algorithms

MDPI

Review

A Comprehensive Review of Metaheuristics for the Modern Traveling Salesman Problem and Drone-Assisted Delivery

Alessio Mezzina [1,2] and Mario Pavone [1,3,*]

[1] Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria 6, 95125 Catania, Italy; alessio.mezzina@phd.unict.it
[2] Koexai S.R.L., Via Jose Maria Escriva 6, 95125 Catania, Italy
[3] ANTs Lab—Advanced New Technologies Research Laboratory, Via Rigolato 51, 95024 Acireale, Italy
* Correspondence: mpavone@dmi.unict.it

Abstract

The Traveling Salesman Problem (TSP) is a fundamental challenge in combinatorial optimization, with wide-ranging applications in logistics, manufacturing, and network design. In addition to the classical formulation, recent years have witnessed the emergence of complex variants, such as the TSP with Drones (TSP-D), TSP with Time Windows, and Prize-Collecting TSP, that incorporate novel constraints reflecting real-world requirements like last-mile delivery and multimodal logistics. This review presents a comprehensive survey of metaheuristic approaches for solving both the classical TSP and its emerging extensions, with particular emphasis on metaheuristic, hybrid methods, and machine learning-enhanced strategies. Recent algorithmic developments, benchmark datasets, and evaluation metrics are investigated, and critical challenges in addressing drone coordination, synchronization, and uncertainty are identified, as well. Bibliometric analysis is further provided to map research trends and the evolution of the field. By synthesizing foundational techniques and state-of-the-art innovations, this review outlines current progress and proposes future directions for metaheuristic optimization in increasingly dynamic and heterogeneous routing scenarios.

Keywords: Traveling Salesman Problem; TSP variants; multiple TSP; TSP with time windows; TSP with Drones; metaheuristics; hybrid algorithms; machine learning; optimization

1. Introduction

In recent decades, improvements in computational power and the development of new optimization algorithms have greatly influenced the field of combinatorial optimization. Among these, the Traveling Salesman Problem (TSP) is particularly notable because it poses significant theoretical challenges while also offering practical benefits. The TSP seeks the most efficient route that visits each location exactly once before returning to the starting point, with the goal of reducing overall distance, time, cost, or energy consumption. Due to its extensive applications in logistics, transportation, manufacturing, and network design [1,2], the TSP has maintained a high level of interest within the optimization community. In parcel delivery, for instance, the sequence in which customers are served greatly influences delivery times, fuel use, and overall system reliability and efficiency [3]. Therefore, modest improvements in route optimization can lead to substantial cost savings and lower emissions when implemented on a large scale [4]. TSP is then essential in several specialized domains such as circuit board drilling [5], genome sequencing [6], telescope

Academic Editors: Angel A. Juan and Nebojsa Bacanin

Received: 21 February 2026
Revised: 19 March 2026
Accepted: 21 March 2026
Published: 2 April 2026

scheduling [7], and robotic inspection [8]. In addition, TSP is also increasingly relevant in the context of drone-enabled logistics. The integration of drones into delivery systems has revolutionized last-mile operations by enabling rapid and efficient parcel delivery in both urban and rural areas. Drones can navigate directly to delivery points, bypassing ground traffic to substantially reduce delivery times. However, their limited payload capacity and battery life introduce new complexities to routing problems. These operational challenges have given rise to the *Traveling Salesman Problem with Drones* (TSP-D) and its related truck-drone routing variants, in which a truck launches and retrieves one or more drones while continuing its own route [9].

For this reason, in addition to reviewing the latest advances in the classical TSP and its variants, this article places particular emphasis on the TSP-D. By focusing on this emerging variant, this review aims to highlight the unique challenges, solution methodologies, and practical applications associated with drone-enabled delivery systems, thereby providing a comprehensive perspective on the state-of-the-art in this dynamic and rapidly evolving field. This article therefore aims to provide a comprehensive review of metaheuristic approaches for solving the Traveling Salesman Problem and its variants, with a particular focus on the integration of machine learning techniques. The main goals of this work are to:

1. summarize the state-of-the-art metaheuristics used for solving TSP and its variants;
2. discuss ML technique integration into metaheuristic frameworks for enhanced the optimization performance;
3. identify current trends, challenges, and future research directions in the application of metaheuristics to the TSP, with particular focus on one of the TSP variants, such as TSP-D.

1.1. Background

The Traveling Salesman Problem is the most-studied combinatorial optimization problem that seeks to find the shortest possible route that visits a set of cities exactly once and returns to the starting city. Formally, given a complete weighted graph $G = (V, E)$, with V being the set of n cities and $c : V \times V \to \mathbb{R}^+$ being the cost function representing the distance between each pair of cities $(u, v) \in E$ (with $u, v \in V$), the goal is to find a permutation $\pi = (\pi_1, \ldots, \pi_n)$ with $\pi_i \in V$ $(i = 1, \ldots, n)$ that minimizes the total cost of the tour; that is:

$$\min \sum_{i=1}^{n} c(\pi_i, \pi_{i+1}), \tag{1}$$

with $\pi_{n+1} = \pi_1$ to ensure the tour forms a closed loop. Despite its straightforward formulation, the TSP is NP-hard [10], and exact methods quickly become infeasible as problem size increases. As a result, the TSP has long served as a benchmark for evaluating optimization algorithms.

Metaheuristic algorithm design for the TSP has evolved significantly over the years, with a focus on developing efficient heuristics that can provide high-quality solutions within reasonable time frames. Early approaches relied on simple heuristics such as nearest-neighbor and insertion methods, which, while effective for small instances, often fail to deliver optimal solutions for larger problems [11]. In recent years, however, hybrid metaheuristics have garnered significant attention for their ability to synergize complementary algorithmic strategies. Indeed, recent advancements in metaheuristics have demonstrated their effectiveness in handling dynamic optimization problems [12]. Notably, integrating Machine Learning (ML) techniques into metaheuristics has emerged as a transformative strategy, enabling them to learn from past search experiences and dynamically enhance their performance [13–15]. Research indicates that incorporating ML methods, such as reinforcement learning, surrogate modeling, and adaptive parameter tuning, can substantially improve performance by optimizing the search process, accelerating convergence, and equipping metaheuristics to tackle increasingly complex and dynamic optimization challenges [16–18]. Consequently, the integration of ML techniques into metaheuristic frameworks represents a promising direction for advancing optimization methodologies in the context of the TSP and its variants [19]. Several studies have already demonstrated that ML-enhanced metaheuristics can outperform traditional methods in terms of both solution quality and computational efficiency [20,21].

Beyond classical benchmark instances, TSP has been extensively promoted and illustrated by large-scale and creative demonstrations. A notable example is the project of mapping an optimal or near-optimal tour that visits every city on Earth above a certain population threshold [22], effectively showcasing the capacity of state-of-the-art TSP solvers to handle tens or even hundreds of thousands of nodes. In another demonstration, a "star map TSP" [23] was constructed by treating individually cataloged stars as cities, underscoring how the TSP framework can be generalized to a broad variety of data points in metric spaces beyond terrestrial geography. In a different direction, Robert Bosch has created a fascinating series of TSP instances that provide continuous-line drawings of well-known artwork [24–26]. Techniques for generating these point sets have evolved over the past several years through work by Bosch and Craig Kaplan [27]. One particularly famous instance is the *Mona Lisa TSP Challenge*, shown in Figure 1, comprising 100,000 cities: the TSP tour on these points recreates Leonardo da Vinci's iconic portrait in a single continuous stroke. Additional art-inspired TSP instances include van Gogh's *Self Portrait 1889*, Botticelli's *The Birth of Venus*, Velázquez's *Juan de Pareja*, Courbet's *The Desperate Man*, and Vermeer's *Girl with a Pearl Earring*. These data sets [28] range in size up to 200,000 cities and serve as large-scale visually engaging test problems for TSP solvers. Table 1 provides an overview of the available files and their sizes.

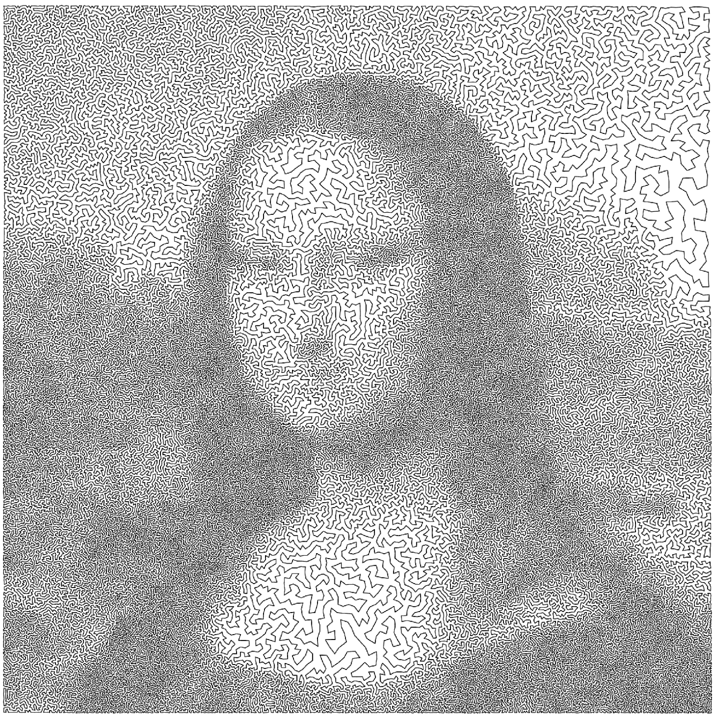

Figure 1. TSP solution for da Vinci's *Mona Lisa* (100,000 cities) [24].

Table 1. TSP art instances by Robert Bosch and collaborators, available at [28].

Original Art	Data Set	Cities
Da Vinci's Mona Lisa	mona-lisa100K.tsp	100,000
Van Gogh's Self Portrait 1889	vangogh120K.tsp	120,000
Botticelli's The Birth of Venus	venus140K.tsp	140,000
Velázquez's Juan de Pareja	pareja160K.tsp	160,000
Courbet's The Desperate Man	courbet180K.tsp	180,000
Vermeer's Girl with a Pearl Earring	earring200K.tsp	200,000

Such demonstrations have been popularized on the *Concorde* TSP website [29], where large-scale tours are visualized and the solver's underlying methods are outlined. The website also highlights record-breaking instances solved, including routes involving thousands of cities across the globe or novel domains such as pixel-level images (i.e., for representing art), three-dimensional point clouds, or astrophysical data sets. Over the decades, substantial efforts have been devoted to exact algorithms for the TSP. Established approaches include branch-and-bound [30], branch-and-cut [31], and cutting-plane techniques [32]. Implementations such as the *Concorde* [29] solver have been highly successful for instances with several thousands of cities [33], although the running time can become prohibitive as problem size grows. The Held–Karp dynamic programming algorithm, with time complexity $O(n^2 2^n)$, provides a theoretical framework but is limited in practice to relatively small n [34,35]. In light of this, exact methods are rarely the first choice when building large-scale or time-sensitive applications. The TSP has served as a canonical testbed for developing and analyzing new algorithmic paradigms [33,36]. Its core requirement, to find a single-cycle tour visiting each node exactly once, captures fundamental challenges in network design, logistics, and manufacturing.

1.2. Scope and Methodology

To ensure a comprehensive and unbiased survey, we conducted a systematic literature review across major scientific databases, including Scopus. This review focuses on peer-reviewed journal articles published between 2015 and April 2025. Studies have been included based on their relevance to the development, analysis, or application of metaheuristic algorithms to the TSP and its variants.

To better understand the scientific context in which metaheuristics for the TSP are studied, we conducted a preliminary bibliometric analysis using the Scopus database. The query was designed to retrieve publications related to the TSP and metaheuristics from 2015 to early 2025. Figure 2 shows the number of publications per year. The data clearly indicate a growing trend, with a marked increase in 2018 and 2020, and a peak in 2024 (>180 articles). The apparent drop in 2025 is to be expected, as the current year has only partially unfolded and many publications may not yet be indexed.

To explore the disciplinary landscape of the field, we categorized the articles retrieved by subject area. As depicted in the left plot of Figure 3, most of the research is concentrated in Computer Science, followed by Engineering and Mathematics. It is worth noting that Decision Sciences also contributes significantly, underscoring the relevance of TSP-related problems in practical optimization and decision-making settings. These findings are further detailed in the pie chart shown in the right plot of Figure 3, which illustrates the relative weight of each subject area among the total corpus. Computer Science alone accounts for more than 30% of the publications, highlighting the algorithmic nature of the problem. Engineering and Mathematics follow with substantial shares, while interdisciplinary contributions from areas such as Decision Sciences, Materials Science, and Social Sciences indicate the widespread applicability of TSP metaheuristics across diverse domains.

https://doi.org/10.3390/a19040278

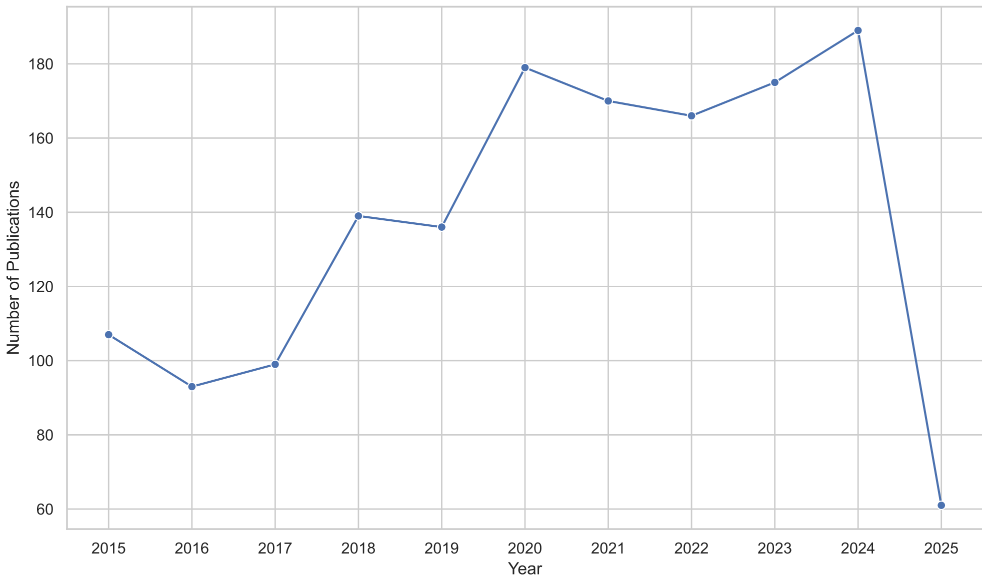

Figure 2. Number of publications per year on metaheuristics for TSP (Scopus, 2015–2025).

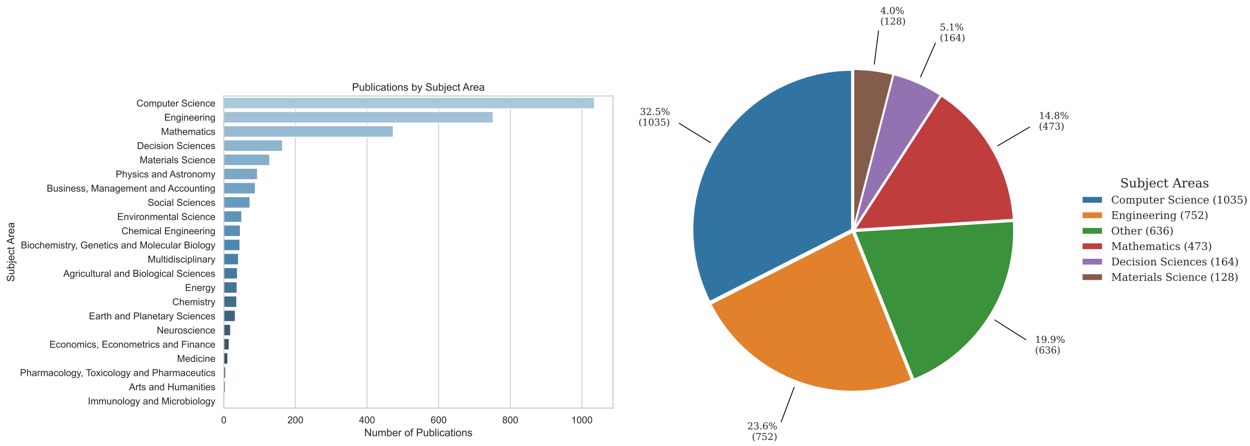

Figure 3. Number of publications (**left plot**) and relative distribution of publications (**right plot**) by subject area.

