from deterministic heuristics toward adaptive, hybrid, and intelligence-driven optimization paradigms. This evolution reflects a broader transformation in algorithmic design philosophy, where modern optimization systems are expected not only to explore efficiently but also to learn from accumulated search experience. Despite significant progress, however, no single optimization paradigm has emerged as universally dominant. Instead, the literature reveals persistent trade-offs among scalability, convergence reliability, interpretability, and computational efficiency.

This section is organized as follows. Section II-A introduces the foundations of heuristic and metaheuristic optimization methods for TSP solving. Section II-B discusses Ant Colony Optimization (ACO), including its adaptations, hybridizations, and structural limitations. Section II-C examines adaptive behavior and algorithmic memory mechanisms. Multi-objective optimization approaches are reviewed in Section II-D. Evolutionary hybrids and probabilistic alternatives are presented in Section II-E, while Pareto local search and many-objective scalability are discussed in Section II-F. Emerging paradigms based on machine learning and quantum-inspired optimization are reviewed in Section II-G. Finally, Section II-H presents recent real-world and sustainable routing extensions together with a comprehensive synthesis of heuristic and metaheuristic paradigms for TSP optimization.

2.1 Foundations of Heuristic and Metaheuristic Optimization

Early heuristic approaches such as nearest-neighbour, greedy insertion, and k-opt local search established the foundation for practical TSP solving [14]. Although computationally efficient, these methods are inherently greedy and frequently become trapped in local optima. In particular, k-opt strategies such as 2-opt and 3-opt improve solution quality by iteratively replacing edges in a tour, but their performance remains sensitive to initial solutions and problem size.

To overcome these limitations, metaheuristic frameworks including Genetic Algorithms (GA), Simulated Annealing (SA), and Ant Colony Optimization (ACO) were introduced [4, 10]. These methods provide a higher-level search strategy capable of escaping local minima by incorporating stochastic operators, probabilistic acceptance rules, or population-based exploration mechanisms. Unlike deterministic heuristics, metaheuristics trade optimality guarantees for scalability and robustness in large search spaces.

Among these methods, Ant Colony Optimization has remained particularly influential since its formalization by Dorigo and Stützle [10]. Inspired by pheromone-based communication in ant colonies, ACO combines heuristic information with collective learning mechanisms to balance exploration and exploitation. In ACO, artificial ants construct solutions probabilistically, guided by pheromone trails and visibility heuristics, where pheromone reinforcement reflects solution quality over iterations. Nevertheless, the performance of ACO remains highly dependent on parameter settings such as pheromone evaporation rate, influence factors, and population size, motivating continuous research into adaptive and hybrid variants.

In addition, theoretical studies have shown that premature convergence may occur when pheromone reinforcement becomes overly dominant, reducing exploration diversity and leading to suboptimal stagnation states.

Ismkhan [15] proposed heuristic improvements to enhance ACO scalability for large-scale optimization problems, while Zhang et al. [16] and Uddin et al. [17] demonstrated that integrating refined 2-opt operators significantly improves local exploitation capability. These hybridizations effectively combine global search ability of ACO with strong local refinement mechanisms, resulting in more balanced search behavior.

Hao et al. [18] further enhanced convergence performance by combining clustering techniques with improved ACO variants. The use of clustering reduces the effective search space by grouping similar nodes, thereby accelerating convergence and improving solution consistency. Although these studies report measurable improvements, most remain incremental by optimizing isolated algorithmic components without fundamentally redefining how search knowledge is represented or propagated.

Scianna [19] introduced AddACO, a bio-inspired extension designed to reinforce exploration dynamics through adaptive pheromone update strategies. Such contributions illustrate continuing efforts to refine pheromone-based learning mechanisms while simultaneously highlighting the persistent challenge of balancing diversification and convergence stability in complex combinatorial landscapes such as the TSP.

Overall, the evolution of heuristic and metaheuristic optimization methods reflects a gradual transition from deterministic local search procedures to adaptive population-based learning systems, where hybridization and self-adaptation are key strategies for improving performance on NP-hard problems.

2.2 Ant Colony Optimization: Adaptation, Hybridization, and Structural Limits

A major limitation of Ant Colony Optimization (ACO) concerns its strong dependence on parameter tuning. Tuani et al. [20] argued that identifying optimal parameter configurations is computationally expensive and often impractical because effective settings vary significantly across problem instances. Parameters such as pheromone evaporation rate, heuristic influence, and exploitation–exploration balance directly affect convergence behavior and solution diversity. To address this issue, they proposed a heterogeneous adaptive ACO in which ants dynamically evolve behavioral traits while integrating a 3-opt local search operator. Experimental results demonstrated improved robustness and solution quality compared to state-of-the-art approaches.

What distinguishes this class of approaches is the reinterpretation of algorithmic parameters as adaptive learning components rather than static controls. This enables the algorithm to adjust its behavior dynamically during the search process. However, such self-adaptation increases algorithmic complexity and may introduce challenges related to stability, reproducibility, and theoretical interpretability, especially when multiple adaptive mechanisms interact.

Structural enhancements have also been widely explored in the literature. Xiao et al. [21] proposed a multi-strategy improved ACO combining nearest-neighbour initialization, mutation-based transition rules, and elitist pheromone updating. These

mechanisms aim to maintain population diversity in early iterations while reinforcing high-quality solutions during later stages, thereby improving convergence speed and solution accuracy.

Similarly, Hertono et al. [22] proposed a hybrid ACO–PSO framework in which Particle Swarm Optimization is used for parameter tuning, while ACO is responsible for constructing candidate solutions. A 3-opt local search procedure is further integrated to refine solutions. Although this hybridization improves solution quality, it also increases computational cost, highlighting the persistent trade-off between intensification and efficiency.

From a broader perspective, these studies demonstrate that most improvements in ACO are achieved through architectural modifications and hybridization strategies that combine multiple metaheuristic components. Nevertheless, a deeper limitation remains in how search history is encoded and exploited through pheromone information. The design of more expressive and adaptive memory structures therefore remains an open research direction in ACO-based optimization. The general hybrid ACO architecture is illustrated in Fig. 2, where solution construction, pheromone learning, local search, and adaptive control mechanisms interact in a unified optimization loop.

2.3 From Adaptive Behavior to Algorithmic Memory

Allowing optimization agents to adapt their behavior dynamically helps prevent stagnation and preserve population diversity during exploration. However, once parameters become self-regulating, optimization algorithms begin to resemble learning systems rather than conventional heuristics.

This shift introduces a critical conceptual issue related to the representation of accumulated search experience. Most ACO variants compress historical knowledge into scalar pheromone values, potentially oversimplifying the statistical richness contained within high-quality tours.

Zhang et al. [23] addressed premature convergence in multi-objective optimization by introducing a Physarum-inspired mathematical model to optimize pheromone initialization. Their approach enhanced shorter paths through positive feedback mechanisms, producing superior compromise solutions compared with conventional MOACO implementations. Nevertheless, reliance on prior knowledge may reduce exploratory flexibility when the search landscape evolves dynamically.

2.4 Multi-Objective Optimization: Expanding the Definition of Optimality

Real-world routing problems rarely involve a single objective since cost, travel time, risk, and environmental impact frequently conflict. Consequently, multi-objective optimization shifts the focus from identifying a single optimal solution toward approximating Pareto-optimal fronts.

García-Martínez et al. [24] demonstrated that evaluating algorithmic performance in such contexts requires measuring trade-offs rather than simple dominance relationships.

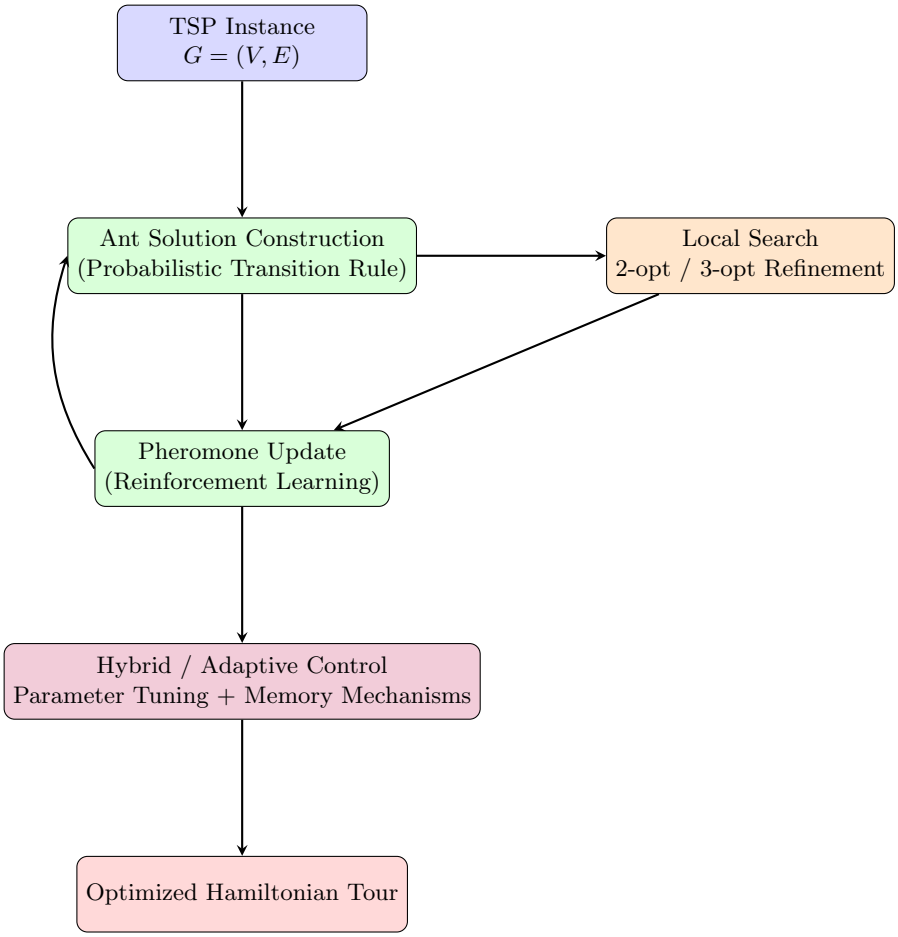


Fig. 2 Colored hybrid Ant Colony Optimization (ACO) architecture integrating solution construction, pheromone updating, local search, and adaptive control mechanisms.