1.3. Organization of This Paper

The remainder of this article is organized as follows: Section 2 explores the classical TSP and its key variants, including mTSP, TSP-TW, PC-TSP, GTSP, and STSP, highlighting their definitions, applications, and challenges. Section 3 focuses on the TSP with Drones (TSP-D), discussing its formulation, practical variants, metaheuristic strategies, and real-world applications. Section 4 reviews metaheuristic approaches for the TSP and its variants, including a discussion of benchmark datasets and performance metrics. Section 5 examines the integration of Machine Learning with metaheuristics, showcasing hybrid strategies for complex TSP variants. Finally, Section 6 identifies open challenges and proposes future research directions in metaheuristic optimization for TSP and its variants.

2. Traveling Salesman Problem and Its Variants

For addressing the TSP, two different variants can be considered: (1) Symmetric TSP (STSP), when $c(i,j) = c(j,i)$ for all $i, j \in V$, as shown in Figure 4a, and (2) Asymmetric TSP (ATSP), when, instead, the costs can differ in each direction, $(c(i,j) \neq c(j,i))$, as shown in Figure 4b. Small and simple changes to the constraints and/or cost structure give rise to a series of TSP variants, such as the Vehicle Routing Problem (VRP), the *TSP with Time*

Windows (TSP-TW), and the Orienteering Problem (OP), each of which reflects distinct real-world demands [37]. Even relatively small instances of any of these variants can become computationally intractable when attempting to find provably optimal solutions. This difficulty has therefore motivated and directed substantial research into metaheuristics, which are high-level strategies that guide lower-level heuristics to escape local optima and broadly explore the solution space [38]. Such algorithms include, for instance, Genetic Algorithms, Simulated Annealing, Tabu Search, and Ant Colony Optimization, among others [39–42]. Typically, these algorithms generate near-optimal solutions in reasonable time on large instance dimensions, unlike exact methods that instead may struggle or be unable to find optimal solutions. As we will discuss in Section 4.1, metaheuristics are especially well-suited for TSP variants characterized by additional constraints or specialized cost structures (e.g., capacity limits, time windows, or multi-objective criteria). In these cases, balancing solution quality and computational feasibility becomes paramount [37,43].

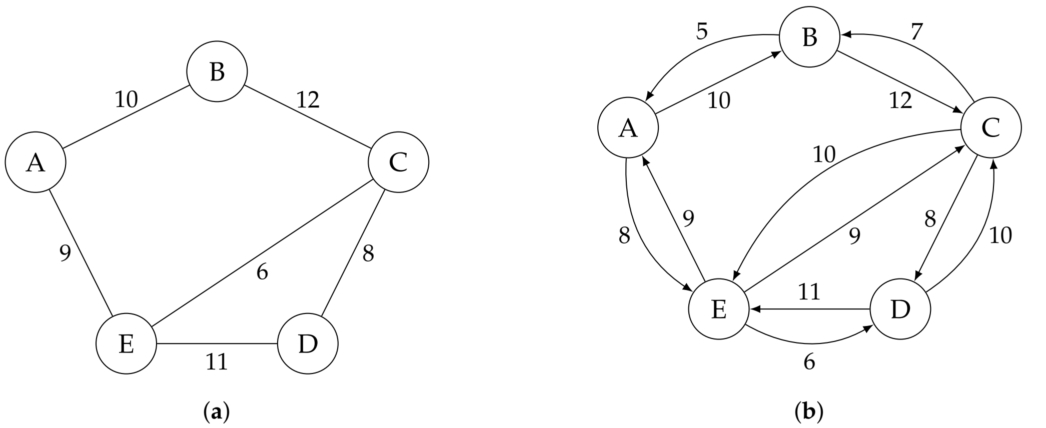

Figure 4. Example of both comparison symmetric and asymmetric TSP instances. (**a**) Symmetric TSP: Equal cost in both directions. (**b**) Asymmetric TSP: Different costs in each direction.

By embedding problem-specific knowledge (e.g., tailored neighborhood moves, problem-aware crossover operators) within a robust metaheuristic framework, researchers and practitioners can address large-scale and highly constrained TSP variants that might otherwise exceed the practical limits of exact optimization. While the classical TSP remains a central benchmark for algorithmic design, many real-world scenarios involve additional constraints or generalizations. These give rise to a rich family of TSP variants, each modeling different operational requirements such as multiple agents, time constraints, partial visits, or probabilistic availability. In the following subsections, key TSP variants that have received significant attention in both the theoretical and applied optimization communities are discussed. We will start by analyzing the scenario where we have more "salesmen".

2.1. Multiple Traveling Salesman Problem (mTSP)

Multiple Traveling Salesman Problem (mTSP) is a generalization of the classical TSP in which $m \geq 1$ salesmen, all starting and ending at a common depot, must collectively visit a set of n cities such that each city is visited exactly once by one salesman, an example of mSTP is shown in figure 5. The goal is to minimize either the total distance traveled by all salesmen or the length of the longest individual tour (min–max variant), depending on the application context [44]. This variant is particularly relevant in real-world scenarios involving fleet coordination, such as multi-vehicle logistics, distribution systems, and ride-sharing platforms. From a modeling perspective, the mTSP can be formulated as a Mixed-Integer Linear Program (MILP) with subtour elimination constraints, although exact methods quickly become impractical as n and m grow [45].

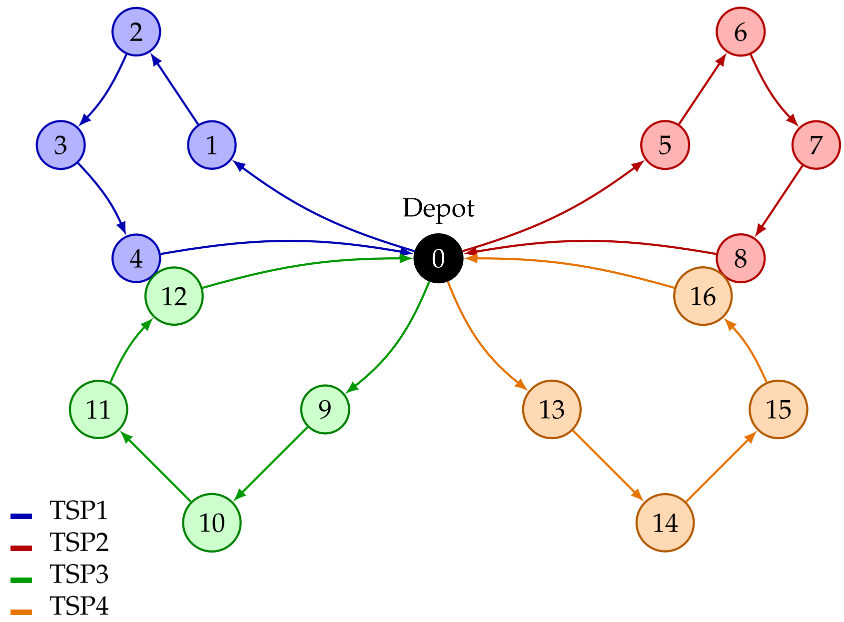

Figure 5. Illustration of a multi-TSP solution with four salesmen and a single depot (node 0).

Metaheuristic adaptations for the mTSP require algorithmic innovations that extend beyond those developed for the classical TSP. A core challenge is the encoding of multiple disjoint tours within a single chromosome in Genetic Algorithms (GAs). Among the most widely used approaches are delimiter-based representations, which explicitly define tour boundaries for each salesman. A recent study by Mahmoudinazlou and Kwon [46] proposes, for instance, a hybrid genetic algorithm where individuals are represented by a TSP sequence. This sequence is evaluated using a dynamic programming algorithm that determines the optimal set of mTSP tours. Their method introduces a novel crossover operator designed to combine similar subtours from parent solutions, thereby promoting diversity while preserving feasibility. The algorithm is further enhanced with self-adaptive local search procedures that refine the offspring quality.

In a complementary line of research, Lupoaie et al. [47] integrate Self-Organizing Maps (SOMs), Evolutionary Algorithms, and Ant Colony Systems to address the min–max variant of the single-depot mTSP. Their work illustrates the potential of hybrid approaches that merge neural network paradigms with metaheuristic strategies to improve exploration and solution quality. Specialized crossover and mutation operators are crucial not only for preserving the feasibility of solutions but also for maintaining population diversity across multiple tours. For swarm-based metaheuristics, such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), sets of cooperative agents explore different regions of the solution space in parallel. These agents coordinate through decentralized information-sharing mechanisms, such as pheromone trails in ACO or shared velocity and position updates in PSO [48]. Tang et al. [49] propose an adaptive ACO framework tailored to large-scale TSP instances, featuring dynamic pheromone updates and embedded local search strategies to accelerate convergence and enhance solution quality. Furthermore, hybrid ACO-PSO methods are introduced to exploit the complementary strengths of both paradigms when applied to the mTSP.

Local search heuristics play a crucial role in improving candidate solutions by refining tour structures. Techniques such as multi-tour, 2-opt and inter-tour node exchanges are commonly applied to balance the workload among salesmen and reduce overall travel cost, contributing to both solution feasibility and optimality [50,51]. Beyond the coordination of multiple agents in mTSP, another class of real-world constraints arises from temporal limitations imposed on service times. In many delivery and routing scenarios, customers must be visited within specific time intervals, motivating the formulation of the Traveling Salesman Problem with Time Windows (TSP-TW). This variant introduces additional feasibility challenges and requires specialized algorithms that balance routing efficiency with strict timing constraints.

2.2. Traveling Salesman Problem with Time Windows (TSP-TW)

Traveling Salesman Problem with Time Windows (TSP-TW) extends the classical TSP by introducing temporal constraints. In this variant, each city i has to be visited within a specified time window $[a_i, b_i]$. If the salesman arrives before a_i, they must wait until the window opens; similarly, arriving after b_i time renders the solution infeasible. This problem is particularly relevant in logistics and transportation scenarios where deliveries or services are constrained by customer availability or regulatory time frames.

Formally, TSP-TW can be defined on a complete directed graph $G = (V, E)$, where $V = \{0, 1, \ldots, n\}$ represents the set of nodes, with the node 0 denoting the depot; each node $i \in V$ has assigned a time window $[a_i, b_i]$ and a service time s_i; and each edge $(i, j) \in E$ has an associated travel time t_{ij} and travel cost c_{ij}. The goal is therefore to determine a minimum-cost Hamiltonian circuit, starting and ending at the depot, such that each node i is visited exactly once within its time window $[a_i, b_i]$. Solving the TSP-TW is particularly challenging due to the feasibility constraints imposed by time windows. A feasible solution must respect the temporal availability of each node, requiring that any route planning technique carefully coordinate arrival and departure times. In practice, violations are often penalized through soft constraints when strict feasibility is relaxed. An example of the TSP-TW is shown in Figure 6.

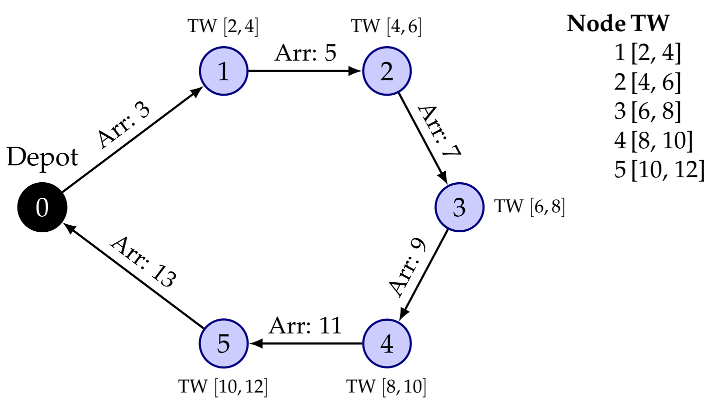

Figure 6. Illustration of a TSP with time windows (TSP-TW). Nodes 1–5 each have a specific time window (TW), and the labeled arrival times along edges show when the salesman arrives at each node. The depot (node 0) is visually distinguished.

Recent research has explored a variety of approaches to tackle the TSP-TW. Traditional metaheuristics such as Genetic Algorithms (GAs) and Ant Colony Optimization (ACO) have been successfully applied, often integrated with local search or constraint repair techniques to maintain feasibility. Alharbi et al. [52] proposed a deep learning-based method using hybrid pointer networks, incorporating time features to predict route sequences that respect time constraints. Their model outperformed classical heuristics on medium-scale instances. To address uncertainty in travel times, Bartolini et al. [53] developed a robust optimization framework for TSP-TW using a knapsack-constrained uncertainty set. Their method applies column generation and dynamic programming, yielding high-quality solutions even under travel time variability. Time-dependent travel costs have also been investigated. Vu et al. [54] introduced new formulations and lower bounds for a generalized time-dependent TSP-TW, where travel times vary with departure time. They proposed a branch-and-price algorithm capable of solving complex dynamic cost scenarios.

Ban and Pham [55] tackled both TSP-TW and the Traveling Repairman Problem using a multifactorial evolutionary algorithm. Their approach applies randomized variable neighborhood search and knowledge transfer between problem variants, resulting in a competitive solution quality across both tasks. Incorporating learning-based components, Chen et al. [56] presented a method that uses look-ahead strategies to improve the legality of constructed solutions. Their model simulates the downstream effects of current routing decisions, improving both solution feasibility and overall tour cost. Holliday et al. [57] explored quantum computing paradigms by applying D-Wave's hybrid quantum-classical solver to the TSP-TW. They demonstrated that quantum annealing, when combined with classical heuristics for time-window repair, can yield competitive results, especially in constrained, small-sized instances.

Despite the practical relevance of the TSP-TW in real-world logistics and scheduling problems, our bibliometric analysis suggests that it remains a relatively niche topic within the broader TSP research landscape. A targeted query on Scopus for journal articles explicitly focused on "TSP with Time Windows" or "TSP-TW" published between 2015 and 2024 yielded only a few articles (13). As shown in Figure 7, several years (e.g., 2017, 2021, and 2022) featured no indexed publications on the subject. Even in recent years, the number of articles per year remained very modest, typically two. This limited publication volume contrasts sharply with the surge of interest in more general metaheuristic approaches for the classical TSP, indicating that TSP-TW has yet to attract widespread or sustained attention despite its theoretical and practical significance.

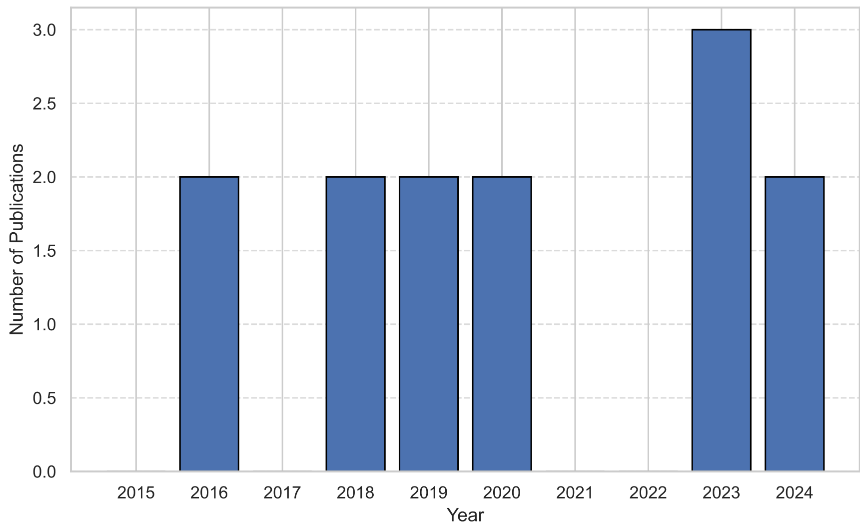

Figure 7. Number of publications per year on TSP with Time Windows (Scopus, 2015–2024).

2.3. *Prize-Collecting Traveling Salesman Problem (PC-TSP)*

The Prize-Collecting Traveling Salesman Problem (PC-TSP) is another generalization of the classical TSP that relaxes the requirement to visit all nodes. In this problem, indeed, each node $i \in V$ is associated with a non-negative prize p_i, and the goal is to construct a tour that minimizes the sum of the tour cost and penalties for uncollected prizes. More formally, the goal in solving PC-TSP is to find a subset $S \subseteq V$, with $0 \in S$, and a tour visiting nodes in S, such that:

$$\text{Minimize} \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{i \notin S} p_i, \quad (2)$$

where $x_{ij} \in \{0,1\}$ indicates whether the edge (i,j) is in the tour and c_{ij} is the cost of traveling from node i to j. The challenge lies in balancing the cost of traveling versus the benefit of collecting prizes [58]. An illustration of this variant is proposed in Figure 8.

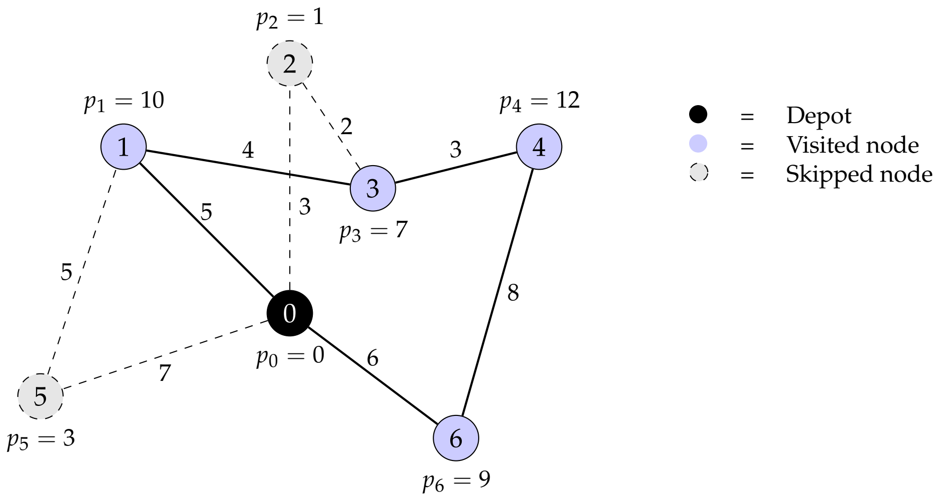

Figure 8. Illustration of a PC-TSP instance. The chosen route is $0 \to 1 \to 3 \to 4 \to 6 \to 0$, visiting a subset of nodes to balance route cost and prize collection. Nodes 2 and 5 are skipped, incurring their respective prize penalties.

This variant introduces a bi-criteria decision-making problem: it is not only about minimizing cost but also about maximizing the utility of visited nodes. The PC-TSP has several practical applications, such as: emergency response planning, where only a subset of critical sites can be visited under time or resource constraints [59]; surveillance and patrolling, which require prioritizing higher-value targets when not all can be covered [60,61]; budget-constrained delivery, selecting profitable customers to serve within a limited delivery route [62,63]. Solving the PC-TSP is more complex than the classical TSP due to the combinatorial explosion introduced by optional node visitation. The solution space becomes significantly larger, as every subset of nodes represents a candidate route, increasing the difficulty of finding near-optimal tours [64,65].

Metaheuristic algorithms for PC-TSP adapt classical TSP approaches with new mechanisms to handle the prize-versus-cost tradeoff. Notable strategies include hybrid metaheuristics, like the one presented in Ref. [66] that introduces a hybrid metaheuristic known as Clustering Search (CS), which combines clustering techniques with evolutionary algorithms. Modified acceptance criteria in Simulated Annealing (SA) [66] within the Clustering Search framework accounts for the prize-versus-cost tradeoff, thus allowing the algorithm to probabilistically accept solutions that may have longer routes but higher collected prizes, especially in the early stages of the search. Also, Tabu Search (TS)was tailored in Ref. [67] to tackle the PC-TSP. The authors adapted the TS algorithm with specialized moves, evaluation metrics, and memory structures to effectively solve the PC-TSP. Their method dynamically explores the trade-off between travel cost and prize collection and demonstrates strong empirical performance on standard benchmarks.

Other methodologies include multi-objective formulations or weighted-sum approaches to simultaneously optimize for total cost and collected prize, depending on user-defined priorities [63,68–70]; pareto-based evolutionary techniques, such as NSGA-II or MOEA/D, that maintain a diverse set of solutions along the tradeoff frontier between cost and reward [71]; or adaptive penalization schemes where the algorithm dynamically adjusts the importance of missing a node based on convergence behavior [72,73]. A representative work is provided by Feillet et al. [74], who compare several exact and heuristic approaches for the PC-TSP and propose benchmark instances, which are available in the TSPLIB [75] but whose standardized performance metrics remain sparse. Common evaluation criteria include: (*i*) total tour cost, (*ii*) collected prize, (*iii*) prize-to-cost ratio, and (*iv*) gap to best-known solutions.

2.4. Generalized Traveling Salesman Problem (GTSP)

The Generalized Traveling Salesman Problem (GTSP) is an extension of the classical TSP in which the set of cities (nodes) is partitioned into m disjoint clusters and the goal is to find a minimum-cost tour that visits exactly one node from each cluster [76]. This model captures real-world scenarios where locations are grouped by geographic regions, functional types, or availability, and the precise selection of a representative node from each cluster introduces additional combinatorial complexity. Formally, let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of nodes, partitioned into m clusters $C_1, C_2, \ldots, C_m$, such that

$$\bigcup_{i=1}^{m} C_i = V \text{ and } C_i \cap C_j = \emptyset \text{ for } i \neq j.$$

The goal is to find a tour of minimum total cost that includes exactly one node from each cluster and returns to the starting node. The GTSP generalizes the TSP and is also NP-hard [76,77], as it reduces to the TSP when each cluster contains a single node.

The GTSP finds application in several fields, such as (1) data aggregation in sensor networks, where each cluster represents a region monitored by sensors, and the tour must collect data from a single representative sensor in each cluster [78]; (2) manufacturing and inspection planning, where machines or workstations are grouped by task type or location, and only one needs to be visited per group [79–82]; and (3) communication networks, where relay stations or routers are grouped geographically or logically, and the goal is to ensure representative coverage with minimal latency [78,83–85].

From a modeling perspective, the GTSP poses a challenge due to the interdependence between cluster selection and tour optimization: the algorithm must determine not only the best sequence of clusters but also the optimal node within each cluster. Addressing this, many approaches adopt a two-phase decomposition strategy, such as (*i*) cluster-first, route-second [86], in which clusters are first reduced to representative nodes using heuristics, then the classic TSP on the reduced graph is solved; and (*ii*) route-first, cluster-second [87], where a complete tour is initially computed ignoring cluster constraints, which is then refined to satisfy the GTSP requirements.

2.5. Stochastic Traveling Salesman Problem (STSP)

The Stochastic Traveling Salesman Problem (STSP) extends the classical TSP by treating one or more problem elements, typically edge costs, travel times, or even vertex availability, as random variables. A common two-stage formulation considers a first-stage tour decision π chosen before the uncertainty $\xi \in \Omega$ is revealed and a recourse action that adapts π after realization [88]. Let $C_{ij}(\xi)$ denote the random travel cost on arc (i, j) and $\rho(\pi, \xi)$ the cost of the recourse; a widely used objective is to minimize the expected tour length, i.e.,

$$\min_{\pi \in S_n} \mathbb{E}_{\xi}\Big[\sum_{i=1}^{n} C_{\pi(i)\pi(i+1)}(\xi) + \rho(\pi, \xi)\Big], \tag{3}$$

with $\pi(n+1) = \pi(1)$. Special cases include Probabilistic TSP (PTSP), where each customer has an independent probability, and variants constrained by chance or robust variants that limit the probability of excessive delay or cost [89–91].

STSP models naturally arise in same-day delivery with uncertain customers, drone or vehicle routing under stochastic weather, service technician dispatch where requests appear dynamically, and humanitarian logistics in disaster-relief operations [92]. Evaluating a single tour often requires Monte-Carlo simulation over many scenarios, making naive enumeration prohibitive. Scenario-reduction techniques [93], sample-average approximation [94], and variance reduction estimators are therefore crucial for practical metaheuristics.

The integration of uncertainty in edge costs and node availability has motivated hybrid metaheuristics that combine classical search methods with advanced probabilistic modeling techniques. The following approaches, presented as examples, are demonstrated in closely related vehicle routing problems, highlighting their broader applicability and their methodological relevance to the STSP context:

1. Sampling-Guided GA/ACO Hybrids: Monte Carlo-based fitness evaluation coupled with local 2-opt or 3-opt has proven effective on PTSP benchmarks;
2. Adaptive Large-Neighborhood Search: Luo et al. propose an adaptive large neighborhood search heuristic to address a vehicle routing problem with stochastic demands and weight-related costs [95]. The method includes dynamic programming to evaluate expected route costs and is validated on 84 benchmark instances;
3. Joe et al. [96] introduce a novel hybrid metaheuristic for dynamic vehicle routing problems with time windows and uncertain customer demands, formulated as a route-based Markov Decision Process. Their approach, DRLSA, combines deep-reinforcement learning, specifically temporal-difference learning with experience replay, with a simulated annealing-based heuristic. This integration allows for near-instantaneous re-routing decisions in highly dynamic environments, achieving superior performance over standard Approximate Value Iteration (AVI) and Multiple-Scenario Approach (MSA), particularly when the degree of dynamism exceeds 0.5;
4. Wang et al. [97] propose a distributively robust optimization model for the Vehicle Routing Problem with Uncertain Customers (VRPUC), leveraging historical data to identify uncertain customer sets. Their approach integrates robust and stochastic paradigms via a cross-moment ambiguity set to minimize the Essential Riskiness Index (ERI), aiming to reduce both the likelihood and severity of vehicle overload. The resulting mixed-integer semidefinite programming model is solved through a branch-and-cut algorithm with a callback mechanism to enforce positive semidefiniteness. Experimental results show this hybrid method outperforms traditional stochastic programming approaches in express logistics settings.

3. Traveling Salesman Problem with Drones (TSP-D)

In this section, we focus on the main topic of this review: theTraveling Salesman Problem with Drones (TSP-D). This extension of the classical TSP integrates aerial drones with ground vehicles, enabling the optimization of routing by leveraging the complementary strengths of both systems. As a result, faster deliveries and enhanced operational flexibility are achievable, making TSP-D highly relevant for addressing the challenges of modern logistics and distribution networks.

TSP-D is gaining significant popularity in all of it forms [98]. It models the coordinated operation of a ground vehicle carrying one or more drones. While the truck follows a classical tour, the drone(s) can be launched at customer i and retrieved at customer j ($i \neq j$), provided that endurance, launch/retrieval times and synchronization constraints are respected [99]. Typical objective functions are (1) *min–sum*, minimizing the total completion time of the last returning vehicle, and (2) *min–max*, minimizing the longest individual route. TSP-D formulations vary depending on the drone/truck coordination protocol and physical limitations of the delivery system. A standard setting assumes (i) a single truck (ground vehicle) and one or more drones; (*ii*) drones are launched and retrieved at designated nodes; (*iii*) both truck and drone may travel simultaneously; (*iv*) drones operate within a fixed flight endurance or range.

The problem can be formulated as follows: let $G = (V, E)$ be a complete graph with the cost matrix C. Each drone sortie is modeled as a triple (i, j, k), where the truck launches a drone from node i, continues to node j, and picks up the drone at node k. A sortie is feasible if:

$$t_{\text{drone}}(i,k) + s_i + s_k \leq \tau, \tag{4}$$

where τ is the drone endurance and s_i, s_k are launch and recovery service times. Synchronization requires the truck to arrive at k no later than the drone. Figure 9 illustrates a TSP-D instance with a single truck and a single drone, where the truck follows a route visiting customers 1 to 5, while the drone is launched at customer 1, serves customer A and is retrieved at customer 2.

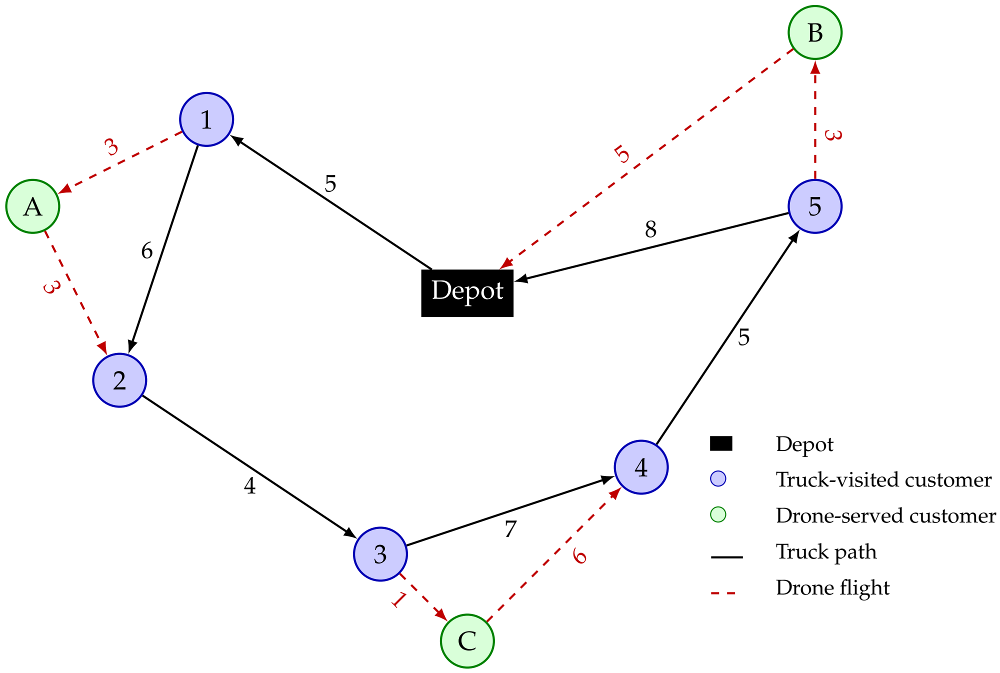

Figure 9. Illustration of a TSP-D instance with a single truck and a single drone. The truck follows a route visiting customers 1 to 5, while the drone is launched at customer 1, serves customer A, and is retrieved at customer 2. The dashed red lines represent the drone's flight paths, while the solid black lines represent the truck's route.

3.1. Problem Variants

TSP-D has several practical variants, each of which introduces specific complexities and reflects different operational priorities. While the base TSP-D often considers one truck and one drone (sometimes called FSTSP), extensions involve multiple drones (mTSP-D) operating from a single truck, or even multiple trucks each potentially carrying drones (VRPD) [100]: Single vs. Multiple Drone TSP-D. Coordinating multiple drones increases the potential for parallel deliveries but significantly increases routing complexity, requiring sophisticated synchronization and task allocation strategies to avoid conflicts and optimize fleet utilization [101,102]. Another important modeling dimension concerns the choice of objective function, particularly the contrast between *min–sum* vs. *min–max* formulation. The choice of objective function critically impacts the resulting routes. Minimizing the makespan (a min–max objective focusing on the completion time of the last vehicle) prioritizes speed of service completion [103]. Conversely, minimizing total operational cost (a min–sum objective considering factors like fuel, energy, and distance) prioritizes efficiency. These objectives can lead to different operational strategies; for instance, minimizing cost might not yield the fastest overall delivery time [104].