Lin et al. [25] extended Multi-Objective Ant Colony Optimization (MOACO) to forest transportation planning by simultaneously minimizing economic cost and environmental impact while generating non-dominated routes for decision support. Experimental results demonstrated the practical relevance of Pareto-based optimization approaches.

Similarly, Chen et al. [26] identified premature convergence and insufficient population diversity as major limitations of genetic algorithms applied to the Multi-Objective TSP. They proposed a Physarum-inspired initialization mechanism combined with hill climbing refinement, achieving superior performance relative to existing approaches.

Ekanayake [27] further demonstrated that modified ACO heuristics can effectively solve bi-objective and tri-objective transportation problems, although much of the validation remains empirical rather than theoretically grounded.

Despite these advancements, multi-objective optimization methods often fragment learning across competing objective structures, raising important questions regarding how effectively knowledge is integrated during the search process.

2.5 Evolutionary Hybrids and Probabilistic Alternatives

Hybrid evolutionary frameworks continue to attract significant attention because they attempt to reconcile global exploration with local exploitation. Psychas et al. [28] introduced Differential Evolution-based hybrids combined with Variable Neighbourhood Search, demonstrating strong optimization performance across problems involving up to five objectives.

From a probabilistic perspective, Zhou et al. [29] proposed a decomposition-based Estimation of Distribution Algorithm (EDA) that constructs probabilistic models using both prior and learned information. Their method outperformed MOACO variants while remaining comparatively insensitive to parameter settings.

This observation is theoretically significant because probabilistic modeling explicitly represents population structure, suggesting that statistical inference may guide optimization more effectively than pheromone accumulation alone.

2.6 Pareto Local Search and Many-Objective Scalability

As the number of objectives increases, maintaining non-dominated solution sets becomes computationally expensive. Cai et al. [30] proposed a grid-weighted Pareto local search strategy that preserves representative solutions within structured grids, thereby reducing both computational complexity and memory requirements while maintaining diversity.

Experimental results demonstrated superior performance compared with conventional Pareto Local Search and decomposition-based approaches. This work highlights an emerging concern: scalability in many-objective optimization depends not only on search operators but also on efficient population management mechanisms.

2.7 Emerging Paradigms: Learning-Based and Quantum-Inspired Optimization

Recent advances in artificial intelligence have introduced data-driven paradigms into TSP optimization. Sui et al. [31] categorized deep learning-based TSP solvers into construction, improvement, hybrid, and large language model-driven approaches, highlighting a transition toward automated heuristic generation.

Luo et al. [32] employed Graph Attention Networks within a reinforcement learning framework to learn efficient 2-opt moves, while Ahmad [33] combined reinforcement learning with graph neural networks to achieve scalable optimization performance with near-optimal accuracy. Although these methods reduce dependence on handcrafted heuristics, they frequently sacrifice interpretability.

Quantum-inspired optimization methods further extend the methodological landscape. Saini et al. [34] demonstrated the feasibility of hybrid quantum-classical evolutionary algorithms on cloud-based quantum platforms, although scalability remains constrained by current hardware limitations. Mohammed et al. [35] similarly identified

quantum annealing as promising for Quadratic Unconstrained Binary Optimization (QUBO) formulations while emphasizing the necessity of hybrid frameworks.

Together, these studies suggest that the field is evolving toward algorithmic intelligence, although consensus regarding the most sustainable architectural foundation has yet to emerge.

2.8 Real-World Extensions and Sustainable Routing

Recent studies increasingly embed the TSP within realistic logistics and sustainability contexts. Babaee Tirkolaee et al. [36] introduced the Traveling Salesman Problem with Drone and Bicycle, demonstrating reductions in both service time and environmental impact.

Similarly, Zhu et al. [37] proposed a hybrid vehicle-drone routing framework that achieved substantial cost reductions relative to vehicle-only delivery systems.

These studies confirm the practical relevance of heuristic optimization methods while simultaneously highlighting a recurring challenge: algorithms validated on benchmark datasets often struggle when confronted with uncertainty, dynamic constraints, and conflicting objectives.

Overall, the evolution of TSP-solving methodologies reflects a broader transformation in optimization, progressing from handcrafted heuristics toward systems capable of structured learning and adaptive intelligence. Future advances are unlikely to emerge from isolated paradigms alone; instead, they will probably arise through principled integration across swarm intelligence, adaptive control, probabilistic reasoning, and machine learning frameworks.

The next generation of metaheuristics will therefore be distinguished less by raw computational speed than by their ability to transform accumulated search experience into structured, decision-guiding knowledge. **Comprehensive Synthesis of Traveling Salesman Problem Optimization Approaches** To consolidate the preceding discussion, Table I provides a structured synthesis of the principal research streams in Traveling Salesman Problem optimization. Rather than cataloguing studies individually, the table groups contributions by methodological paradigm and highlights their core ideas, strengths, limitations, and broader research implications. This integrative perspective clarifies the evolution of heuristic and metaheuristic design while exposing the structural challenges that continue to motivate further investigation.

The comparative analysis reveals that although methodological diversity has expanded significantly, many approaches share a common structural limitation: the restricted representation of historical search knowledge. Pheromone trails, parameter updates, or learned policies often encode only partial information about solution quality and edge reliability. Therefore, the central challenge is no longer exploration alone, but how optimization algorithms capture, retain, and exploit accumulated experience. This observation motivates the development of approaches that incorporate richer edge-level intelligence to strengthen convergence while preserving adaptive search behavior.

Table 1 Comprehensive synthesis of heuristic and metaheuristic paradigms for TSP optimization

Paradigm	Representative Studies	Core Idea	Strengths	Key Limitations	Research Implication
Classical Heuristics	Nearest Neighbour, k-opt variants; [16, 38]	Construct feasible tours using deterministic improvement strategies	Computationally efficient; simple to implement	Highly myopic; prone to local optima	Motivated the transition toward stochastic metaheuristics
Swarm Intelligence (ACO)	[15, 39, 40]	Collective learning via pheromone-mediated search	Strong exploration–exploitation balance; adaptable	Parameter sensitivity; stagnation risk	Established distributed learning as a foundation for routing optimization
Adaptive and Hybrid ACO	[41–43]	Combine local search, mutation, or parameter self-adaptation	Improved robustness and convergence	Increased algorithmic complexity; runtime trade-offs	Signals a shift from static heuristics toward self-regulating systems
Bio-inspired Enhancements	[44, 45]	Use Physarum-inspired feedback or improved initialization	Better diversity and compromise solutions	Prior knowledge may bias exploration	Highlights the importance of initialization and search memory
Multi-objective Optimization	[46–48]	Approximate Pareto fronts instead of single optima	Supports decision-making under conflicting criteria	Fragmented learning across objectives	Redefines optimality as a trade-off rather than a single solution
Evolutionary Hybrids	[49, 50]	Integrate global evolutionary search with local refinement	Strong performance across complex landscapes	Higher computational demand	Cooperation between global and local search improves stability
Probabilistic Modeling	[29, 30]	Explicitly model population distributions	Reduced parameter dependence; scalable	Modelling overhead	Suggests a transition from pheromone memory to statistical inference
Learning-Based Approaches	[31–33]	Learn decision policies using neural architectures	Adaptive; reduced need for handcrafted heuristics	Limited interpretability; data dependence	Marks the emergence of algorithmic autonomy
Quantum and Hybrid Quantum Methods	[34, 35]	Leverage quantum principles for combinatorial search	Strong theoretical promise	Hardware and scalability constraints	Likely to evolve through hybrid classical–quantum systems
Real-World and Sustainable Extensions	[36, 51]	Embed TSP within multimodal and green logistics	High practical relevance	Sensitive to uncertainty and operational constraints	Demonstrates the growing alignment between optimization and sustainability

3 Resolution Methods

Combinatorial Optimization Problems (COPs) are simple to formulate yet generally extremely complex to solve, as many are classified as NP-hard [52]. This complexity, coupled with their practical importance, has led to the development of various solution methodologies. Algorithms used to tackle these problems are generally categorized into two families: Exact methods and Approximate methods.

3.1 Exact Methods

Exact algorithms guarantee the discovery of an optimal solution and prove its optimality for every finite instance of a COP within a specific, finite runtime. Notable exact methods include Integer Programming (IP) and Dynamic Programming (DP). Well-known IP techniques include Branch and Bound, Branch and Cut, and Branch and Price.

3.2 Approximate Methods

When computing optimal solutions becomes impractical due to large-scale problem constraints, it is common to prioritize computational efficiency over the guarantee of optimality. In such cases, the assurance of finding the global optimum is sacrificed for the ability to obtain high-quality solutions using approximate algorithms within a reasonable timeframe.

- **Heuristic Methods:** These are tailored for specific problem types and provide satisfactory solutions without ensuring optimality.

– *Constructive Heuristics:* These algorithms begin with an empty solution and iteratively extend it until a complete feasible solution is obtained.
– *Local Search (Improvement) Heuristics:* These start with an initial feasible solution and apply local modifications to enhance solution quality.

- **Metaheuristic Methods:** Derived from the Greek words *meta* ("beyond" or "higher level") and *heuristic* ("to find" or "to discover"), metaheuristics generally refer to a "higher level of heuristics" [53]. Metaheuristics are designed to explore large search spaces and are categorized into two main groups:

  1. **Single-solution-based algorithms:** These focus on local and neighborhood search trajectories.
  2. **Population-based algorithms:** These include evolutionary algorithms and swarm intelligence-based algorithms, which maintain a diverse set of candidate solutions.

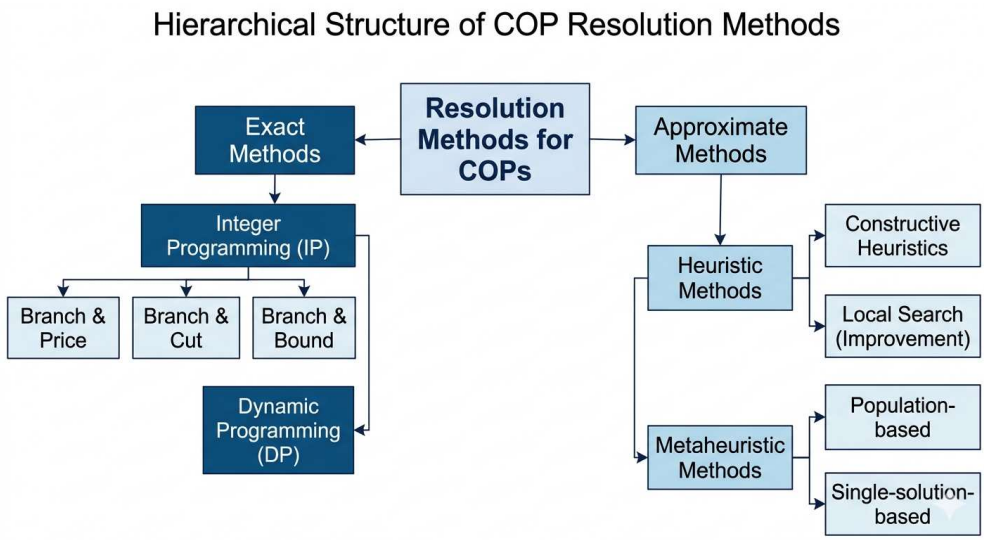


Fig. 3 Hierarchical Structure of COP Resolution Methods

4 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a stochastic metaheuristic inspired by the collective foraging behavior of real ant colonies. Originally introduced by Marco Dorigo [54], ACO was first applied to solve NP-hard combinatorial optimization problems such as the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP).

4.1 Basic ACO Algorithm

The fundamental concept of the Ant Colony Optimization (ACO) algorithm is inspired by the collective foraging behavior of real ants in their search for food sources. Ants are able to discover the shortest path between a food source and their nest through a mechanism of indirect communication known as stigmergy [55].

Initially, ants follow random paths and deposit a chemical substance called pheromone on the ground. When an ant discovers a food source, it returns to the nest while reinforcing the pheromone trail. Other ants are statistically more likely to follow paths with higher pheromone concentrations. As more ants traverse the shortest path, the pheromone density increases, eventually leading the entire colony to converge on the optimal route [55].

According to [56], the basic ACO algorithm consists of four primary stages:

1. **Initialization:** All global parameters are defined, including the number of ants (m), maximum iterations, pheromone influence (α), heuristic influence (β), and the evaporation rate (ρ). The problem-specific objective matrix and initial pheromone levels are also set.
2. **Construct Ant Solutions:** In each iteration, every ant constructs a complete solution by moving between nodes. The transition from node i to node j is governed by a probabilistic moving rule:

$$p_{i,j} = \begin{cases} \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\sum_{k \in allowed_k} [\tau_{i,k}]^\alpha [\eta_{i,k}]^\beta} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

   where τ_{ij} represents the pheromone information, η_{ij} is the heuristic information (usually the inverse of distance), and P_{ij}^k is the probability of ant k choosing node j when at node i.
3. **Local Search (Optional):** This step involves the refinement of constructed solutions using local search heuristics (e.g., 2-opt) to improve solution quality before the pheromone update.
4. **Pheromone Update:** Once all ants have completed their tours, the pheromone trails are updated. This process begins with pheromone evaporation to prevent unlimited accumulation and allow for the exploration of new paths:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{4}$$

   where ρ is the evaporation rate and $\Delta\tau_{ij}^k$ is the quantity of pheromone deposited by ant k, typically proportional to the quality of the solution it found.

5 Proposed Methods

5.1 Based Edge Frequency Ranking Heuristic (BEFRA)

In this section we present our method named Based Edge Frequency Ranking Algorithm (BEFRA). BEFRA is a constructive heuristic for Traveling Salesman Problem that constructs a solution step by step from scratch; BEFRA starts from empty solution and repeatedly we add one edge at time until an Hamiltonian Circuit is formed. The chosen edge is the well ranked edge based on his frequency in a set of solutions found by the Ant Colony Optimization Algorithm.

The procedure of BEFRA is shown in Fig (1), it takes INPUTs all ACO's solutions and returns improved solution or the best ACO's solution else.

Algorithm 1: BEFRA Procedure

Input: $Tours$: Set of all solutions generated by ACO;
It: Termination criteria ($It \in [1, \text{size}(Tours) - n]$);
X, D: Data and Distance matrix;
$Best_sol$: Best solution found by ACO;
$k = 1$
Output: Imp_sol: Improved Solution

Begin
`// Step 1: Calculate Edge Frequency`
Create $freq_{ij}$ and calculate the occurrences of each edge (i, j) in $Tours$
`// Step 2: Sort Edges Frequencies`
Sort $freq_{ij}$ in descending order to prioritize high-frequency edges
`// Step 3: Iterative Construction and Comparison`
while $k \leq It$ **do**
  Construct a new solution (Hamiltonian circuit) starting from the k-th sorted edge
  if *Cost of Constructed Solution* $<$ *Cost of* $Best_sol$ **then**
    $Imp_sol \leftarrow$ Constructed Solution
    $Best_sol \leftarrow$ Constructed Solution `// Update global best`
  end
  $k = k + 1$
end
`// Step 4: Return Result`
return $\max(Imp_sol, Best_sol)$
End

In this section we illustrate BEFRA's procedure with an example of 11 cities TSP. The Based Edge Frequency Ranking Algorithm (BEFRA) start from a set of all solutions generated by ACO Algorithm shown in Fig 4. Taking in account that we start from a complete graph with fully Edges and we have (n*(n-1)/2) possible Edges in our solutions, the first step consist of creating a new structure that contains the Frequency of each Edge labelled $freq_ij$ in Solutions set.

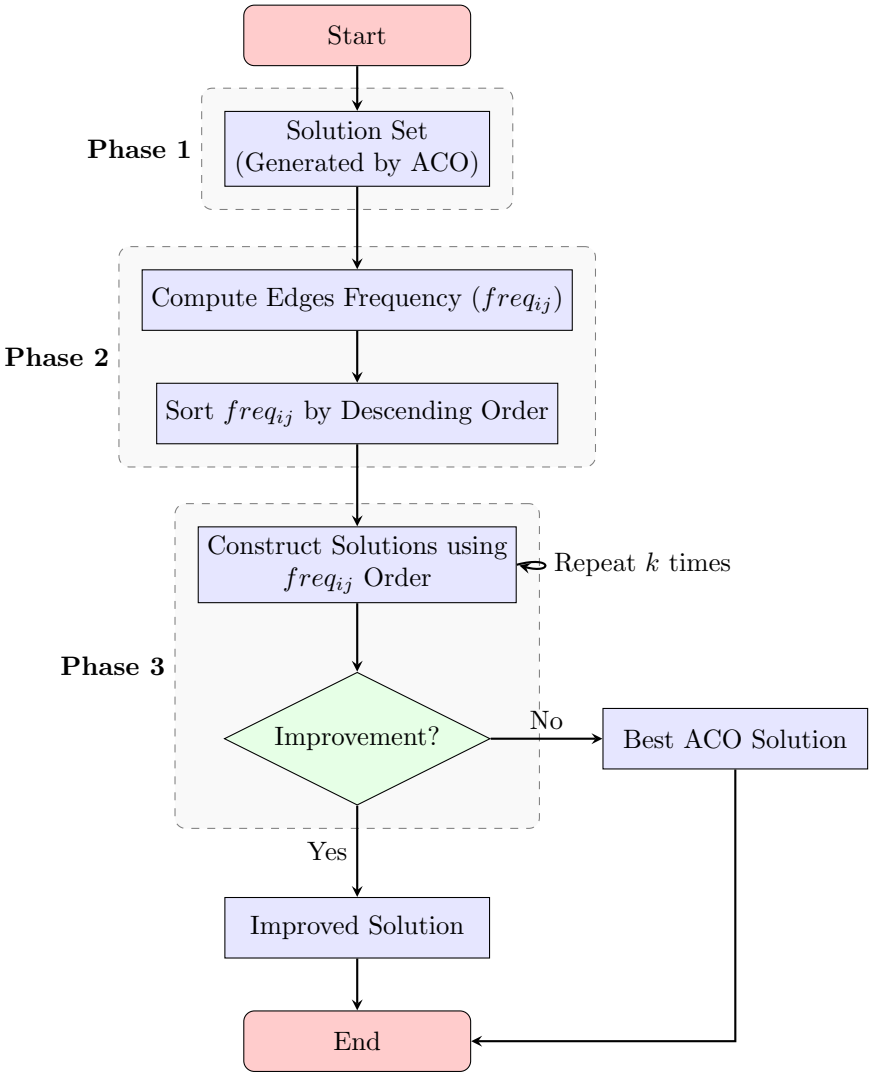


Fig. 4 Detailed flowchart of the BEFRA procedure with segmentation by computational phases.

5.2 Illustrative Example: 11-City TSP

In this section, we illustrate the BEFRA procedure using a 11-city Traveling Salesman Problem (TSP) instance.