Uncertainty is another critical challenge, addressed in stochastic TSP-D variants. Stochastic variants model randomness in factors like drone flight times (due to weather, especially wind), battery consumption rates, or even unexpected customer unavailability [105,106]. Addressing this uncertainty requires advanced optimization techniques, such as chance constrained programming, which limits the probability of violating operational constraints (such as vehicle or drone capacity) [107], or robust distributional approaches. Developing exact solution methods like branch-and-cut-and-price for these stochastic problems presents significant challenges, particularly when considering correlations between uncertain parameters [107].

Finally, battery management is central to many TSP-D variants, given the strict endurance limits of drones. Several models incorporate explicit battery swapping or recharging strategies [108,109]. In some cases, drones may return to trucks or dedicated stations not only to pick up new packages but also to replace or recharge batteries [110]. Optimizing the timing and location of these operations adds another layer of complexity to the routing problem, and the choice between swapping and recharging can have significant implications for both total time and environmental impact [109].

3.2. Solution Techniques and Metaheuristics

The combinatorial complexity inherent in the TSP-D, arising from the coupled routing and scheduling decisions for both the truck and drone(s), significantly surpasses that of the classical TSP. Consequently, while exact methods based on mathematical programming are crucial for providing structural insights and solving smaller instances to optimality, heuristic and metaheuristic approaches dominate the literature for tackling problems of practical scale [111,112].

3.2.1. Exact Methods and Decomposition

Mixed-Integer Linear Programming (MILP) formulations provide a rigorous framework for modeling TSP-D variants [113,114]. These typically involve constraints to manage vehicle routing, drone sortie feasibility (including endurance limits), and intricate truck–drone synchronization requirements. Recently, Boccia et al. [115] introduced an arc-based MILP formulation for the Flying Sidekick TSP (FS-TSP), avoiding big-M constraints and using a polynomial number of variables. They developed a branch-and-cut algorithm that exploits vehicle coordination inequalities and drone endurance-based variable fixing strategies, achieving optimal or improved results in many instances previously unsolved in the literature. To enhance computational tractability for larger instances, decomposition techniques are frequently employed. Branch-and-price algorithms, based on set partitioning formulations, have shown promise by decomposing the problem into a master problem coordinating routes/sorties and a subproblem generating feasible truck paths and drone operations [116]. Other approaches include Benders decomposition and tailored branch-and-cut algorithms incorporating problem-specific valid inequalities [117]. Despite these advances, obtaining provably optimal solutions for medium-to-large TSP-D instances within reasonable time remains a significant computational challenge.

3.2.2. Heuristic and Metaheuristic Strategies

Given the limitations of exact methods, researchers have turned to a spectrum of heuristic and metaheuristic algorithms for TSP-D. The most straightforward belong to the family of constructive two-phase heuristics: they first compute an (almost) optimal truck-only TSP tour, then they greedily weave in feasible drone sorties where the marginal benefit is positive, often polishing the result with quick local refinements [113]. Although this sequential strategy is extremely fast, it explores only a narrow slice of the solution space (truck and drone decisions remain largely decoupled), so substantial improvements may still be left on the table. To achieve higher-quality solutions, numerous metaheuristics have been adapted or specifically designed for TSP-D. Key considerations included are:

1. Solution Representation, as encoding a TSP-D solution effectively is a non-trivial task. Strategies include multi-component representations (e.g., separate sequences for truck nodes and drone assignments) [46], integrated permutations where special markers might denote drone operations, or graph-based representations;
2. Neighborhood Structures and Operators: operators must be designed to navigate the complex solution space, including moves that modify the truck's path (e.g., 2-opt, Or-opt adapted for drone impacts), reassign customers between truck and drone, change drone launch/rendezvous nodes, or optimize drone sortie timings [114].

Specific metaheuristics applied include Genetic Algorithms (often hybridized with local search) [46,118,119], Simulated Annealing [109,120,121], Tabu Search, Ant Colony Optimization [122], and notably Adaptive Large Neighborhood Search (ALNS) [117]. ALNS, leveraging sophisticated destroy and repair operators tailored to TSP-D constraints, has proven particularly effective for various drone routing problems. Recognizing that no single metaheuristic is universally excellent, hybrid approaches combining algorithmic components are prevalent. Common examples include embedding local search within population-based frameworks, Memetic Algorithms [123], (see [124] for a comprehensive review of MAs), integrating GRASP with path-relinking [125,126], or combining different neighborhood search paradigms (e.g., VNS). Matheuristics that integrate exact optimization techniques (like MILP solvers for subproblems) within a heuristic framework (e.g., using MILP to optimize drone sorties for a fixed truck path within an ALNS iteration) represent a powerful new trend for obtaining high-quality solutions [127,128]. In particular, for routing/drones, some notable works regarding matheuristics can be found in [114,129]. The development and comparative analysis of these different solution strategies remain an active area of research, driven by the need for scalable and effective algorithms for increasingly complex drone-assisted logistics scenarios.

3.2.3. TSP-D Benchmark Instances

Unlike the classical TSP, which benefits from the long-established TSPLIB suite, benchmark standardization for TSP-D is still evolving. The earliest test beds were proposed by Murray and Chu [113], who introduced small FSTSP instances derived from TSPLIB node coordinates. Agatz et al. [114] subsequently contributed structured Euclidean instances for systematic evaluation of optimization approaches. A major step toward standardization was achieved by Dell'Amico et al. [130], who released an open benchmark set with proven optimal solutions for several problem settings; this collection has since become the de facto reference for both exact and heuristic comparisons. For the 2021 Amazon Last-Mile Routing Research Challenge, Merchán et al. [131] published 9184 asymmetric routing instances derived from real delivery van routes across the United States, including GPS coordinates, service times, and driver-chosen sequences [132]; although originally designed for the VRP, these instances can be adapted for TSP-D studies by selecting customer subsets and introducing drone sortie constraints. Despite these contributions, no unified TSP-D benchmark library analogous to TSPLIB exists yet, and many studies still rely on ad hoc instance generators, which complicates cross-study comparison and reproducibility.

3.2.4. Most Effective Algorithmic Approaches

Among the wide range of metaheuristics applied to TSP-D, certain algorithmic families have consistently achieved superior results. Adaptive Large Neighborhood Search (ALNS) has emerged as one of the most effective frameworks, owing to its ability to accommodate the complex destroy-and-repair operators needed for joint truck–drone scheduling; the work by Mara et al. [117] demonstrated strong performance on the FSTSP with multiple drops. Hybrid Genetic Algorithm approaches, which combine genetic operators with problem-specific local search, have also proved competitive: the GA + ACO hybrid by Gunay-Sezer et al. [133] improved several best-known TSP-D solutions. More recently, the Iterative Chainlet Partitioning (ICP) algorithm by Lee et al. [134], which models the truck–drone tour as chainlets and employs dynamic programming within a ruin-and-recreate metaheuristic, improved the state-of-the-art by an average of 2.6% across 1249 public instances while reducing CPU time by roughly 91%. On the exact side, the branch-and-cut algorithm proposed by Boccia et al. [115] closed bounds for several previously unsolved FSTSP instances. Two-phase constructive heuristics (TSP-first, drone-assignment second) [113], while less sophisticated, remain competitive baselines due to their simplicity and speed. Overall, the most effective approaches tend to combine constructive and improvement phases, exploit problem-specific neighborhood structures, and integrate exact subproblem solvers within heuristic frameworks (matheuristics).

3.3. Practical Applications and Deployment

Thanks in part to steady progress on air-worthiness standards (e.g., the U.S. FAA's Part 135 certification for package-delivery drones) and growing public acceptance, a number of truck–drone systems have moved from laboratory studies to live operations. These deployments provide valuable evidence that the theoretical benefits captured by TSP-D models translate into meaningful real-world gains chiefly as faster cycle times, lower cost per stop, and substantial CO_2 savings when drones replace short van hops.

In discussing last-mile parcel delivery, a relevant example is provided by Amazon Prime Air: following the FAA's approval for Beyond Visual Line of Sight (BVLOS) operations in June 2024 [135], Amazon expanded its drone delivery services in College Station, Texas, and announced plans for a new hub in San Antonio, aiming to achieve same-day delivery of packages weighing up to 3 kg [136,137]. These developments are documented in environmental assessments published by the Federal Aviation Administration (FAA), which detail the operational parameters, safety measures, and logistical frameworks supporting drone integration into last-mile delivery networks [136,137]. Although Amazon's current model primarily focuses on direct drone deliveries without a truck moving alongside, the logistics behind drone launching, scheduling, and synchronization are conceptually very similar to TSP-D models, where coordination between trucks and drones is optimized to minimize overall delivery time.

Alphabet's Wing (Walmart & Wing) began drone delivery operations for Walmart in the Dallas-Fort Worth (DFW) metroplex in August 2023, initially serving approximately 60,000 households. In early 2025, Wing had completed over 75,000 deliveries in the Dallas area, maintaining an average flight time from takeoff to delivery of 3 min and 27 s. Walmart announced plans to expand drone delivery services to up to 75% of the DFW population, covering approximately 1.8 million households [138–140].

Another real-world application where the TSP-D accurately models operational logistics is the medical supply chain. Since 2019, UPS's drone subsidiary, UPS Flight Forward, in partnership with Matternet, has operated daily drone deliveries of medical specimens at WakeMed Hospital in Raleigh, North Carolina [141]. Building on this success, similar drone-enabled logistics loops have been launched in Florida and California, notably at UC

San Diego Health, to expedite the transport of lab specimens, supplies, and documents between campuses [142]. In these systems, fixed hubs, such as hospitals, serve as launching and receiving points for drones, effectively combining ground-based staging with aerial sorties, a main characteristic of the TSP-D framework. Typical transport cycles cover 5–9 km in less than 10 min, achieving delivery speeds approximately three times faster than courier vans operating in urban traffic.

4. Metaheuristic Approaches for TSP

A diverse set of metaheuristic approaches are investigated in this section, crafted to tackle the NP-hard nature of the Traveling Salesman Problem and its extensions. Metaheuristic methods provide effective and scalable alternatives when exact algorithms become computationally prohibitive [143].

4.1. Metaheuristic Algorithms

Metaheuristics drive virtually all state-of-the-art results on medium-to-large TSP instances because they offer a pragmatic balance between solution quality and computation time. In broad strokes, they combine exploration, i.e., searching new regions of the solution space, with exploitation, i.e., intensifying the search around high-quality tours, through carefully designed move operators, adaptive parameters, and memory structures. The following subsections review the "classical five" single-method families and then survey the growing body of hybrid schemes that blend their strengths.

Metaheuristics for permutation problems like TSP typically share four components: (1) solution encoding (usually a direct permutation or path representation); (2) generator of neighboring tours (e.g., 2-opt, 3-opt, or Or-opt); (3) a selection rule that decides whether a neighbor is accepted; and (4) optional long-term feedback such as adaptive cooling, pheromone trails, or Tabu memories [144]. Runtime is dominated by evaluating edge exchanges, so incremental cost updates and candidate-list pruning are crucial implementation tricks [145,146]. Recent surveys [147] emphasize that parallelism [148], adaptive parameter control [149] and problem-specific neighborhoods [11] remain the most effective avenues for improvement across essentially all metaheuristics. In the following paragraphs, we will explore how classic metaheuristics work concerning the TSP problem.

4.1.1. Genetic Algorithms (GAs)

GAs encode a tour as a chromosome and evolve it through selection, crossover and mutation. Modern variants use specialized operators such as Edge-Assembly Crossover (EAX) [123], which preserves high-quality subtours, and self-adaptive restart strategies. Recent multi-operator GAs have achieved new best solutions in several TSPLIB instances by mixing four crossovers and embedding local 2-opt search after every generation [150,151]. For larger problems, hybrid GA frameworks remain competitive in min–max mTSP [46] and in time-dependent or release-date extensions [152,153].

4.1.2. Tabu Search (TS)

Tabu Search (TS) iteratively explores the best non-Tabu neighbor while forbidding or penalizing recently visited moves to prevent cycling. Early TSP-specific milestones include Glover's ejection chains [154] and the Johnson–McGeoch case study [155]. More recent TS algorithms incorporate geometric sparsification to prune long edges based on angular metrics [156] and employ adaptive Tabu tenures for dynamic environments.

4.1.3. Ant Colony Optimization (ACO)

ACO uses a colony of artificial ants to build tours probabilistically according to pheromone trails [157] reinforced on short edges. The original Ant System and its refinement, Ant Colony System (ACS), produced near-optimal solutions on standard benchmarks with only a few dozen ants [42]. Improvements focus on max–min pheromone limits [158], candidate lists, and 2-opt-based local search [159]. A recent study by Tang et al. [49] introduces an adaptive Ant-Colony system that adjusts the state-transition and pheromone-update rules and applies selective 2-opt, reporting the Percentage Error (PE) $\leq$ 1% and Err $\leq$ 2%, across 45 TSPLIB instances. For large Euclidean TSPs, CPU-oriented ACO variants and GPU-aware implementations deliver strong performance at scale. On the CPU side, Skinderowicz [160] introduced Focused ACO (FACO), a refinement of the Max–Min Ant System (MMAS) [158], which achieves subpercent gaps to the best-known tours on very large instances while keeping runtimes practical on commodity multicore hardware. On the GPU side, a warp-specialized Ant Colony System accelerates solution construction (with a static–dynamic balanced candidate-set strategy) and pheromone updates, yielding substantial speed-ups over prior GPU ACS baselines (with competitive tour quality) [161]. Complementing this, Dawson and Stewart [162] parallelize the candidate-set (candidate-list) mechanism on GPUs and report up to an order-of-magnitude speed-up while preserving solution quality. Overall, these results show that both the design of architecture-aware algorithms (CPU/GPU) and parallelization of key ACO data structures (such as candidate lists) can scale ACS-style methods to very large TSP instances without sacrificing accuracy. For the drone-assisted variant, Gunay-Sezer et al. [133] presented the first GA + ACO hybrid that improves several best-known TSP-D solutions, and Dell'Amico et al. [130] provided the standard open benchmark set with proven optimal tours.

4.1.4. Particle Swarm Optimization (PSO)

Although classical PSO operates in a continuous space, several discrete encodings such as swap-sequence, edge exchange, or random-key representations have been successfully adapted to the TSP. Wang et al. (2023) [163] introduced a discrete PSO variant tailored for the TSP, demonstrating that PSO can be effectively reformulated for combinatorial contexts by redefining position and velocity operators accordingly. Additionally, Wu (2020) [164] conducted an empirical comparison of PSO, GA, and ACO on TSP instances, showing that PSO's performance aligns closely with that of GA on medium-sized Euclidean problems.

4.2. Benchmark Instances

A fair comparison between metaheuristics is based on two crucial aspects: (1) selection of representative test instances; and (2) choice of robust quantitative metrics to evaluate performance. This section reviews the benchmark libraries that have become standard in TSP research.

Reinelt's TSPLIB is de facto the standard suite for symmetric and asymmetric TSP and related problems such as the Hamiltonian circuit [165] problem and vehicle-routing problem (CVRP) (The official mirror is maintained at http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/, accessed on 9 november 2025).

TSPLIB groups instances based on structural properties (EU = Euclidean; CEIL = rounded Euclidean; GEO = great-circle; etc.) and size (14–85,900 nodes). Exact optimal tours are known for every instance, which makes TSPLIB indispensable as a "sanity check" for new algorithms. A recent community effort added "hard" Euclidean instances that force Concorde to run orders of magnitude longer than on classic TSPLIB cases; Hougardy's public release [166] provides ready-to-use files up to 200 nodes and a generator for larger sizes.

4.2.1. n-TSP Synthetic Sets

Learning-based solvers rarely train on TSPLIB because the number of distinct instances is too small. Instead, they rely on large corpora of random Euclidean instances generated by sampling n points uniformly in the unit square. The community convention is to label these sets TSPn, where n is the corresponding number of nodes (e.g., TSP20, TSP50, TSP100, etc.). Kool et al. released one million training cases and made the generator open source, which has become the default in ICLR/NeuroIPS articles on neural combinatorial optimization [167]. Because optimal tours are unknown for $n > 100$, researchers report the gap to Concorde's solution or the best-known tour after a time limit. Beyond routing-specific benchmark libraries, synthetic weighted benchmarks have also been proposed in the metaheuristics literature to study the behavior of stochastic search processes under controlled conditions [168].

4.2.2. Real-World Data

Researchers are increasingly testing on geospatial or application-specific instances that capture road-network distances; UAV energy models; PCB drills; or genome ordering. Examples include:

- 24,978-city Sweden tour (https://www.math.uwaterloo.ca/tsp/sweden/index.html, accessed on 1 March 2026); the 15,112-city Germany tour (https://www.math.uwaterloo.ca/tsp/d15sol/index.html, accessed on 1 March 2026); and other country-wide instances;
- PCB/VLSI drilling data: classical TSPLIB files such as `pcb442`, `pcb1173`, and the 102-instance VLSI suite model tool-head moves when drilling or etching circuit boards and wafers; they span 131 to 744,710 nodes and are still widely used in metaheuristic studies [75,169];
- Last-mile logistics zones: the 2021 Amazon Last-Mile Routing Research Challenge opened 9184 asymmetric TSP/VRP instances derived from real van routes, complete with GPS coordinates, service times, and driver-chosen sequences [131,132]. Similar GIS-extracted data sets now appear in studies on drone-enabled delivery and urban micro-fulfilment.

Unlike TSPLIB, these data often lack certified optima; therefore, comparative studies must rely on relative indicators (see Section 4.3).

Synthetic Euclidean sets are invaluable for controlled experiments: they allow systematic scaling in n and admit fast distance lookups, but they gloss over domain peculiarities such as asymmetry, forbidden edges, or clustered customer patterns. A balanced benchmark campaign therefore should be as follows: (*i*) validate the algorithm on TSPLIB (guaranteed optima); (*ii*) stress-tests on large synthetic n-TSP data (scalability); and (*iii*) demonstrate applicability on at least one domain-specific real instance set (external validity). Table 2 reports and summarizes the widely used benchmark data sets for the TSP.

Table 2. Widely used benchmark data sets for the Traveling Salesman Problem (TSP). SYM = symmetric distances, ASY = asymmetric distances.

Dataset	Nodes n	Type	Ref.	Algorithms Commonly Tested
TSPLIB 95	14–85,900	SYM/ASY	[165]	Concorde, LKH, GA, ACO, SA
Hard Euclidean TSPLIB	$\leq$200 (gen.)	SYM	[166]	Concorde, LKH
Random Eucl. n-TSP	20–100 test (1 M) train	SYM	[167]	Attention Model, POMO, RL-TSP
VLSI/PCB suite (102)	131–744,710	SYM	[169]	LKH, ACO, large-scale heuristics
World TSP	1,904,711	GEO distance	[170]	LKH (record tour), D&C
GalaxyTSP	1.69×10^9	SYM	[171]	ML-guided D&C + LKH
DIMACS TSP Challenge	up to 10,000,000	SYM	[172]	Concorde, CLK, LKH

4.3. Performance Metrics

In evaluating the performance of metaheuristic algorithms for TSP and its variants, several quantitative metrics have emerged as standard measures, such as:

- *Best Known Solution* (BKS): the shortest tour length $L_{(BKS)}$ currently found in literature, or the one returned by Concorde given a generous time limit;
- *Optimality Gap*: let L be the length of a candidate tour. Then, the optimality gap is given by:
$$\text{Gap } (\%) = 100\left(\frac{L - L_{\text{BKS}}}{L_{\text{BKS}}}\right). \tag{5}$$
Gaps should be averaged over an entire class of instances (e.g., all EUC-2D problems with $n \leq 1000$) and accompanied by the standard deviation;
- *Time-To-Target plots* (TTT-plots): the wall-clock time required to hit a fixed gap threshold (often 0% or 0.1%). Reporting TTT curves conveys the anytime behavior of metaheuristics more faithfully than a single "final" gap;
- *Robustness Across Seeds*: stochastic algorithms must quote the number of independent runs r; mean μ_L; and standard deviation σ_L (or coefficient of variation) of tour lengths. A low σ_L value indicates stable performance; a high σ_L value suggests reliance on lucky random choices;
- *Success rate*: proportion of runs that reach the target gap within the cut-off time. This is particularly informative for hard TSPLIB or adversarial instances;
- *Memory and energy*: for ML-based solvers, peak GPU memory and average power draw are increasingly reported, supporting green-AI comparisons.

5. Integration with Machine Learning and Other Techniques

Recent work has shown that data-driven components can complement or in some cases replace hand-crafted metaheuristics for the TSP and its variants. Deep neural networks now build high-quality tours in milliseconds; reinforcement learning modules steer neighborhood search; and quantum or language model tools provide additional acceleration or human-readable insights. This section reviews the main directions, grouping them into two issues: (1) machine learning enhancements in classical optimization; and (2) emerging hardware and generative AI techniques.

5.1. End-to-End Neural Solvers

Pointer networks marked the first end-to-end model for tour construction, and recent works have extended this paradigm using graph attention mechanisms and reinforcement learning for dynamic tour refinement [173]. A more recent structure-aware Transformer augments standard attention with closeness-centrality and spatial encodings; when trained with policy-gradient RL, it achieves mean deviations of 0.03%, 0.16%, and 1.13% from Concorde in 20, 50, and 100 node Euclidean instances, respectively, with inference times around 17 ms for TSP20 and under 85 ms for TSP100 [174]. Moving beyond supervised or RL-based training, the UTSP framework (Unsupervised Learning for TSP) learns a graph neural network with a surrogate cycle-consistency loss and filters its edge heat-map with local 2-opt; it outperforms previous learned heuristics on the same testbed while using only 10% of their parameters and 0.2% of the training data [175].

5.2. Learning-Enhanced Hyper-Heuristics

Bouazza's [176] 2024 survey processes a corpus of 963 hyper-heuristic (HH) papers with text-mining to map the field and proposes a compact three-axis taxonomy: whether the high-level heuristic (HLH) selects or generates low-level heuristics; whether the search is *constructive* or *perturbative*; and whether learning feedback is absent, offline, online or mixed. It shows that the selection HHs now vastly outnumber generation HHs and that reinforcement-learning variants, especially Q-learning, Deep Q-Networks and large-state RL with GNN embeddings, dominate modern implementations across scheduling, routing, packing and warehouse domains. Rule-generation studies are fewer but highlight interpretable or data-efficient models such as decision tree, random forest and deep symbolic regression policies for dispatch rules and knapsack scoring. The review also tracks emergent themes: deeper ML integration; multi-objective evolutionary HHs; and translation of methods into real-world logistics and industrial testbeds. Finally, it identifies unresolved barriers, namely scalability on large instance sets, cross-domain generalization, and the need for explainable policies, which it argues will drive next-generation research toward distributed inference, transfer/meta-learning and transparent RL controllers. A very recent contribution by Lee et al. [134] introduces the Iterative Chainlet Partitioning (ICP) algorithm for the Traveling Salesman Problem with Drone and a graph-neural "neuro" acceleration (NICP) that fits naturally within a ruin-and-recreate metaheuristic. ICP models the truck–drone tour as a sequence of launch–rendezvous "rings", which are grouped into overlapping *chainlets*. At each iteration, the chainlet with the highest predicted gain is selected and re-optimized using a dynamic programming subroutine (TSP-ep-all), while the rest are cached for reuse. A theoretical bound ensures that beyond the initial iteration, at most 25 subroutine calls are required, yielding in a total work that scales linearly in the number of customers. Across 1249 public TSP-D instances, ICP improved the state-of-the-art by an average of 2.6% while reducing CPU time by 91.3%. Replacing expensive subroutine calls with a graph-transformer predictor leads to Neuro-ICP, which cuts runtime by a further 28.6% at the cost of just 0.14% in solution quality because each move is still certified by an exact partitioner.

5.3. Dynamic and Stochastic Settings

Deep RL is especially attractive when travel times or customer sets change online. Chen et al. [177] propose an actor–critic framework based on a Dynamic Graph Temporal Attention model (DGTA-RL) for the Dynamic Traveling Salesman Problem with Time-Dependent and Stochastic travel times (DTSP-TDS). Their architecture encodes evolving travel time distributions via a multi-head dual-attention mechanism and a temporal pointer decoder, achieving near real-time decisions under uncertainty. Experiments show that DGTA-RL consistently outperforms rolling baselines and prior RL methods in both quality and runtime, with strong generalization to unseen instances. Zhang et al. [178] address the DTSP and its pickup-and-delivery variant using a dual encoder–decoder architecture that separately models traffic dynamics and customer updates. One encoder handles time-dependent travel costs, while the other re-encodes the environment after each customer visit. Their policy-gradient agent, tested on real-world Beijing traffic, yields high-quality routes with millisecond-level inference and outperforms both rolling baselines and dynamic programming. Konovalenko and Hvattum [179] studied a dynamic vehicle routing problem (DVRP) where customer orders arrive stochastically and must be accepted or rejected in real time. They frame the DVRP as a Markov Decision Process and train a Proximal Policy Optimization (PPO) agent. Through controlled experiments, they show that enriching the state space with spatial and derived features, such as distance-to-order and load-demand ratios, improves rewards significantly, while careless feature expansion (e.g., redundant one-hot encodings) hinders learning. While dynamic and stochastic settings emphasize real-time adaptation through learning-based policies, a complementary line of research explores how exact optimization methods can be hybridized with algorithmic or learning components to enhance performance on structured, static variants of drone-enabled routing problems. In the following, we present some relevant works.

5.4. Hybrid Exact Learning Approaches

Montemanni and Dell'Amico [180] introduced the first constraint programming (CP) model for the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP), a bi-agent routing problem where a truck and multiple drones operate in parallel from a central depot. Implemented using Google OR-Tools CP-SAT and parallelized over eight threads, the solver closes all 720 benchmark instances with 10–20 customers within 180 s and establishes the first exact solutions for several larger TSPLIB-derived cases. For three instances, it improves the best-known upper bounds by up to 15%. Compared to MILP-based solvers, the CP model exhibits strong parallel scalability and proves optimality where previous methods returned only heuristic results. The authors highlight constraint programming as a promising and underutilized tool for drone–truck coordination.

Boccia et al. [115] propose a new arc-based MILP formulation for the Flying Sidekick Traveling Salesman Problem (FS-TSP), which models a hybrid delivery system that involves one truck and one drone. Their formulation avoids big-M constants and time-indexed variables, instead relying on a compact three-index model with polynomial growth in variables. To enhance solver performance, they designed a Branch-and-Cut algorithm incorporating several classes of valid inequalities (targeting infeasible sorties and coordination gaps) as well as variable fixing strategies based on drone endurance. Experiments on standard benchmarks show that the method improves or closes bounds for several previously unsolved instances, outperforming prior formulations in both quality and runtime.

Cheng et al. [181] address the passenger-and-parcel Share-A-Ride Problem with Drones (SARP-D), where demand-responsive buses equipped with autonomous drones coordinate the transport of both passengers and packages. The problem is formulated as a mixed-integer nonlinear program (MINP) that models synchronized drone launches during passenger boarding and alighting. To scale to real-world sizes (up to 200 requests), the authors developed an adaptive large neighborhood search (ALNS) metaheuristic that dynamically selects destroy and repair operators, guided by a time-slack strategy and simulated annealing acceptance. Outcomes show that the algorithm produces high-quality solutions efficiently, supporting urban multi-modal logistics with soft time windows and energy constraints.

Bruni et al. [182] developed a logic-based Benders decomposition (LBD) algorithm for the Multi-Trip Traveling Repairman Problem with Drones (MTRPD), where a single truck and multiple drones collaborate to minimize customer waiting times. The problem is tackled as a mixed-integer linear program over an extended graph, allowing each drone to perform multiple trips launched from arbitrary truck stops. The LBD approach decouples truck routing (master problem) from drone scheduling (subproblem) and iteratively adds cuts based on trip feasibility. The algorithm solves instances with up to 25 customers and outperforms standard CPLEX models by up to an order of magnitude, demonstrating how decomposition-based exact methods can manage the combinatorial complexity of synchronized drone–truck operations. As algorithmic advances continue to push the frontier of drone-enabled optimization, emerging research also explores how novel hardware platforms and generative AI tools can unlock further gains in scalability, realism, and interactive solution design.

5.5. Emerging Hardware and Generative Tools

Ciacco et al. [183] explore the potential of quantum annealing (QA) to solve the Steiner Traveling Salesman Problem (STSP), a variant of TSP where non-mandatory "Steiner" nodes may be traversed to reduce overall travel cost. The authors formulate the problem as an integer linear program (ILP) and then convert it into a Quadratic Unconstrained Binary Optimization (QUBO) model suitable for quantum devices. To address the scalability bottleneck of current QA hardware, they developed a preprocessing method (PMRA) that significantly reduces the problem size by pruning redundant arcs and disconnected nodes. The reduced QUBO (RQUBO) formulation is tested on both D-Wave's QPU and LeapBQM hybrid solver. Results show that PMRA enables the quantum models to outperform their unreduced counterparts in both solution quality and runtime, with LeapBQM achieving a 55% lower cost on average compared to the standard formulation. This study represents the first known application of QA to the STSP and underscores the promise of hybrid classical-quantum solvers for complex graph-based routing problems.

Ramezani et al. [184] propose novel quantum encoding for the Traveling Salesman Problem (TSP) that reduces the required qubit count from the conventional n^2 to $nlog_2n$, enabling more efficient deployment on current Noisy Intermediate-Scale Quantum (NISQ) devices. Built within the Quantum Approximate Optimization Algorithm (QAOA) framework, their method replaces position-based binary variables with logarithmic city encodings and formulates a corresponding Hamiltonian that preserves solution feasibility. Simulation experiments on 4-city TSP instances show that this compact encoding outperforms the standard one across multiple metrics, including approximation ratio, rank, and true solution probability, while reducing quantum resource requirements. Although gate complexity increases in theory, the lower qubit demand makes the method promising for real-world quantum processors, paving the way for more scalable quantum optimization in the NISQ era.

Wang et al. [185] propose DEITSP, a diffusion-based non-autoregressive (NAR) solver for the Traveling Salesman Problem that balances high-quality solutions with fast inference. Unlike standard denoising diffusion models that simulate lengthy Markov chains, DEITSP performs direct one-step denoising through a self-consistency-enhanced training regime. A novel dual-modality graph Transformer jointly encodes node and edge features to improve structural reasoning, while an efficient iterative strategy alternates between adding and removing noise to enhance solution diversity. The model supports configurable iteration steps to trade off quality and latency. Across 20–1000 node TSP instances and real-world maps, DEITSP outperforms prior NAR solvers and even competes with autoregressive methods such as Transformer and POMO, achieving near-optimal results at a fraction of their runtime. This work demonstrates how diffusion models can scale to combinatorial settings when tailored to task-specific structure and inference goals.

Basson and Preux [186] introduce IDEQ, an improved diffusion-based neural solver for the Traveling Salesman Problem that sets new performance records among generative models. Building on DIFUSCO and T2TCO, IDEQ introduces two key innovations: (1) it enforces Hamiltonian tour structure during inference via a reconstruction operator and local 2-opt refinement; and (2) it replaces the strict one-tour supervision target with a distribution over 2-opt equivalence classes to smooth the learning signal. These refinements improve both convergence and generalization without requiring full retraining. Experiments on TSPLIB and synthetic benchmarks show that IDEQ consistently outperforms previous diffusion solvers and achieves optimality gaps of just 0.3% on TSP-500 and 0.5% on TSP-1000, surpassing even LKH3 on select large instances. The model also exhibits reduced variance in solution quality and better scalability with problem size, highlighting the robustness of structure-aware generative design in neural combinatorial optimization.