The Based Edge Frequency Ranking Algorithm (BEFRA) begins with a set of all solutions generated by the ACO algorithm, as depicted in Fig. 4. Considering a complete graph where all edges are initially present, there are $n(n-1)/2$ possible edges within the solution space. The primary step involves constructing a new data structure to record the frequency of each edge, denoted as $freq_{ij}$, across the entire set of solutions.

Table 2 Solutions generated by ACO.

No.	Tour Path (Sequence of Cities)										
1	6	8	7	10	9	11	4	3	1	2	5
2	8	6	1	9	5	4	3	10	7	2	11
3	4	5	11	7	6	1	2	9	3	10	8
4	5	9	11	4	3	10	7	8	6	1	2
5	4	5	11	9	8	6	1	7	10	3	2
6	2	4	5	9	3	10	7	8	6	1	11
...											
23	1	7	2	10	3	9	5	11	4	8	6
24	1	7	8	6	3	10	9	5	4	2	11
25	3	10	9	7	8	6	1	2	4	5	11
26	8	6	1	2	4	9	5	10	7	11	3
27	10	7	2	4	5	11	9	3	1	6	8
28	8	6	1	7	10	3	5	11	9	4	2
...											
61	1	4	3	10	9	5	6	8	7	11	2
62	1	2	4	5	11	9	10	3	7	8	6
63	2	4	5	11	10	3	9	1	7	8	6
64	2	5	9	3	4	1	10	11	7	8	6
65	9	3	1	2	4	5	11	10	7	8	6
66	5	11	9	7	8	6	2	10	4	3	1
...											
95	9	3	10	7	1	2	4	5	11	6	8
96	6	1	7	10	5	9	11	3	4	2	8
97	10	7	8	6	1	2	9	5	4	3	11
98	10	9	3	4	5	1	6	8	7	11	2
99	2	5	4	9	1	7	8	6	10	3	11
100	2	4	3	10	1	7	8	6	11	9	5

The second step consist of sorting Edges Frequencies by descending order , Fig (3) shows how many time each edge appears in ACO's solutions (Edges frequencies), edge(8-6) appears in 88% of solutions, edge (7-8) appears in 78% and edge (3-8) appears just once in 100 ACO's solutions.

Table 3 Ranking of edges based on their frequency of occurrence ($freq_{ij}$) in the ACO solution set.

Edge		Freq	Edge		Freq	Edge		Freq
8	6	88	1	3	16	10	5	7
7	8	78	10	1	14	1	4	7
3	10	64	6	2	14	3	5	7
4	5	55	11	3	14	7	6	7
9	5	54	7	11	13	8	10	7
6	1	52	2	3	12	3	7	6
5	11	50	4	9	11	6	4	6
1	2	49	2	5	10	6	3	6
10	7	42	7	2	10	2	8	5
3	4	42	6	9	9	6	11	5
2	4	34	9	1	9	1	5	5
11	2	34	10	4	9	6	5	5
1	7	33	2	10	8	4	8	4
9	3	32	1	11	7	11	8	4
11	9	31	6	10	7	4	7	4
11	4	26	9	8	7	5	7	4

6 BEFRA Solution Construction

Following the frequency ranking, we construct Hamiltonian circuits based on the sorted edge order. Fig. 3 illustrates the solutions generated by the BEFRA procedure. The final output of the algorithm is either the newly discovered improved solution or the original best solution from the ACO phase.

For the first iteration ($k = 1$), the process begins with the highest-ranked edge, $(8, 6)$. We then construct a Hamiltonian circuit while respecting the frequency-based priority. In Fig. 3, selected edges are indicated by a black dot, while non-selected edges are marked with a red dot.

The first BEFRA solution is defined as:

$$S^1_{BEFRA} = \{8, 6, 1, 2, 11, 9, 5, 4, 3, 10, 7, 8\} \tag{5}$$

For any subsequent iteration k, the construction starts from the k-th ranked edge. For example, Fig. 3 displays the 15^{th} iteration of the BEFRA procedure, which initiates from the 15^{th} ranked edge, $(11, 9)$.

The resulting 15^{th} BEFRA solution is:

$$S^{15}_{BEFRA} = \{11, 9, 10, 1, 3, 2, 6, 7, 5, 8, 4, 11\} \tag{6}$$

Table 4 Edge selection process for BEFRA construction. Black dots (•) indicate selected edges; red dots (•) indicate non-selected edges.

Stat.	Edge		Freq	Stat.	Edge		Freq
•	8	6	88	•	9	3	32
•	7	8	78	•	11	9	31
•	3	10	64		11	4	26
•	4	5	55		10	9	26
•	9	5	54		2	9	16
•	6	1	52		11	10	16
•	5	11	50		1	3	16
•	1	2	49		10	1	14
•	10	7	42		6	2	14
•	3	4	42		11	3	14
•	2	4	34		7	11	13
•	11	2	34		2	3	12
•	1	7	33		4	9	11

6.1 Rationale for the Iterative Nature of BEFRA

The repetition of the BEFRA procedure is a critical design feature for solving large-scale Traveling Salesman Problems (TSPs). During the construction of a new solution, certain high-ranking edges are often excluded because their inclusion would create a sub-tour, which is strictly forbidden in a valid Hamiltonian circuit. These excluded edges are represented by red dots in Fig. 4 and Fig. 5.

Table 5 Edge frequency ranking and selection for the 15th iteration of the BEFRA procedure. Black dots (●) denote selected edges; red dots (●) denote non-selected edges.

St.	Edge		Fr.	St.	Edge		Fr.	St.	Edge		Fr.	St.	Edge		Fr.
●	8	6	88	●	9	3	32	●	2	5	10	●	8	10	7
●	7	8	78	●	11	9	31	●	7	2	10	●	3	7	6
●	3	10	64	●	11	4	26	●	6	9	9	●	6	4	6
●	4	5	55	●	10	9	26	●	9	1	9	●	6	3	6
●	4	5	54	●	2	9	16	●	10	4	9	●	2	8	5
●	6	1	52	●	11	10	16	●	2	10	8	●	6	11	5
●	5	11	50	●	1	3	16	●	1	11	7	●	1	5	5
●	1	2	49	●	10	1	14	●	6	10	7	●	6	5	5
●	10	7	42	●	6	2	14	●	9	8	7	●	4	8	4
●	3	4	42	●	11	3	14	●	10	5	7	●	11	8	4
●	2	4	34	●	7	11	13	●	1	4	7	●	4	7	4
●	11	2	34	●	2	3	12	●	3	5	7	●	5	7	4
●	1	7	33	●	4	9	11	●	7	6	7	●	9	7	3
												●	1	8	2
												●	5	8	2
													3	8	1

By iteratively repeating Steps 3 and 4 and initiating the construction from a different "seed" edge in each iteration, we allow previously excluded high-frequency edges to participate in subsequent Hamiltonian circuits. This dynamic approach significantly increases the probability of discovering an improved solution (Imp_sol) that the standard ACO might have missed.

As illustrated in the 15^{th} iteration (Table 5), once the edge $(2, 3)$ is selected, the subsequent 13 edges (from $(4, 9)$ to $(3, 5)$) must be excluded to maintain feasibility. This is followed by the selection of edge $(7, 6)$, which in turn forces the exclusion of another sequence of 8 edges (from $(8, 10)$ to $(6, 5)$). This phenomenon demonstrates why the repetitive nature of BEFRA is crucial for exploring the solution space effectively.

7 Hybrid Model: BEFRA-2opt Heuristic

In this variant, we integrate a local search heuristic, specifically the 2-opt algorithm, to further refine the best solution produced by the BEFRA procedure. This hybridization aims to escape local optima by exploring the immediate neighborhood of the BEFRA output.

The BEFRA-2opt process begins with the best solution found during the iterative BEFRA phase. The 2-opt heuristic then iteratively swaps two edges within the Hamiltonian circuit to identify a shorter total path. This dual-stage refinement can be summarized as a two-fold improvement process:

1. **Metaheuristic Improvement (BEFRA):** Enhances the initial solution set generated by the ACO algorithm through frequency-based reconstruction.
2. **Local Search Improvement (2-opt):** Refines the BEFRA result by eliminating edge crossings and optimizing local segments of the tour.

If an improvement is successfully identified, it signifies that the model has reached a higher level of optimization than either ACO or standalone BEFRA could achieve. The complete logical flow of the BEFRA-2opt procedure is illustrated in Fig. 5.

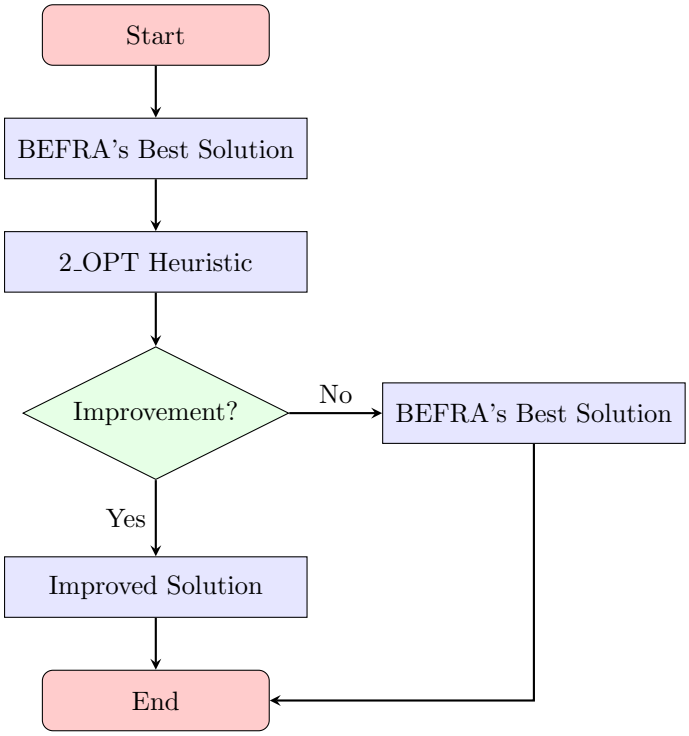


Fig. 5 Flowchart of the Hybrid BEFRA-2opt procedure.