Masoud et al. [187] explore the use of Large Language Models (LLMs), specifically GPT-3.5 Turbo, to solve the Traveling Salesman Problem (TSP) via prompting and fine-tuning. They evaluate zero-shot, few-shot, and chain-of-thought (CoT) prompting strategies, showing that well-crafted prompts can produce valid and reasonably accurate tours for instances with up to 20 nodes. The authors further fine-tune GPT-3.5 on fixed-size TSP instances and find that the model generalizes effectively to larger problem sizes. To enhance output quality without retraining, they introduce a self-ensemble approach that queries the model multiple times and selects the best result, improving robustness and performance. The study demonstrates that general-purpose LLMs can exhibit latent combinatorial reasoning abilities when paired with careful prompt design and lightweight adaptation.

5.6. Convergence Guarantees of ML-Hybrid Metaheuristics

Despite their strong empirical performance, most ML-hybrid metaheuristics for routing problems lack formal convergence or optimality guarantees. End-to-end neural solvers trained with REINFORCE or Proximal Policy Optimization converge to stationary points of the policy-gradient loss, yet a stationary point may correspond to a poor solution distribution for the underlying NP-hard combinatorial problem. Caramanis et al. [188] provide the first positive theoretical result: for a broad class of combinatorial problems including TSP, they show that polynomial-size generative models exist whose policy-gradient landscape has no sub-optimal stationary points, and they introduce a regularization procedure that provably escapes bad stationary regions. However, this analysis applies to carefully designed surrogate models rather than to the Transformer or GNN architectures deployed in practice. On the empirical side, Liu et al. [189] systematically benchmark neural combinatorial optimization solvers on TSP instances of increasing size and show that training-loss convergence does not imply consistent solution quality: performance degrades markedly when the evaluation size exceeds the training distribution, and no tested neural solver consistently matches traditional solvers such as LKH-3 across all scales. Manchanda et al. [190] corroborate this finding, demonstrating that attention-based constructive solvers generalize poorly to instances even moderately larger than those seen during training. For hybrid schemes that embed reinforcement learning within a metaheuristic loop (e.g., deep reinforcement learning for operator selection in ALNS [191] or graph-neural accelerators inside ruin-and-recreate frameworks [134]), convergence proofs reduce to classical RL results (tabular Q-learning converges under infinite exploration, but Deep Q-Networks provide no finite-time guarantee). In summary, certified optimality or bounded sub-optimality for ML-augmented routing solvers remains an open theoretical challenge, and bridging this gap is essential before such methods can be trusted in safety-critical or high-value logistics applications.

6. Research Gaps and Conclusions

Research on metaheuristics has advanced from their development for the early tour-improvement tricks for the classical Traveling Salesman Problem to sophisticated hybrids that tackle large-scale, drone-enabled logistics in real time. However, a fully mature field-ready toolkit remains just beyond reach because five obstacles still loom: (*i*) even the most versatile neighborhoods and self-tuning parameters can falter on adversarial instances or heavily constrained variants, a sign that quantitative guidance on search-space exploration and automated configurators is still under-used; (*ii*) neural and other learning-driven solvers create high-quality routes in seconds, but their value is blunted by brittle behavior outside the training regime, opaque decision paths, and a rarely quantified energy bill; Concretely, attention-based constructive models and RL-trained improvement operators frequently see optimality gaps double or triple when evaluated on instances only moderately larger than those in the training set [189,190]; no mainstream neural solver yet provides worst-case performance certificates analogous to the constant-factor approximation ratios available for classical heuristics. Equally concerning is the lack of interpretability: practitioners cannot audit why a learned policy selects a particular drone launch or truck detour, which limits adoption in regulated logistics environments. Future work should therefore pursue (a) formal out-of-distribution robustness bounds, (b) post hoc explainability layers (e.g., attention-weight attribution or counterfactual analysis), and (c) standardized energy-cost accounting that reports GPU-hours alongside solution quality; (*iii*) drone-assisted formulations idealize operations by treating battery endurance as constant, ignoring sequential sorties, and omitting regulated air corridors, leaving a gap between models and practice; In particular, most TSP-D and FSTSP models assume a fixed flight range independent of payload weight and ambient conditions; even recent battery-aware studies such as Es Yurek [192] model energy consumption as proportional to distance, explicitly noting that real-world factors including drone speed, wind, and payload affect discharge in ways their model does not capture. Furthermore, the predominant single-sortie paradigm (one launch, one delivery, one retrieval) excludes sequential multi-delivery sorties in which a drone visits several nearby customers before returning to the truck; allowing such sorties could significantly increase the utilization of airborne assets, yet the combinatorial complexity of multi-stop drone routing under dynamic energy constraints is largely unexplored. Equally neglected are regulatory airspace restrictions: no-fly zones around airports, hospitals, and critical infrastructure, as well as altitude ceilings imposed by national aviation authorities, constrain the feasible drone graph but are rarely embedded in optimization models. Closing this gap requires energy models that capture weight-dependent, time-varying discharge rates, multi-delivery sortie scheduling within a single launch–retrieval cycle, and geofence-aware path planning that respects real-world airspace regulations; (*iv*) reproducibility is uneven: full code, seed lists, anytime curves, and shareable real-world instances are not always available, which hinders fair comparison and cumulative progress; (*v*) finally, exact and heuristic approaches still form separate camps; warm-starting exact solvers with heuristic incumbents and, conversely, steering heuristics with dual bounds offer a promising but mostly unexplored bridge. The "matheuristic" paradigm, i.e., embedding MILP or constraint-programming sub-solvers inside a metaheuristic shell, has proved effective in classical vehicle routing [127,128] but remains rare in TSP-D research. For example, an ALNS framework could delegate drone-assignment or time-window sub-problems to a MILP solver at each destroy–repair iteration, exploiting the tight bounds that exact methods provide on small neighborhoods while the metaheuristic manages the global search. Conversely, heuristic incumbents can warm-start branch-and-cut or Benders decomposition, drastically pruning the search tree, as demonstrated by Boccia et al. [115] and Bruni et al. [182] for related drone–truck coordination problems. Systematically investigating such hybrid exact-heuristic strategies for TSP-D variants, including the use of column generation with metaheuristic pricing and logic-based decompositions with learning-enhanced master problems, is a high-priority direction for the field.

Addressing these challenges will demand algorithms that combine adaptive search with interpretable learning modules, models that reflect dynamic batteries and regulated airspace, and a community culture of open data and transparent reporting. Success on those fronts promises routing solutions that are faster, greener, and more robust, ensuring that the next generation of metaheuristics keeps pace with the ever-evolving logistics landscape. In particular, establishing formal convergence and approximation guarantees for ML-hybrid metaheuristics, designing energy-aware sortie models with realistic battery dynamics, bridging the exact-heuristic divide through matheuristic decompositions, and enforcing transparent, reproducible experimental protocols are the most pressing next steps toward field-ready drone–truck optimization.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the project PIACERI-PIAno di inCEntivi per la Ricerca di Ateneo 2024/2026—Linea di Intervento i "Progetti di ricerca collaborativa".

Data Availability Statement: Data reported in the manuscript was obtained from the well-known literature database, such as Scopus, WoS, etc, as well as from the related journal/conference proceedings website.

Acknowledgments: The authors wish to thank to the anonymous reviewers for their important and valuable comments.

Conflicts of Interest: Author Alessio Mezzina was employed by the company Koexai S.R.L. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Courier Corporation: North Chelmsford, MA, USA, 1998.
2. Hoffman, K.L.; Padberg, M.; Rinaldi, G. Traveling salesman problem. *Encycl. Oper. Res. Manag. Sci.* **2013**, *1*, 1573–1578.
3. Pahwa, A.; Jaller, M. A cost-based comparative analysis of different last-mile strategies for e-commerce delivery. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102783. [CrossRef]
4. Hernandez, F.; Sotelo, R.; Forets, M. Optimization algorithms for adaptative route sequencing on real-world last-mile deliveries. *Ingenius* **2024**, *31*, 64–80. [CrossRef]
5. Ghindaoanu, S.V.; Ancau, M. The Current State of Scientific Research in the Field of Optimizing Printed Circuit Board Processing Technologies by Minimizing Drilling Path Using Genetic Algorithms. *Acad. J. Manuf. Eng.* **2024**, *22, 126–133*.
6. Luo, J.; Zhang, Z.; Ma, X.; Yan, C.; Luo, H. GTasm: A genome assembly method using graph transformers and HiFi reads. *Front. Genet.* **2024**, *15*, 1495657. [CrossRef] [PubMed]
7. Athigiri, S.; Krishna S, A.; Ahmed, T. Optimizing GEO Belt Observation Through Analytical Methods and the Traveling Salesman Problem. In Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, Maui, HI, USA, 17–20 September 2024; p. 70.
8. Tasneem, O.; Pieters, R. Automatic Robot Path Planning for Active Visual Inspection on Free-Form Surfaces. In *Proceedings of the 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, 28 August–1 September 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 173–180.
9. Vásquez, S.A.; Angulo, G.; Klapp, M.A. An exact solution method for the TSP with Drone based on decomposition. *Comput. Oper. Res.* **2021**, *127*, 105127. [CrossRef]
10. Biasi, M.D. Minimal TSP Tour is coNP-Complete. *arXiv* **2014**, arXiv:1403.3431. [CrossRef]
11. Toaza, B.; Esztergár-Kiss, D. A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. *Appl. Soft Comput.* **2023**, *148*, 110908. [CrossRef]
12. Cutello, V.; Mezzina, A.; Pavone, M.; Zito, F. A Real-Time Adaptive Tabu Search for Handling Zoom In/Out in Map Labeling Problem. In *Learning and Intelligent Optimization*; Springer Nature: Cham, Switzerland, 2025; pp. 108–122. [CrossRef]
13. Zito, F.; Talbi, E.G.; Cavallaro, C.; Cutello, V.; Pavone, M. Metaheuristics in automated machine learning: Strategies for optimization. *Intell. Syst. Appl.* **2025**, *26*, 200532. [CrossRef]
14. Zito, F.; Cutello, V.; Pavone, M. Data-driven forecasting and its role in enhanced decision-making. *Eng. Appl. Artif. Intell.* **2025**, *154*, 110934. [CrossRef]
15. Cavallaro, C.; Cutello, V.; Pavone, M.; Zito, F. Discovering anomalies in big data: A review focused on the application of metaheuristics and machine learning techniques. *Front. Big Data* **2023**, *6*, 1179625. [CrossRef]
16. Fan, M.; Li, J. Surrogate-assisted genetic algorithms for the travelling salesman problem and vehicle routing problem. In *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
17. Zheng, J.; Zhong, J.; Chen, M.; He, K. A reinforced hybrid genetic algorithm for the traveling salesman problem. *Comput. Oper. Res.* **2023**, *157*, 106249. [CrossRef]
18. Zito, F.; Cutello, V.; Pavone, M. A Novel Reverse Engineering Approach for Gene Regulatory Networks. In *Complex Networks and Their Applications XI*; Springer International Publishing: Berlin/Heidelberg, Germany, 2023; pp. 310–321. [CrossRef]
19. Hutter, F.; Xu, L.; Hoos, H.H.; Leyton-Brown, K. Algorithm runtime prediction: Methods & evaluation. *Artif. Intell.* **2014**, *206*, 79–111. [CrossRef]
20. Zito, F.; Cutello, V.; Pavone, M. A General-Purpose Neural Architecture Search Algorithm for Building Deep Neural Networks. In *Metaheuristics*; Springer Nature: Cham, Switzerland, 2024; pp. 126–141. [CrossRef]

21. Cutello, V.; Pavone, M.; Zito, F. Inferring a Gene Regulatory Network from Gene Expression Data. An Overview of Best Methods and a Reverse Engineering Approach. In *From Computational Logic to Computational Biology*; Springer Nature: Cham, Switzerland, 2024; pp. 172–185. [CrossRef]
22. Cook, W.J. The World TSP: A Traveling Salesman Problem Tour of the Earth. 2003. Available online: http://www.math.uwaterloo.ca/tsp/world/index.html (accessed on 14 April 2025).
23. Applegate, D.L.; Bixby, R.E.; Chvátal, V.; Cook, W.J. Star-TSP Tours: Traveling Salesman Problem in Space. 2002. Available online: http://www.math.uwaterloo.ca/tsp/star/index.html (accessed on 14 April 2025).
24. Bosch, R.; Herman, A. Continuous line drawings via the traveling salesman problem. *Oper. Res. Lett.* **2004**, *32*, 302–303. [CrossRef]
25. Bosch, R. Opt art. *Math Horiz.* **2006**, *13*, 6–9. [CrossRef]
26. Bosch, R.; Herman, A. Pointillism via linear programming. *UMAP J.* **2005**, *26*, 405–412.
27. Kaplan, C.S.; Bosch, R. TSP art. In Proceedings of the Renaissance Banff: Mathematics, Music, Art, Culture, Banff, AB, Canada, 31 July–3 August 2005; pp. 301–308.
28. Bosch, R.; Kaplan, C. TSP Art Instances. Available online: https://www.math.uwaterloo.ca/tsp/data/art/index.html (accessed on 14 April 2025).
29. Applegate, D.L.; Bixby, R.E.; Chvátal, V.; Cook, W.J. Concorde TSP Solver. 2001. Available online: http://www.math.uwaterloo.ca/tsp/concorde.html (accessed on 14 April 2025).
30. Boyd, S.; Mattingley, J. *Branch and Bound Methods*; Notes EE364b; Stanford University: Stanford, CA, USA, 2007.
31. Mitchell, J.E. Branch-and-cut algorithms for combinatorial optimization problems. *Handb. Appl. Optim.* **2002**, *1*, 65–77.
32. Kelley, J.E., Jr. The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 703–712. [CrossRef]
33. Cook, W.J.; Applegate, D.L.; Bixby, R.E.; Chvátal, V. *The Traveling Salesman Problem: A Computational Study*; Princeton University Press: Princeton, NJ, USA, 2011.
34. Bellman, R. Dynamic programming treatment of the travelling salesman problem. *J. ACM (JACM)* **1962**, *9*, 61–63.
35. Held, M.; Karp, R.M. A dynamic programming approach to sequencing problems. *J. Soc. Ind. Appl. Math.* **1962**, *10*, 196–210. [CrossRef]
36. Biggs, N. The traveling salesman problem a guided tour of combinatorial optimization. *Bull. Lond. Math. Soc.* **1986**, *18*, 514–515. [CrossRef]
37. Gendreau, M.; Potvin, J.Y. *Handbook of Metaheuristics*; Springer: New York, NY, USA, 2010; Volume 2.
38. Osman, I.H.; Kelly, J.P. Meta-heuristics theory and applications. *J. Oper. Res. Soc.* **1997**, *48*, 657. [CrossRef]
39. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; The MIT Press: Cambridge, MA, USA, 1992. [CrossRef]
40. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
41. Glover, F.; Laguna, M. *Tabu Search*; Springer: Berlin/Heidelberg, Germany, 1998.
42. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
43. Lau, H.C.; Sim, M.; Teo, K.M. Vehicle routing problem with time windows and a limited number of vehicles. *Eur. J. Oper. Res.* **2003**, *148*, 559–569. [CrossRef]
44. Bektas, T. The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega* **2006**, *34*, 209–219. [CrossRef]
45. Fischetti, M.; Gonzalez, J.J.S.; Toth, P. Solving the orienteering problem through branch-and-cut. *INFORMS J. Comput.* **1998**, *10*, 133–148. [CrossRef]
46. Mahmoudinazlou, S.; Kwon, C. A Hybrid Genetic Algorithm for the min-max Multiple Traveling Salesman Problem. *arXiv* **2023**, arXiv:2307.07120. [CrossRef]
47. Lupoaie, V.I.; Chili, I.A.; Breaban, M.E.; Raschip, M. SOM-Guided Evolutionary Search for Solving MinMax Multiple-TSP. *arXiv* **2019**, arXiv:1907.11910. [CrossRef]
48. de Castro Pereira, S.; Solteiro Pires, E.J.; de Moura Oliveira, P.B. Ant-Balanced Multiple Traveling Salesmen: ACO-BmTSP. *Algorithms* **2023**, *16*, 37. [CrossRef]
49. Tang, K.; Wei, X.F.; Jiang, Y.H.; Chen, Z.W.; Yang, L. An Adaptive Ant Colony Optimization for Solving Large-Scale Traveling Salesman Problem. *Mathematics* **2023**, *11*, 4439. [CrossRef]
50. Singh, D.R.; Singh, M.K.; Singh, T.; Prasad, R. Genetic algorithm for solving multiple traveling salesmen problem using a new crossover and population generation. *Comput. Sist.* **2018**, *22*, 491–503. [CrossRef]
51. Croes, G.A. A method for solving traveling-salesman problems. *Oper. Res.* **1958**, *6*, 791–812. [CrossRef]
52. Alharbi, M.G.; Stohy, A.; Elhenawy, M.; Masoud, M.; El-Wahed Khalifa, H.A. Solving Traveling Salesman Problem with Time Windows Using Hybrid Pointer Networks with Time Features. *Sustainability* **2021**, *13*, 12906. [CrossRef]
53. Bartolini, E.; Goeke, D.; Schneider, M.; Ye, M. The robust traveling salesman problem with time windows under knapsack-constrained travel time uncertainty. *Transp. Sci.* **2021**, *55*, 371–394. [CrossRef]

54. Vu, D.M.; Hewitt, M.; Vu, D.D. Solving Time-Dependent Traveling Salesman Problem with Time Windows under Generic Time-Dependent Travel Cost. *arXiv* **2023**, arXiv:2311.08111. [CrossRef]
55. Ban, H.B.; Pham, D.H. Solving optimization problems simultaneously: The variants of the traveling salesman problem with time windows using multifactorial evolutionary algorithm. *PeerJ Comput. Sci.* **2023**, *9*, e1192. [CrossRef] [PubMed]
56. Chen, J.; Gong, Z.; Liu, M.; Wang, J.; Yu, Y.; Zhang, W. Looking Ahead to Avoid Being Late: Solving Hard-Constrained Traveling Salesman Problem. *arXiv* **2024**, arXiv:2403.05318. [CrossRef]
57. Holliday, J.B.; Blount, D.; Osaba, E.; Luu, K. Advanced Quantum Annealing Approach to Vehicle Routing Problems with Time Windows. *arXiv* **2025**, arXiv:2503.24285. [CrossRef]
58. Balas, E. The prize collecting traveling salesman problem. *Networks* **1989**, *19*, 621–636. [CrossRef]
59. Ntaimo, L.; Arrubla, J.A.G.; Stripling, C.; Young, J.; Spencer, T. A stochastic programming standard response model for wildfire initial attack planning. *Can. J. For. Res.* **2012**, *42*, 987–1001. [CrossRef]
60. Ausiello, G.; Bonifaci, V.; Laura, L. The online prize-collecting traveling salesman problem. *Inf. Process. Lett.* **2008**, *107*, 199–204. [CrossRef]
61. Ruiz, J.; Gonzalez, C.; Chen, Y.; Tang, B. Prize-collecting traveling salesman problem: A reinforcement learning approach. In *Proceedings of the ICC 2023-IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 4416–4421.
62. Paul, A.; Freund, D.; Ferber, A.; Shmoys, D.B.; Williamson, D.P. Prize-Collecting TSP with a Budget Constraint. In Proceedings of the Embedded Systems and Applications, Hsinchu, Taiwan, 16–18 August 2017.
63. Paul, A.; Freund, D.; Ferber, A.; Shmoys, D.B.; Williamson, D.P. Budgeted prize-collecting traveling salesman and minimum spanning tree problems. *Math. Oper. Res.* **2020**, *45*, 576–590. [CrossRef]
64. Khodamoradi, K.; Krishnamurti, R. Prize collecting travelling salesman problem-fast heuristic separations. In *Proceedings of the International Conference on Operations Research and Enterprise Systems*; SCITEPRESS: Setúbal, Portugal, 2016; Volume 2, pp. 380–387.
65. Faigl, J.; Hollinger, G.A. Self-organizing map for the prize-collecting traveling salesman problem. In *Proceedings of the Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 10th International Workshop, WSOM 2014, Mittweida, Germany, 2–4 July 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 281–291.
66. Chaves, A.; Lorena, L. Hybrid Metaheuristic for the Prize Collecting Travelling Salesman Problem. In *Evolutionary Computation in Combinatorial Optimization*; van Hemert, J., Cotta, C., Eds.; EvoCOP 2008. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4972, pp. 123–134. [CrossRef]
67. Pedro, O.; Saldanha, R.; Camargo, R. A Tabu Search Approach for the Prize Collecting Traveling Salesman Problem. *Electron. Notes Discret. Math.* **2013**, *41*, 261–268. [CrossRef]
68. Paul, A.; Freund, D.; Ferber, A.; Shmoys, D.B.; Williamson, D.P. Prize-collecting TSP with a budget constraint. In *Proceedings of the 25th Annual European Symposium on Algorithms (ESA 2017), Vienna, Austria, 4–6 September 2017*; Schloss Dagstuhl–Leibniz-Zentrum für Informatik: Dagstuhl, Germany, 2017; pp. 62:1–62:14.
69. Faigl, J.; Hollinger, G.A. Autonomous data collection using a self-organizing map. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 1703–1715. [CrossRef]
70. Tang, L.; Wang, X. An iterated local search heuristic for the capacitated prize-collecting travelling salesman problem. *J. Oper. Res. Soc.* **2008**, *59*, 590–599. [CrossRef]
71. Luo, Y.; Liu, M.; Hao, Z.; Liu, D. An improved NSGA-II algorithm for multi-objective traveling salesman problem. *TELKOMNIKA Indones. J. Electr. Eng.* **2014**, *12*, 4413–4418. [CrossRef]
72. Coit, D.W.; Smith, A.E.; Tate, D.M. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS J. Comput.* **1996**, *8*, 173–182. [CrossRef]
73. Krasnogor, N.; Smith, J. A memetic algorithm with self-adaptive local search: TSP as a case study. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, Vegas, NV, USA, 10–12 July 2000*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000; pp. 987–994.
74. Feillet, D.; Dejax, P.; Gendreau, M. Traveling salesman problems with profits. *Transp. Sci.* **2005**, *39*, 188–205. [CrossRef]
75. Reinelt, G. TSPLIB—A traveling salesman problem library. *ORSA J. Comput.* **1991**, *3*, 376–384. [CrossRef]
76. Noon, C.E. *The Generalized Traveling Salesman Problem*; University of Michigan: Ann Arbor, MI, USA, 1988.
77. Noon, C.E.; Bean, J.C. An efficient transformation of the generalized traveling salesman problem. *INFOR Inf. Syst. Oper. Res.* **1993**, *31*, 39–44. [CrossRef]
78. Senturk, I.F.; Kebe, G.Y. A novel shortest path routing algorithm for wireless data collection in transportation networks. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
79. Laporte, G.; Asef-Vaziri, A.; Sriskandarajah, C. Some applications of the generalized travelling salesman problem. *J. Oper. Res. Soc.* **1996**, *47*, 1461–1467. [PubMed]

80. Zhao, X.; Lin, J.L.; Huang, H.; Hao, Z.F. Void Vertex Genetic Algorithm for Production-Distribution Supply Chain GTSP Model. *J. Inf. Comput. Sci.* **2006**, *1*, 259–265.
81. Bähnemann, R.; Lawrance, N.; Chung, J.J.; Pantic, M.; Siegwart, R.; Nieto, J. Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem. In *Proceedings of the Field and Service Robotics: Results of the 12th International Conference, Tokyo, Japan, 29–31 August 2019*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 277–290.
82. Nekovář, F.; Faigl, J.; Saska, M. Multi-tour set traveling salesman problem in planning power transmission line inspection. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6196–6203. [CrossRef]
83. Pop, P.C.; Cosma, O.; Sabo, C.; Sitar, C.P. A comprehensive survey on the generalized traveling salesman problem. *Eur. J. Oper. Res.* **2024**, *314*, 819–835. [CrossRef]
84. Samshad, M.; Rajawat, K. Communication and Energy-Aware Multi-UAV Coverage Path Planning for Networked Operations. *arXiv* **2024**, arXiv:2411.02772.
85. Min, G.; Liu, L.; Zhai, W.; Wang, Z.; Lu, W. An efficient data collection algorithm for partitioned wireless sensor networks. *Future Gener. Comput. Syst.* **2023**, *140*, 53–66. [CrossRef]
86. Prins, C.; Lacomme, P.; Prodhon, C. Order-first split-second methods for vehicle routing problems: A review. *Transp. Res. Part C Emerg. Technol.* **2014**, *40*, 179–200.
87. Ekiz, M.K.; Bozdemir, M.; Türkkan, B.Ö. Route first-cluster second method for personal service routing problem. *J. Eng. Stud. Res.* **2019**, *25*, 18–24. [CrossRef]
88. Adasme, P.; Andrade, R.; Leung, J.; Lisser, A. A two-stage stochastic programming approach for the traveling salesman problem. In *Proceedings of the International Conference on Operations Research and Enterprise Systems, Rome, Italy, 23–25 February 2016*; SCITEPRESS: Setúbal, Portugal, 2016; Volume 2, pp. 163–169.
89. Wissink, P.L. The Traveling Salesman Problem with Stochastic and Correlated Customers. *Transp. Sci.* **2023**, *57*, 1321–1339. [CrossRef]
90. Xiao, P.; Zhang, Z.; Chen, J.; Wang, J.; Zhang, Z. Neural Combinatorial Optimization for Robust Routing Problem with Uncertain Travel Times. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 134841–134867.
91. Lienkamp, B.; Hewitt, M.; Schiffer, M. Branch and price for the stochastic traveling salesman problem with generalized latency. *Transp. Sci.* **2025**, *59*, 229–249. [CrossRef]
92. Florio, A.M.; Gendreau, M.; Hartl, R.F.; Minner, S.; Vidal, T. Recent advances in vehicle routing with stochastic demands: Bayesian learning for correlated demands and elementary branch-price-and-cut. *Eur. J. Oper. Res.* **2023**, *306*, 1081–1093. [CrossRef]
93. Zhang, W.; Wang, K.; Jacquillat, A.; Wang, S. Optimized scenario reduction: Solving large-scale stochastic programs with quality guarantees. *INFORMS J. Comput.* **2023**, *35*, 886–908. [CrossRef]
94. Verweij, B.; Ahmed, S.; Kleywegt, A.J.; Nemhauser, G.; Shapiro, A. The sample average approximation method applied to stochastic routing problems: A computational study. *Comput. Optim. Appl.* **2003**, *24*, 289–333. [CrossRef]
95. Luo, Z.; Qin, H.; Zhang, D.; Lim, A. Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *85*, 69–89. [CrossRef]
96. Joe, W.; Lau, H.C. Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In Proceedings of the International Conference on Automated Planning and Scheduling, Nancy, France, 14–19 June 2020; Volume 30, pp. 394–402.
97. Wang, X.; Zhao, J. Distributionally robust optimization of the vehicle routing problem with uncertain customers. *J. Ind. Manag. Optim.* **2025**, *21*, 1983–2006. [CrossRef]
98. Kanesen, Y.; Zakaria, H.L.; Asi, M.; Othman, R.R.; Ariffin, W.N.S.F.W.; Ramli, N.; Abd Ghani, M.A.H. Bibliometric Analysis Of Travelling Salesman Problem With Drones (TSP-D). *Int. J. Adv. Commun. Technol. (IJACT)* **2024**, *4*, 29–46. [CrossRef]
99. Kong, J.; Xie, M.; Wang, H. Integrating Autonomous Vehicles and Drones for Last-Mile Delivery: A Routing Problem with Two Types of Drones and Multiple Visits. *Drones* **2025**, *9*, 280. [CrossRef]
100. Morandi, N.; Leus, R.; Matuschke, J.; Yaman, H. The TSP with drones: The benefits of retraversing the arcs. *Transp. Sci.* **2023**, *57*, 1340–1358. [CrossRef]
101. Montemanni, R.; Dell'Amico, M.; Corsini, A. Parallel drone scheduling vehicle routing problems with collective drones. *Comput. Oper. Res.* **2024**, *163*, 106514. [CrossRef]
102. Dell'Amico, M.; Montemanni, R.; Novellani, S. Modeling the flying sidekick traveling salesman problem with multiple drones. *Networks* **2021**, *78*, 303–327. [CrossRef]
103. Ren, X.; Froger, A.; Jabali, O.; Liang, G. A competitive heuristic algorithm for vehicle routing problems with drones. *Eur. J. Oper. Res.* **2024**, *318*, 469–485. [CrossRef]
104. Schermer, D.; Moeini, M.; Wendt, O. The Drone-Assisted Traveling Salesman Problem with Robot Stations. In Proceedings of the HICSS, Maui, HI, USA, 7–10 January 2020; pp. 1–10.
105. Berhan, E.; Beshah, B.; Kitaw, D.; Abraham, A. Stochastic vehicle routing problem: A literature survey. *J. Inf. Knowl. Manag.* **2014**, *13*, 1450022. [CrossRef]

106. He, Y.; Zheng, Z.; Li, H.; Deng, J. A Stochastic Drone-Scheduling Problem with Uncertain Energy Consumption. *Drones* **2024**, *8*, 430. [CrossRef]
107. Dinh, T.; Fukasawa, R.; Luedtke, J. Exact algorithms for the chance-constrained vehicle routing problem. *Math. Program.* **2018**, *172*, 105–138. [CrossRef]
108. Ren, X.X.; Fan, H.M.; Bao, M.X.; Fan, H. The time-dependent electric vehicle routing problem with drone and synchronized mobile battery swapping. *Adv. Eng. Inform.* **2023**, *57*, 102071. [CrossRef]
109. Yurek, E.E.; Ozmutlu, H.C. Traveling salesman problem with drone under recharging policy. *Comput. Commun.* **2021**, *179*, 35–49. [CrossRef]
110. Bhuiyan, T.H.; Walker, V.; Roni, M.; Ahmed, I. Aerial drone fleet deployment optimization with endogenous battery replacements for direct delivery of time-sensitive products. *Expert Syst. Appl.* **2024**, *252*, 124172. [CrossRef]
111. Otto, A.; Agatz, N.; Campbell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. [CrossRef]
112. Macrina, G.; Pugliese, L.D.P.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102762. [CrossRef]
113. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109.
114. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [CrossRef]
115. Boccia, M.; Mancuso, A.; Masone, A.; Sterle, C. A new MILP formulation for the flying sidekick traveling salesman problem. *Networks* **2023**, *82*, 254–276. [CrossRef]
116. Roberti, R.; Ruthmair, M. Exact methods for the traveling salesman problem with drone. *Transp. Sci.* **2021**, *55*, 315–335. [CrossRef]
117. Mara, S.T.W.; Rifai, A.P.; Sopha, B.M. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Syst. Appl.* **2022**, *205*, 117647. [CrossRef]
118. Pilcher, T. A self-adaptive genetic algorithm for the flying sidekick travelling salesman problem. *arXiv* **2023**, arXiv:2310.14713. [CrossRef]
119. Özoğlu, B.; Çakmak, E.; Koç, T. Clarke & Wright's savings algorithm and genetic algorithms based hybrid approach for flying sidekick traveling salesman problem. *Eur. J. Sci. Technol.* **2019**, 185–192. [CrossRef]
120. Masmoudi, M.A.; Mancini, S.; Baldacci, R.; Kuo, Y.H. Vehicle routing problems with drones equipped with multi-package payload compartments. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102757.
121. Yu, V.F.; Lin, S.W.; Jodiawan, P.; Lai, Y.C. Solving the flying sidekick traveling salesman problem by a simulated annealing heuristic. *Mathematics* **2023**, *11*, 4305. [CrossRef]
122. Tong, B.; Wang, J.; Wang, X.; Zhou, F.; Mao, X.; Zheng, W. Optimal route planning for truck–drone delivery using variable neighborhood tabu search algorithm. *Appl. Sci.* **2022**, *12*, 529. [CrossRef]
123. Nagata, Y.; Kobayashi, S. A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *INFORMS J. Comput.* **2013**, *25*, 346–363. [CrossRef]
124. Krasnogor, N.; Smith, J. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Trans. Evol. Comput.* **2005**, *9*, 474–488. [CrossRef]
125. Resende, M.G.; Ribeiro, C.C. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2010; pp. 283–319.
126. Duhamel, C.; Lacomme, P.; Prins, C.; Prodhon, C. A GRASP× ELS approach for the capacitated location-routing problem. *Comput. Oper. Res.* **2010**, *37*, 1912–1923. [CrossRef]
127. Archetti, C.; Speranza, M.G. A survey on matheuristics for routing problems. *EURO J. Comput. Optim.* **2014**, *2*, 223–246. [CrossRef]
128. Boschetti, M.A.; Maniezzo, V. Matheuristics: Using mathematics for heuristic design. *4OR* **2022**, *20*, 173–208. [CrossRef]
129. Han, Y.q.; Li, J.q.; Liu, Z.; Liu, C.; Tian, J. Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420920031. [CrossRef]
130. Dell'Amico, M.; Montemanni, R.; Novellani, S. Benchmark Instances and Optimal Solutions for the Traveling Salesman Problem with Drone. *arXiv* **2021**, arXiv:2107.13275. [CrossRef]
131. Merchán, D.; Arora, J.; Pachon, J.; Konduri, K.; Winkenbach, M.; Parks, S.; Noszek, J. 2021 Amazon last mile routing research challenge: Data set. *Transp. Sci.* **2024**, *58*, 8–11. [CrossRef]
132. Cook, W.J. Amazon Challenge Data. University of Waterloo TSP Pages. 2023. Available online: https://www.math.uwaterloo.ca/tsp/amz/data.html (accessed on 5 May 2025)
133. Gunay-Sezer, N.S.; Cakmak, E.; Bulkan, S. A Hybrid Metaheuristic Solution Method to Traveling Salesman Problem with Drone. *Systems* **2023**, *11*, 259. [CrossRef]