8 Based Edge Quality Ranking Heuristic (BEQRA)

The Based Edge Quality Ranking Algorithm (BEQRA) follows a similar logic to BEFRA but introduces three fundamental shifts in how edges are prioritized.

8.1 Cumulated Pheromone Matrix

Unlike the standard ACO update, we maintain a cumulated pheromone variable, $\tau_{i,j}^{cum}$, which ignores evaporation. This serves as a direct indicator of edge quality; as edges appearing in superior solutions receive higher pheromone deposits, their "quality" score increases. The pheromone deposited by ant k is defined as:

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k & \text{if edge } (i,j) \in \text{tour constructed by ant } k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where L_k represents the total length of the tour for ant k.

8.2 Pheromone Update Rule

The update rule for the cumulated pheromone trail $\tau_{i,j}^{cum}$ excludes the evaporation coefficient $(1-\rho)$ to preserve the historical quality of each edge:

$$\tau_{i,j}^{cum}(t+1) = \tau_{i,j}^{cum}(t) + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}(t) \tag{8}$$

This ensures that the more an edge (i,j) participates in high-quality solutions, the higher its ranking becomes in the BEQRA structure.

9 Hybrid Model: BEQRA-2opt Heuristic

The BEQRA-2opt variant integrates a local search refinement phase to optimize the quality-driven solutions produced by the BEQRA algorithm. This model utilizes the 2-opt heuristic to systematically explore the neighborhood of the best solution identified during the BEQRA iterations.

Starting from the best Hamiltonian circuit found by BEQRA, the 2-opt heuristic iteratively performs edge swaps. If a swap results in a shorter total path, the solution is updated. This hybrid approach achieves a dual-layered improvement:

1. **BEQRA Refinement:** Reconstructs the solution by prioritizing edges with higher cumulated pheromone (τ_{ij}^{cum}), representing superior historical "quality."
2. **2-opt Local Optimization:** Further refines the high-quality BEQRA solution by eliminating inefficient edge crossings and reducing the overall tour length.

When an improvement is identified through 2-opt, the model is considered to have improved twice: first by the BEQRA metaheuristic logic and second by the deterministic local search. This synergy ensures a more robust convergence toward the global optimum for the TSP.

10 Experiments and Results

In this section, we present a comprehensive evaluation of the proposed algorithms. To ensure a fair comparison, all experiments were conducted in a consistent computational environment, comparing our hybrid models (**BEFRA, BEFRA-2opt, BEQRA, and BEQRA-2opt**) against established heuristics and metaheuristics from the literature [13, 57, 58].

10.1 Experimental Setup and Parameters

The performance of the algorithms was tested on seven standard benchmarks from the TSPLIB library: **Att48, Eil51, Eil76, Eil101, Berlin52, St70, and KroA100**. Each experiment was executed under the following conditions:

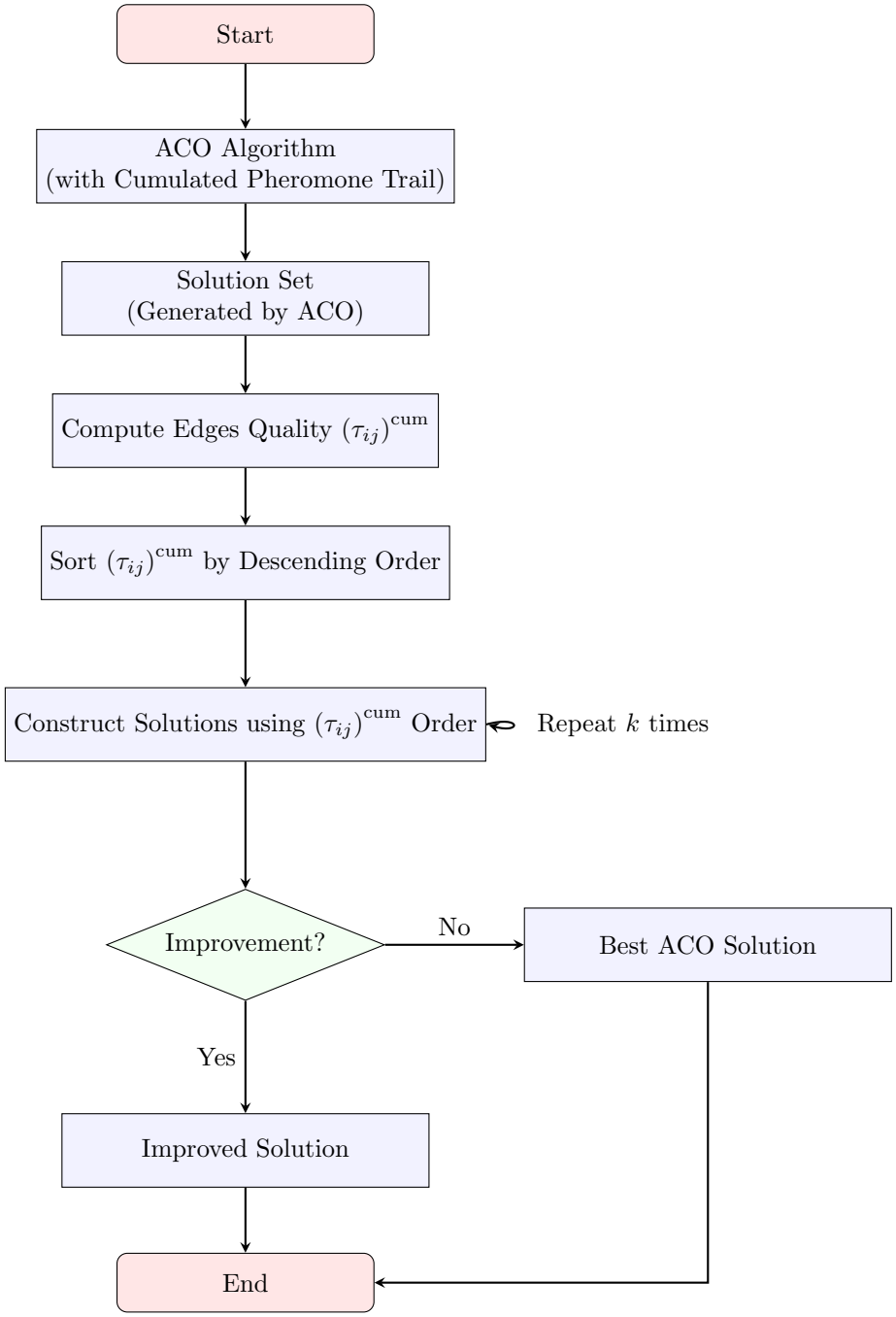


Fig. 6 The logical procedure of the Based Edge Quality Ranking Algorithm (BEQRA).

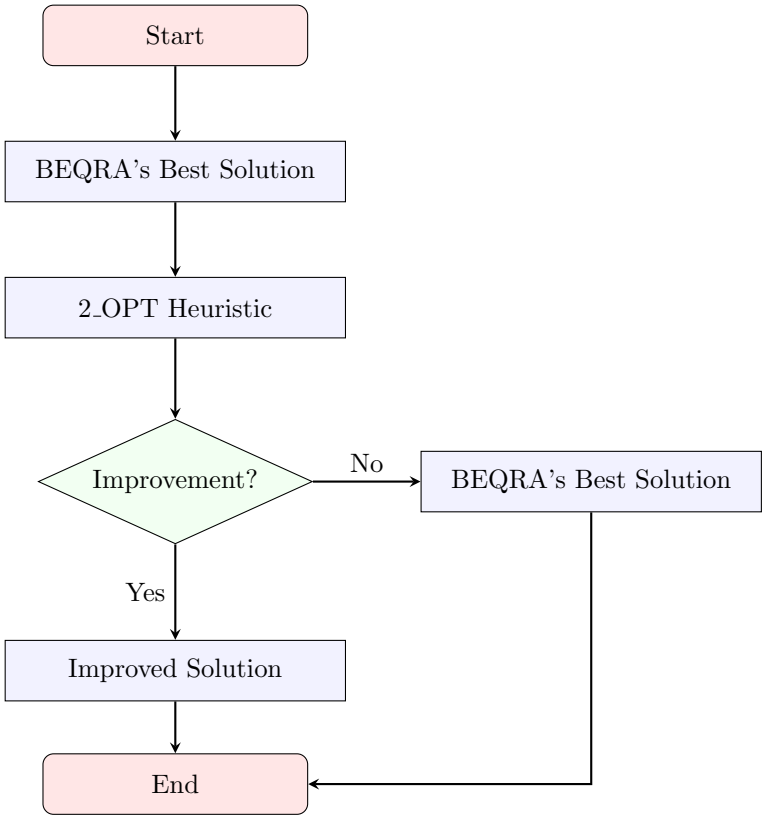


Fig. 7 Flowchart of the Hybrid BEQRA-2opt procedure.

- **Population Size:** 100 agents (ants/bees/particles).
- **Iterations:** 100 iterations per run.
- **Statistical Validity:** 10 independent runs for each instance.

The computational environment utilized for all simulations is specified in Table 6.

Table 6 Hardware and Software Specifications.

Component	**Specification**
Processor	12th Gen Intel(R) Core(TM) i7-1255U (1.70 GHz)
RAM	16.0 GB
Operating System	Windows 11 (64-bit)
Implementation Language	[Insert Language, e.g., MATLAB / Python / C++]

10.2 Comparative Algorithms

Our results are benchmarked against a wide range of algorithms as presented by Maquin et al. [57–59], including:

- **Classical Heuristics:** Nearest Neighborhood (NN), Clarke and Wright (C&W), and 2-opt Heuristic.
- **Nature-Inspired Metaheuristics:** Artificial Bee Colony (ABC), Black Hole Algorithm (BHA), Genetic Algorithm (GA), and standard Ant Colony Optimization (ACO).

Table 7 Comparison between proposed methods and classical heuristics/metaheuristics. Best values are highlighted in bold.

Instance	ACO		BEFRA		BFR2OPT		BEQRA		BQR2OPT	
	Best	Mean	Best	Mean	Best	Mean	B	Mean	Best	Mean
ATT48	36723	37862	36212	40632	**33524**	34148	37778	39649	**33524**	34289
EIL51	477.37	495.838	449.95	487.35	429.53	437.55	453.16	483.3	**433.2**	437.91
EIL76	649.41	655.095	583.99	608.24	556.05	568.68	604.62	615.91	**553.97**	**562.9**
EIL101	822.54	834.106	701.05	754.25	649.11	662.71	698.2	746.5	**645.8**	**658.0**
BERLIN52	8185.2	8553.42	8081.1	8559.41	**7544.4**	7796.74	8008.3	8590.46	**7544.4**	7768.04
KROA100	26964	28091.5	24439	26461.6	**21307**	22207	25049	26111.3	21395	21991.6
ST70	785.35	830.13	754.9	797.87	**689.09**	703.53	770.18	803.99	692.07	707.21

Table 1 presents a comparative analysis of four proposed algorithms BEFRA, BEFRA2OPT, BEQRA, and BEQRA2OPT with ACO metaheuristic over seven TSPLIB standard benchmark instances of the Traveling Salesman Problem. For each algorithm, two performance indicators are reported:

- **BEST:**the best solution (minimum tour length) obtained over 10 runs;
- **MEAN:** the average solution quality across all runs, reflecting both performance and robustness.

Generally the four proposed algorithms BEFRA,BEQRA,BEFRA2OPT and BQR2OPT outperform ACO Algorithm across all benchmarks.

The proposed BEFRA2OPT algorithm consistently yields the best mean performance across all benchmarks, demonstrating both accuracy and stability. When BEQRA2OPT variant slightly improves the best solutions in some instances (e.g., EIL76, EIL101), indicating its effectiveness in local search refinement.

Both BEFRA2OPT and BEQRA2OPT outperform the classical ACO , BEQRA and BEFRA methods, which exhibit higher mean deviations and weaker consistency.

The results confirm that BEFRA2OPT and BEQRA2OPT outperform classical ant colony ACO, BEFRA and BEQRA algorithms on all standard TSP benchmarks.

The hybridization strategy significantly enhances convergence speed, solution quality, and robustness.

This table illustrates the results via the mean ranking based on Best cost . Both BEFRA2OPT and BEQRA2OPT have similar performance with small advantage for

Table 8 Best Cost Ranking comparisons :Result in bold indicates best result of each category.

Instance	ACO	BEFRA	BEFRA2OPT	BEQRA	BEQRA2 OPT
ATT48	3	2	1	4	1
EIL51	5	3	1	4	2
EIL76	5	4	2	3	1
EIL101	5	4	2	3	1
BERLIN52	5	3	1	4	2
KROA100	5	3	1	4	2
ST70	5	3	1	4	2
MEAN_RANK	**4.71**	**3.00**	**1.28**	**3.57**	**1.42**

BEFRA2OPT where BEFRA2OPT have lowest average best cost in 5/7 instances with an average ranking of 1.28

Table 9 Mean Ranking Comparisons :Result in bold indicates best result of each category.

Instance	ACO	BEFRA	BEFRA2OPT	BEQRA	BEQRA2OPT
ATT48	4	5	1	3	2
EIL51	5	4	1	3	2
EIL76	5	3	2	4	1
EIL101	5	4	2	3	1
BERLIN52	3	5	2	4	1
KROA100	5	4	2	3	1
ST70	5	3	1	4	2
MEAN_RANK	**4.57**	**4.00**	**1.57**	**3.42**	**1.42**

Table 10 Mean Ranking Comparisons :Result in bold indicates best result of each category.

Instance	ACO	BEFRA	BEFRA2OPT	BEQRA	BEQRA2OPT
ATT48	4	5	1	3	2
EIL51	5	4	1	3	2
EIL76	5	3	2	4	1
EIL101	5	4	2	3	1
BERLIN52	3	5	2	4	1
KROA100	5	4	2	3	1
ST70	5	3	1	4	2
MEAN_RANK	**4.57**	**4.00**	**1.57**	**3.42**	**1.42**

This table illustrates the results via the mean ranking based on Mean Cost of 10 executions . Both BEFRA2OPT and BEQRA2OPT have similar performance with small advantage for BEQRA2OPT where BEQRA2OPT have lowest average mean

cost in 4/7 instances with an average ranking of 1.42 and BFR2OPT have lowest average mean cost in 3/7 instances with an average ranking of 1.57 .

Comparisons between the four proposed approaches and some heuristics and metaheuristics [BEST]

Here we compare our approaches with heuristics (Nearest neighbour, Clarke-Wright and 2-Opt heuristic) and metaheuristics (Genetic Algorithm, Artificial Bee Algorithm, Black Hole Algorithm and Ant Colony Optimization) proposed in (Maquin and al, 2025) on Best solutions. (Codes are available on Matlab file Exchange: https://fr.mathworks.com/matlabcentral/fileexchange/158496-tsp2024)

Table 11 Comparison between proposed methods and classical heuristics/metaheuristics. Best results are highlighted in bold.

Instance	NN	CW	2OPT	ABC	BHA	GA	ACO	BEFRA	BFR2OPT	BEQRA	BQR2OPT
ATT48	39760	34432	34332	36081	34668	34374	36723	36212	**33524**	37778	**33524**
EIL51	506	438	432	487	445	440	477	450	430	453	**433**
EIL76	629	570	561	627	602	562	649	584	556	605	**554**
EIL101	750	688	652	801	786	679	823	701	649	698	**646**
BERLIN52	8867	7948	7713	8409	7816	7794	8185	8081	**7544**	8008	**7544**
KROA100	26729	21985	21395	30756	27519	22387	26964	24439	**21307**	25049	21395

This table shows that hybrid approaches BEQRA2OPT and BEFRA2OPT outperform all other algorithms across all TSP Benchmarks, however BEFRA's and BEQRA's results are better than (NN,ABC,ACO) results for all benchmarks except for Att48. BEFRA's and BEQRA's results are worse than (CW,2OPT, GA) results for all benchmarks. BEFRA and BEQRA outperform BHA on large benchmarks (Eil 101, KroA100) but on Small benchmarks (att48,Eil51,Eil76,Berlin52) BHA's results are better than BEFRA's and BEQRA's results.

Table 12 Comparisons between proposed approaches and classical heuristics/metaheuristics (MEAN values). Best results are highlighted in bold.

Instance	NN	CW	2OPT	BEE	BHA	GA
ATT48	41225.00	35480.65	34817.19	42910.58	36918.71	35286.50
EIL51	546.46	452.52	438.07	529.41	473.70	461.66
EIL76	662.27	582.13	569.94	689.59	636.63	591.20
EIL101	787.48	691.71	667.53	862.70	843.36	697.95
BERLIN52	9398.68	8183.08	7820.30	9330.47	8557.16	8218.86
KROA100	27316.00	23125.20	22016.61	35912.39	30952.79	23300.33

Instance	ACO	BEFRA	BFR2OPT	BEQRA	BQR2OPT
ATT48	37862.00	40632.00	**34148.00**	39649.00	34289.00
EIL51	495.838	487.35	**437.55**	483.30	437.91
EIL76	655.095	608.24	568.68	615.91	**562.90**
EIL101	834.106	754.25	662.71	746.50	**658.00**
BERLIN52	8553.42	8559.41	7796.74	8590.46	**7768.04**
KROA100	28091.50	26461.60	22207.00	26111.30	**21991.60**

This table shows that hybrid approaches BEQRA2OPT and BEFRA2OPT outperform all other algorithms across all TSP Benchmarks also ,except for KroA100 where 2OPT is better than BEFRA2OPT. However BEFRA and BEQRA always beat NN and ABC, always lose to CW, 2OPT, GA in this dataset and have mixed results versus BHA and ACO.

Fig 8 shows solutions obtained by applying different algorithms on TSPLib benchmark (Eil 101), Fig 8(a) shows the best solution of ACO algorithm (Tour length =827.55), this solution is the best of 10000 constructed solutions by 100 ant in 100 iteration. Fig 8(b) shows BEFRA's best solution (Tour Length = 764.96) of 10 constructed solutions applying BEFRA's Algorithm (we applied BEQRA 10 times) and tours lengths are shown in Fig 8(c) (768.05,825.71,769.75,779.57,770.82,817.42,768.05,801.71,764.96,831.12), the ACO best solution is improved 9 times from 10 executions and the best solution is improved 2 times (1ST and 9th iteration). Fig 9(d) shows BEFRA_2OPT solution (tour length =661.73) obtained by applying 2_OPT heuristic on solution in Fig 8(b).