134. Lee, J.H.; Kim, M.; Park, J.; Kwon, C. The Iterative Chainlet Partitioning Algorithm for the Traveling Salesman Problem with Drone and Neural Acceleration. *arXiv* **2025**, arXiv:2504.15147. [CrossRef]
135. Amazon Staff. Prime Air Expands Drone Deliveries After FAA Approval. 2024. Available online: https://www.aboutamazon.com/news/transportation/amazon-drone-prime-air-expanded-delivery-faa-approval (accessed on 28 April 2025).
136. Federal Aviation Administration. Final Supplemental Environmental Assessment-College Station, Texas. 2024. Available online: https://www.faa.gov/sites/faa.gov/files/FINAL-College-Station-Supplemental-EA-23Sept2024.pdf (accessed on 27 April 2025).
137. Federal Aviation Administration. Environmental Assessment-Amazon Prime Air Operations in Lockeford, California. 2022. Available online: https://www.faa.gov/sites/faa.gov/files/2022-09/EA-Amazon-Prime-Air_Lockeford-CA_9-8-22_508.pdf (accessed on 27 April 2025).
138. Wing Team. Wing Celebrates 3 Years of Drone Delivery in DFW. 2025. Available online: https://wing.com/news/wing-celebrates-3-years-of-drone-delivery-in-dfw (accessed on 28 April 2025).
139. Walmart Inc. Sky High Ambitions: Walmart To Make Largest Drone Delivery Expansion of Any U.S. Retailer. 2024. Available online: https://corporate.walmart.com/news/2024/01/09/sky-high-ambitions-walmart-to-make-largest-drone-delivery-expansion-of-any-us-retailer (accessed on 28 April 2025).
140. sUAS News. Wing and Walmart Continue to Expand Service to Millions of Customers in Dallas-Fort Worth Amid New Dallas-Wide FAA Approvals. 2024. Available online: https://www.suasnews.com/2024/01/wing-and-walmart-continue-to-expand-service-to-millions-of-customers-in-dallas-fort-worth-amid-new-dallas-wide-faa-approvals/ (accessed on 28 April 2025).
141. Business Insider. UPS and Matternet Just Made the First Revenue-Generating Drone Delivery in the US. 2019. Available online: https://www.businessinsider.com/ups-first-revenue-generating-drone-delivery-with-matternet-2019-3 (accessed on 28 April 2025).
142. Unmanned Airspace. San Diego to Test Medical Deliveries in Pilot Project with UPS and Matternet. 2020. Available online: https://www.unmannedairspace.info/latest-news-and-information/san-diego-to-test-medical-deliveries-in-pilot-project-with-ups-and-matternet/ (accessed on 28 April 2025).
143. Greco, S.; Pavone, M.; Talbi, E.G.; Vigo, D. *Metaheuristics for Combinatorial Optimization*; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2021; Volume 1332. [CrossRef]
144. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv. (CSUR)* **2003**, *35*, 268–308. [CrossRef]
145. Johnson, D.S.; Gutin, G.; McGeoch, L.A.; Yeo, A.; Zhang, W.; Zverovitch, A. Experimental analysis of heuristics for the ATSP. In *The Traveling Salesman Problem and Its Variations;* Springer: Boston, MA, USA, 2007; pp. 445–487.
146. Bentley, J.J. Fast algorithms for geometric traveling salesman problems. *ORSA J. Comput.* **1992**, *4*, 387–411. [CrossRef]
147. Alorf, A. A survey of recently developed metaheuristics and their comparative analysis. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105622.
148. Lazarova, M.; Borovska, P. Comparison of parallel metaheuristics for solving the TSP. In *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, Gabrovo, Bulgaria, 12–13 June 2008*; Association for Computing Machinery: New York, NY, USA, 2008; pp. II.12-1–II.12-6.
149. Aleti, A.; Moser, I. A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Comput. Surv. (CSUR)* **2016**, *49*, 1–35. [CrossRef]
150. Varadarajan, S.; Whitley, D. A parallel ensemble genetic algorithm for the traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference, Lille, France, 10–14 July 2021*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 636–643.
151. Gutiérrez-Aguirre, P.; Contreras-Bolton, C. A multioperator genetic algorithm for the traveling salesman problem with job-times. *Expert Syst. Appl.* **2024**, *240*, 122472. [CrossRef]
152. Ban, H.B. The hybridization of aco+ ga and rvns algorithm for solving the time-dependent traveling salesman problem. *Evol. Intell.* **2022**, *15*, 309–328. [CrossRef]
153. Soares, G.; Bulhões, T.; Bruck, B. An efficient hybrid genetic algorithm for the traveling salesman problem with release dates. *Eur. J. Oper. Res.* **2024**, *318*, 31–42. [CrossRef]
154. Glover, F. Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discret. Appl. Math.* **1996**, *65*, 223–253. [CrossRef]
155. Johnson, D.S.; McGeoch, L.A. The traveling salesman problem: A case study in local optimization. *Local Search Comb. Optim.* **1997**, *1*, 215–310.
156. Cavagnini, R.; Schneider, M.; Theiß, A. A tabu search with geometry-based sparsification methods for angular traveling salesman problems. *Networks* **2024**, *83*, 30–52. [CrossRef]
157. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1996**, *26*, 29–41. [CrossRef]
158. Stützle, T.; Hoos, H.H. MAX–MIN ant system. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [CrossRef]

159. Gambardella, L.M.; Dorigo, M. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS J. Comput.* **2000**, *12*, 237–255. [CrossRef]
160. Skinderowicz, R. Improving Ant Colony Optimization efficiency for solving large TSP instances. *Appl. Soft Comput.* **2022**, *120*, 108653. [CrossRef]
161. Huang, Z.B.; Fu, G.T.; Fa, T.H.; Dong, D.Y.; Bai, P.; Xiao, C. High performance ant colony system based on GPU warp specialization with a static–dynamic balanced candidate set strategy. *Future Gener. Comput. Syst.* **2021**, *125*, 136–150. [CrossRef]
162. Dawson, L.; Stewart, I.A. Candidate Set Parallelization Strategies for Ant Colony Optimization on the GPU. In *Proceedings of the Algorithms and Architectures for Parallel Processing*; Kołodziej, J., Di Martino, B., Talia, D., Xiong, K., Eds.; Springer: Cham, Switzerland, 2013; pp. 216–225.
163. Wang, K.P.; Huang, L.; Zhou, C.G.; Pang, W. Particle swarm optimization for traveling salesman problem. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE cat. no. 03ex693), Xi'an, China, 5 November 2003*; IEEE: Piscataway, NJ, USA, 2003; Volume 3, pp. 1583–1585.
164. Wu, Z. A comparative study of solving traveling salesman problem with genetic algorithm, ant colony algorithm, and particle swarm optimization. In *Proceedings of the 2020 2nd International Conference on Robotics Systems and Vehicle Technology, Xiamen, China, 3–5 December 2020*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 95–99.
165. Reinelt, G. *TSPLIB95*; Interdisciplinary Center for Scientific Computing (IWR): Heidelberg, Germany, 1995; Volume 338, pp. 1–16.
166. Hougardy, S.; Zhong, X. Hard to solve instances of the Euclidean Traveling Salesman Problem. *Math. Program. Comput.* **2021**, *13*, 51–74. [CrossRef]
167. Kool, W.; Van Hoof, H.; Welling, M. Attention, learn to solve routing problems! *arXiv* **2018**, arXiv:1803.08475.
168. Cutello, V.; Mezzina, A.; Pavone, M.; Zito, F. A Weighted Binary String Benchmark to Assess the Efficiency of Stochastic Search Processes. In *Advances in Swarm Intelligence*; Lecture Notes in Computer Science; Springer: Singapore, 2025; Volume 16011, pp. 116–127. [CrossRef]
169. Cook, W.J. TSP Test Data. University of Waterloo. Available online: https://www.math.uwaterloo.ca/tsp/data/index.html (accessed on 5 May 2025).
170. Cook, W.J. World Traveling Salesman Problem. 2021. Available online: https://www.math.uwaterloo.ca/tsp/world/ (accessed on 17 June 2025).
171. Drori, I.; Kates, B.J.; Sickinger, W.R.; Kharkar, A.G.; Dietrich, B.; Shporer, A.; Udell, M. GalaxyTSP: A New Billion-Node Benchmark for TSP. In Proceedings of the Learning Meets Combinatorial Algorithms at NeurIPS2020, Virtual, 11–12 December 2020.
172. 8th DIMACS Implementation Challenge: The Traveling Salesman Problem. 2000. Available online: http://archive.dimacs.rutgers.edu/Challenges/TSP/ (accessed on 1 March 2026).
173. Luo, J.; Heng, H.; Wu, G. Graph attention, learning 2-opt algorithm for the traveling salesman problem. *Complex Intell. Syst.* **2025**, *11*, 117. [CrossRef]
174. Zhao, C.S.; Wong, L.P. A transformer-based structure-aware model for tackling the traveling salesman problem. *PLoS ONE* **2025**, *20*, e0319711. [CrossRef] [PubMed]
175. Min, Y.; Bai, Y.; Gomes, C.P. Unsupervised learning for solving the travelling salesman problem. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 47264–47278.
176. Bouazza, W. Machine Learning-Based Hyper-Heuristics: A Clear Insight. In *Proceedings of the 2024 7th International Conference on Computational Intelligence and Intelligent Systems (CIIS '24), Nagoya, Japan, 22–24 November 2024*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 29–37. [CrossRef]
177. Chen, D.; Imdahl, C.; Lai, D.; Van Woensel, T. The Dynamic Traveling Salesman Problem with Time-Dependent and Stochastic travel times: A deep reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2025**, *172*, 105022. [CrossRef]
178. Zhang, Z.; Liu, H.; Zhou, M.; Wang, J. Solving dynamic traveling salesman problems with deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 2119–2132. [CrossRef] [PubMed]
179. Konovalenko, A.; Hvattum, L.M. Optimizing a Dynamic Vehicle Routing Problem with Deep Reinforcement Learning: Analyzing State-Space Components. *Logistics* **2024**, *8*, 96. [CrossRef]
180. Montemanni, R.; Dell'Amico, M. Solving the parallel drone scheduling traveling salesman problem via constraint programming. *Algorithms* **2023**, *16*, 40. [CrossRef]
181. Cheng, R.; Jiang, Y.; Nielsen, O.A.; Pisinger, D. An adaptive large neighborhood search metaheuristic for a passenger and parcel share-a-ride problem with drones. *Transp. Res. Part C Emerg. Technol.* **2023**, *153*, 104203. [CrossRef]
182. Bruni, M.E.; Khodaparasti, S.; Moshref-Javadi, M. A logic-based Benders decomposition method for the multi-trip traveling repairman problem with drones. *Comput. Oper. Res.* **2022**, *145*, 105845. [CrossRef]
183. Ciacco, A.; Guerriero, F.; Osaba, E. Steiner Traveling Salesman Problem with Quantum Annealing. *arXiv* **2025**, arXiv:2504.02388. [CrossRef]

184. Ramezani, M.; Salami, S.; Shokhmkar, M.; Moradi, M.; Bahrampour, A. Reducing the Number of Qubits from n^2 to $nlog_2(n)$ to Solve the Traveling Salesman Problem with Quantum Computers: A Proposal for Demonstrating Quantum Supremacy in the NISQ Era. *arXiv* **2024**, arXiv:2402.18530.
185. Wang, M.; Zhou, Y.; Cao, Z.; Xiao, Y.; Wu, X.; Pang, W.; Jiang, Y.; Yang, H.; Zhao, P.; Li, Y. An Efficient Diffusion-based Non-Autoregressive Solver for Traveling Salesman Problem. *arXiv* **2025**, arXiv:2501.13767.
186. Basson, M.; Preux, P. IDEQ: An improved diffusion model for the TSP. *arXiv* **2024**, arXiv:2412.13858. [CrossRef]
187. Masoud, M.; Abdelhay, A.; Elhenawy, M. Exploring combinatorial problem solving with large language models: A case study on the travelling salesman problem using gpt-3.5 turbo. *arXiv* **2024**, arXiv:2405.01997. [CrossRef]
188. Caramanis, C.; Fotakis, D.; Kalavasis, A.; Kontonis, V.; Tzamos, C. Optimizing solution-samplers for combinatorial problems: The landscape of policy-gradient method. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 14035–14069.
189. Liu, S.; Zhang, Y.; Tang, K.; Yao, X. How Good is Neural Combinatorial Optimization? A Systematic Evaluation on the Traveling Salesman Problem. *IEEE Comput. Intell. Mag.* **2023**, *18*, 14–28. [CrossRef]
190. Manchanda, S.; Michel, S.; Drakulic, D.; Andreoli, J.M. On the Generalization of Neural Combinatorial Optimization Heuristics. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases, Turin, Italy, 18–22 September 2023*; Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G., Eds.; Springer: Cham, Switzerland, 2023; pp. 426–442.
191. Reijnen, R.; Zhang, Y.; Lau, H.C.; Bukhsh, Z. Online Control of Adaptive Large Neighborhood Search Using Deep Reinforcement Learning. In *Proceedings of the 34th International Conference on Automated Planning and Scheduling (ICAPS 2024)*; Association for the Advancement of Artificial Intelligence (AAAI Press): Palo Alto, CA, USA, 2024; Volume 34, pp. 475–483. [CrossRef]
192. Es Yurek, E. Impact of Drone Battery Recharging Policy on Overall Carbon Emissions: The Traveling Salesman Problem with Drone. *Drones* **2024**, *8*, 108. [CrossRef]