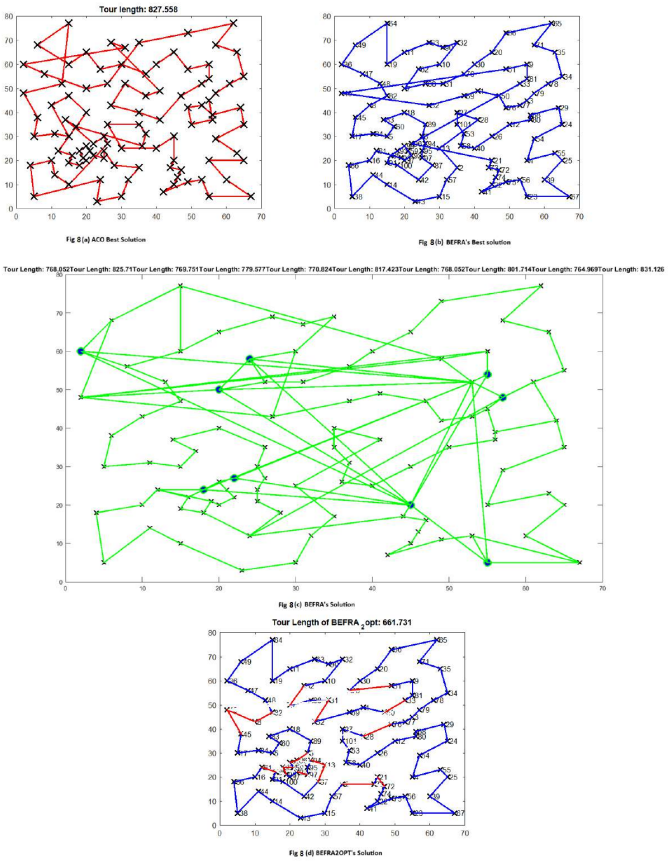


Fig. 8 Roadmap for proposed algorithms BEFRA, ,BEFRA2OPT..

Fig 9(a) shows ACO's best solution (the same in Fig 8 (a) ,BEQRA's best solution is shown in Fig 9(a)(b) (Tour Length = 747.072) of 10 constructed solutions applying BEQRA's Algorithm (we applied BEQRA 10 times) and tours lengths are shown in Fig 8(c) (761.55,756.86,763.44,767.77,758.39,747.07,807.19,783.85,794.14,755.96) , the ACO best solution is improved 10 times from 10 executions and the best solution is improved 3 times (1st , 2nd and 6th iteration). Fig 9(d) shows BEQRA2OPT solution (tour length = 652.177) obtainedby applying 2 OPT heuristic on solution in Fig 9(a)(b).

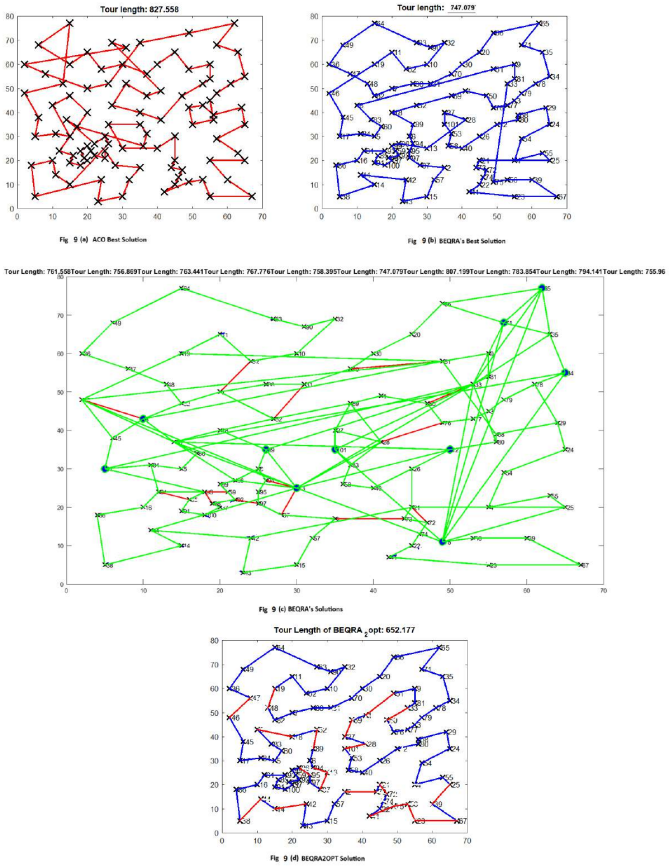


Fig. 9 Roadmap for proposed algorithms BEQRA, ,BEQRA2OPT.

Table 13 (BEFRA2OPT / BEQRA2OPT) Comparisons with IMPRV[X], ACA, Q-ACA, and IWSMACO (Xiao et al., 2018). Best results are highlighted in bold.

	IMPRV[X]		ACA		Q-ACA		IWSMACO		BFR2OPT		BQR2OPT	
Instance	B	M	B	M	B	M	B	M	B	M	B	M
ATT48	33522	33624	34078	34356	33804	34158	33524	33737	**33524**	34148	33524	34289
EIL51	426	429	445	466	433	449	426	441	429.53	437.55	**433.20**	437.91
EIL76	538	546	578	591	551	565	546	559	556.05	568.68	**553.97**	562.90
ST70	679	688	696	721	686	710	680	704	689.09	703.53	**692.07**	707.21

In this table we compare our best algorithms BEFRA2OPT and BEQRA2OPT with four algorithms (ACA, Q-ACA, IWSMACO and IMPRV[X]) (Xiao et al., 2018) Table VII shows that BEFRA2OPT and BEQRA2OPT dominate ACA and Q-ACA across all benchmarks (best and mean) except Q-ACA's best for Eil76 and ST70 where results are so close. IMPRV[X] outperform all algorithms for best and mean criteria, IWSMACO is better than our best algorithm on best criteria and remain very close for mean criteria.

11 Conclusion

This paper investigates the impact of exploiting edge-level information on the performance of population-based optimization algorithms for solving the Traveling Salesman Problem. In this context, two novel heuristics, BEFRA and BEQRA, are proposed based on edge frequency and edge quality, respectively, together with their hybrid variants enhanced by the 2-OPT local search procedure. The experimental results obtained on standard TSPLIB benchmark instances demonstrate that the proposed hybrid approaches, namely BEFRA2OPT and BEQRA2OPT, consistently outperform classical Ant Colony Optimization (ACO) and several well-established heuristics and metaheuristics in terms of both solution quality and robustness. These findings confirm that incorporating structured knowledge extracted from population-based solutions can significantly enhance optimization performance. More importantly, this study establishes, for the first time, a meaningful relationship between edge frequency and edge quality, highlighting their crucial role in guiding the search process. This contribution goes beyond incremental improvements and opens a new perspective for designing intelligent and interpretable metaheuristic frameworks. Future research will focus on extending the proposed approaches to larger-scale instances, multi-objective variants of the TSP, and other combinatorial optimization problems. Furthermore, integrating the proposed edge-based strategies with alternative population-based algorithms and more advanced local search techniques represents a promising direction for further improvement.

Future work requires the investigation of the new approaches on larger TSP instances and will be applied on other Population Based Metaheuristics like Genetic Algorithm, Artificial Bee Colony, Black Holl Algorithm, Firefly Algorithm. Other local search heuristics can be combined with BEFRA and BEQRA (K-opt heuristics). BEFRA and BEQRA will be adapted to solve the Multi-objective TSP.

Declarations

Ethics approval and consent to participate

Not applicable. This study does not involve human participants, human data, or animal subjects.

Consent for publication

Not applicable. This manuscript does not contain any individual person's data, images, or videos requiring consent for publication.

Funding

The authors received no specific funding for this work.

Availability of data and materials

The datasets used and analyzed in this study consist of publicly available Traveling Salesman Problem (TSP) benchmark instances obtained from TSPLIB. The benchmark instances are publicly available through TSPLIB. The generated experimental results and additional data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflict of interest

The authors declare that they have no competing interests.

Code availability

The implementation of the proposed algorithm is available from the corresponding author upon reasonable request.

Author contributions

All authors contributed to the conception and design of the study. Algorithm development, experimentation, data analysis, and manuscript preparation were performed by the authors. All authors reviewed, revised, and approved the final manuscript.

References

[1] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley, Chichester, UK (1985)

[2] Reinelt, G.: The Traveling Salesman: Computational Solutions for TSP Applications. Lecture Notes in Computer Science, vol. 840. Springer, Berlin, Germany (1994). https://doi.org/10.1007/3-540-48661-5

[3] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York, NY, USA (1979). https://archive.org/details/computersintract0000gare

[4] Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and

Artificial Intelligence. MIT Press, Cambridge, MA, USA (1992). https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/

[5] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4. Perth, WA, Australia, pp. 1942–1948 (1995). https://doi.org/10.1109/ICNN.1995.488968 . IEEE. https://doi.org/10.1109/ICNN.1995.488968

[6] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software **69**, 46–61 (2014) https://doi.org/10.1016/j.advengsoft.2013.12.007

[7] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. Journal of Global Optimization **39**(3), 459–471 (2007) https://doi.org/10.1007/s10898-007-9149-x

[8] Hatamlou, A.: Black hole: A new heuristic optimization approach for data clustering. Information Sciences **222**, 175–184 (2013) https://doi.org/10.1016/j.ins.2012.08.023

[9] Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation **1**(1), 53–66 (1997) https://doi.org/10.1109/4235.585892

[10] Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Books. Springer, Berlin, Germany (2004). https://doi.org/10.7551/mitpress/1295.001.0001 . https://doi.org/10.7551/mitpress/1295.001.0001

[11] Stützle, T., Hoos, H.H.: Aco algorithms for the traveling salesman problem. In: Miettinen, K., Mäkelä, M.M., Neittaanmäki, P., Périaux, J. (eds.) Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, pp. 163–183. John Wiley & Sons, Chichester, UK (1999)

[12] Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms, 2nd edn. Luniver Press, Frome, UK (2010). https://www.luniver.com/contents/nature-inspired-metaheuristic-algorithms.html

[13] Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ, USA (2006). https://doi.org/10.1515/9781400841103 . https://doi.org/10.1515/9781400841103

[14] Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the travelling-salesman problem. Operations Research **21**(2), 498–516 (1973) https://doi.org/

10.1287/opre.21.2.498

[15] Ismkhan, H.: Effective heuristics for ant colony optimization to handle large-scale problems. Swarm and Evolutionary Computation **32**, 140–149 (2017) https://doi.org/10.1016/j.swevo.2016.06.006

[16] Zhang, F., *et al.*: Title of the paper by zhang et al. on classical heuristics for tsp. Journal Name **00**(0), 000–000 (2018)

[17] Gunay-Sezer, N.S., Cakmak, E., Bulkan, S.: A hybrid metaheuristic solution method to traveling salesman problem with drone. Systems **11**(5), 259 (2023) https://doi.org/10.3390/systems11050259

[18] Hao, T., Yingnian, W., Jiaxing, Z., Jing, Z.: Study on a hybrid algorithm combining enhanced ant colony optimization and double improved simulated annealing via clustering in the traveling salesman problem (tsp). PeerJ Computer Science **9**, 1609 (2023) https://doi.org/10.7717/peerj-cs.1609

[19] Scianna, M.: The addaco: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems. Mathematics and Computers in Simulation **218**, 357–382 (2024) https://doi.org/10.1016/j.matcom.2023.12.003

[20] Tuani, A.F., Keedwell, E.: Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. Applied Soft Computing **97**, 106720 (2020) https://doi.org/10.1016/j.asoc.2020.106720

[21] Ning, J., Zhang, C., Sun, P., Feng, Y.: Comparative study of ant colony algorithms for multi-objective optimization. Information **10**(1), 11 (2019) https://doi.org/10.3390/info10010011

[22] Hertono, G.F., Handari, B.D., Fitriana, I., *et al.*: A hybrid ant colony optimization for the traveling salesman problem. In: Journal of Physics: Conference Series, vol. 1108, p. 012111 (2018). https://doi.org/10.1088/1742-6596/1108/1/012111 . IOP Publishing. https://doi.org/10.1088/1742-6596/1108/1/012111

[23] Zhang, Z., Gao, C., Lu, Y., Liu, Y., Liang, M.: Multi-objective ant colony optimization based on the physarum-inspired mathematical model for bi-objective traveling salesman problems. PLoS ONE **11**(1), 0146709 (2016) https://doi.org/10.1371/journal.pone.0146709

[24] Garcia-Martinez, C., Cordón, O., Herrera, F.: A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria tsp. European Journal of Operational Research **180**(1), 116–148 (2007) https://doi.org/10.1016/j.ejor.2006.03.041

[25] Lin, P., Zhang, J., Contreras, M.A.: Applying pareto ant colony optimization to solve bi-objective forest transportation planning problems. In: 2014 IEEE 15th International Conference on Information Reuse and Integration (IRI), pp. 795–802 (2014). https://doi.org/10.1109/IRI.2014.7051970 . IEEE. https://doi.org/10.1109/IRI.2014.7051970

[26] Zhang, X., Wang, Q., Adamatzky, A., Chan, F.T.S.: An improved physarum polycephalum algorithm for the shortest path problem. The Scientific World Journal **2014**, 487069 (2014) https://doi.org/10.1155/2014/487069

[27] Ekanayake, E.M.U.S.B.: To use an ant colony technique to solve a crispy type bi- and tri-objective transportation problem. Journal of Computational and Cognitive Engineering **2**(4), 294–303 (2023) https://doi.org/10.47852/bonviewJCCE3202378

[28] Ouhader, H., El Kyal, M., El Hassani, A.H., Benhlima, L.: Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Applied Soft Computing **37**, 196–203 (2015) https://doi.org/10.1016/j.asoc.2015.08.020

[29] Zhou, Y., Lai, X., Li, Y., Dong, W.: Ant colony optimization with combining gaussian eliminations for matrix multiplication. IEEE Transactions on Systems, Man, and Cybernetics: Systems **43**(1), 347–357 (2013) https://doi.org/10.1109/TSMCB.2012.2207717

[30] Cai, X., Wang, Y., Zhao, M., Li, J.: Mobile robot path planning based on ant colony optimization with a* heuristic method. Frontiers in Neurorobotics **13**, 15 (2019) https://doi.org/10.3389/fnbot.2019.00015

[31] Sui, J., Li, Y., Huang, H.: Improved reputation evaluation for reliable federated learning on blockchain. IET Communications **18**, 421–428 (2024) https://doi.org/10.1049/cmu2.12743

[32] Lu, J., Lin, R.: High-performance language parallel database processing learning based on ant colony optimization algorithm of swarm intelligence. Concurrency and Computation: Practice and Experience (2025) https://doi.org/10.1177/18724981251324023

[33] Abdelbar, A.M., Wunsch II, D.C.: Training neural networks with a self-adaptive ant colony algorithm. Neural Computing and Applications **37**, 19773–19791 (2025) https://doi.org/10.1007/s00521-025-11423-y

[34] Saini, R., Mani, A., Prasad, M.S., Bhattacharyya, S., *et al.*: Evolutionary algorithm for the traveling salesman problem with innovative encoding on hybrid quantum-classical machines. IEEE Access (2025) https://doi.org/10.1109/ACCESS.2025.3554690

[35] Mohammed, H., Ibrahim, M., Raoof, A., Jaleel, A., Al-Dujaili, A.Q.: Modified

ant colony optimization to improve energy consumption of cruiser boundary tour with internet of underwater things. Computers **14**(2), 74 (2025) https://doi.org/10.3390/computers14020074

[36] Tirkolaee, E.B., Goli, A., *et al.*: Multi-visit vehicle routing problem with multiple heterogeneous drones. Transportation Research Part C: Emerging Technologies **172**, 105026 (2025) https://doi.org/10.1016/j.trc.2025.105026

[37] Zhu, L.: Hybrid crow-wolf for industrial iot. Future Generation Computer Systems (2025)

[38] Uddin, M.M., Hasan, M.M., Rahman, M.A.: An improved heuristic mechanism ant colony optimization algorithm for solving path planning. Knowledge-Based Systems **271**, 110540 (2023) https://doi.org/10.1016/j.knosys.2023.110540

[39] Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, MA (2004)

[40] Scianna, M.: The addaco: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems. Mathematics and Computers in Simulation **218**, 357–382 (2024) https://doi.org/10.1016/j.matcom.2023.12.014

[41] Ibrahim, T., Fayeez, A., Keedwell, E., Collett, M.: Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. Applied Soft Computing **97**, 106720 (2020) https://doi.org/10.1016/j.asoc.2020.106720

[42] Xiao, J., Qian, Y., Zhang, H., Liang, J.: Ant colony optimization with partial-complete searching for attribute reduction. Journal of Computational Science **25**, 170–182 (2018) https://doi.org/10.1016/j.jocs.2017.05.007

[43] Hertono, G.F., Ubadah, Handari, B.D.: The modification of hybrid method of ant colony optimization, particle swarm optimization and 3-opt algorithm in traveling salesman problem. Journal of Physics: Conference Series **974**(1), 012032 (2018) https://doi.org/10.1088/1742-6596/974/1/012032

[44] Zhang, X., Adamatzky, A., Yang, X.-S., Yang, H., Mahadevan, S., Deng, Y.: A physarum-inspired approach to supply chain network design. Science China Information Sciences **59**(5), 052203 (2016) https://doi.org/10.1007/s11432-015-5417-4

[45] Chen, G., Liu, J.: Mobile robot path planning using ant colony algorithm and improved potential field method. Computational Intelligence and Neuroscience **2019**, 1932812 (2019) https://doi.org/10.1155/2019/1932812

[46] Sarucan, A., Berkaya, M.F.: Combining 3-opt and improved discrete cuckoo search

algorithm for the traveling salesman problem. Discrete Dynamics in Nature and Society **2024**, 1953277 (2024) https://doi.org/10.1155/2024/1953277

[47] Lin, A.: Title of the lin article. Journal Name **1**, 1–10 (2024)

[48] Ekanayake, E.M.U.S.B.: To use an ant colony technique to solve a crispy type bi- and tri-objective transportation problem. Journal of Computational and Cognitive Engineering **2**(4), 294–303 (2023) https://doi.org/10.47852/jcce3202378

[49] Psychas, I., Samaras, N., Papakostas, G., Tzovaras, D.E.: Ant colony optimization-based feature selection for classification problems. In: Proceedings of the 7th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–6 (2015). https://doi.org/10.1109/IISA.2016.7785426

[50] Hao, J., Zhang, J., Li, W., Wang, P.: Study on a hybrid algorithm combining enhanced ant colony optimization and double improved simulated annealing via clustering in the traveling salesman problem. PeerJ Computer Science **9**, 1396 (2023) https://doi.org/10.7717/peerj-cs.1396

[51] Zhu, X.: Hybrid ant colony and whale optimization algorithms for qos-aware loading balancing in cloud computing environment. Journal of Engineering and Applied Science **72**, 169 (2025) https://doi.org/10.1186/s44147-025-00739-0

[52] Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, ??? (1992)

[53] Rajan, A., Sharma, S., Schrammel, P., Kroening, D.: Accelerated test execution using gpus. In: Proceedings of the ACM/IEEE International Conference on Automated Software Engineering (ASE), pp. 97–102 (2014). https://doi.org/10.1145/2642937.2642957

[54] Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **26**(1), 29–41 (1996) https://doi.org/10.1109/3477.484436

[55] Larsson, J.: Ant colony optimization: A survey on recent developments and applications. Artificial Intelligence Review **49**(4), 523–545 (2018)

[56] Chen, E., Li, Y., Zhang, J.: An improved ant colony optimization algorithm for solving the traveling salesman problem. Procedia Engineering **15**, 1561–1565 (2011) https://doi.org/10.1016/j.proeng.2011.08.290

[57] Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J.: A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing **8**(2), 239–287 (2009) https://doi.org/10.1007/s11047-008-9098-4

[58] Gendreau, M., Potvin, J.-Y. (eds.): Handbook of Metaheuristics, 2nd edn. Springer, New York, NY, USA (2010). https://doi.org/10.1007/978-1-4419-1665-5

[59] Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ, USA (2006). https://doi.org/10.1515/9781400841103 . https://doi.org/10.1515/9781400841103