Koc. J. Sci. Eng., 2026(17): 188-202 https://doi.org/10.34088/kojose.2041394

Kocaeli University

Kocaeli Journal of Science and Engineering

http://dergipark.org.tr/kojose

Scale-Dependent Performance Analysis of Metaheuristic Approaches for the Multi-Drone En-Route Truck-Drone Delivery Problem

Samet DİRİ [1, *]

[1] *Department of Information Systems Engineering, Kocaeli University, Kocaeli, Türkiye,* ***ORCID:*** *0000-0003-2849-0219*

Article Info

Research paper

Received : September 16, 2026
Accepted : October 21, 2026

Keywords

Truck-drone delivery problem,
En-route interception,
Metaheuristic optimization,
Genetic Algorithm,
Tabu Search,
African Vultures Optimization Algorithm

Abstract

Truck-drone collaborative last-mile delivery systems extend the classical Traveling Salesperson Problem through complex range, capacity, and synchronization constraints. While most existing studies restrict drone-truck rendezvous to predetermined customer nodes, this study proposes an extended delivery model permitting en-route interception along road segments with multiple concurrent drones to minimize total makespan. The optimization problem is resolved via an O(n) single-pass, event-driven decoding procedure that incorporates a protective ground-wait strategy for the shared single-platform bottleneck and a segment-locking mechanism to prevent retroactive schedule disruption. The framework benchmarks two native-discrete paradigms—a Genetic Algorithm (GA) and a discrete Tabu Search (TS) baseline—against two continuous-domain, sigmoid-binarized metaheuristics: Binary Particle Swarm Optimization (BPSO) and the Binary African Vultures Optimization Algorithm (BAVOA), evaluated across 15 benchmark instances spanning three problem scales (n≈19, n ≈49, and n ≈99) under strictly equal function-evaluation budgets. Nonparametric statistical analyses reveal no significant performance divergence at the small scale (Friedman test, p =0.392). However, as the problem scale increases, native-discrete methods demonstrate statistically significant superiority over both BPSO (p=0.0051) and BAVOA (p=0.0033), achieving near-zero percentage gaps relative to the best-found solutions (GA: 0.016%, TS: 0.011%), while showing no significant difference from each other (Wilcoxon test, p =0.499). Furthermore, runtime evaluations demonstrate that this performance advantage incurs no computational overhead, with GA and TS exhibiting identical execution latencies. These findings provide empirical evidence that direct binary search operators yield significantly more robust and higher-quality solutions than the stochastic transfer functions used in continuous metaheuristics for discrete synchronization problems.

1. Introduction

The rapid expansion of e-commerce and the escalatings launch from a mobile platform mounted on the truck, perform deliveries to individual customer locations, and subsequently rendezvous with the truck. This synchronized operation substantially reduces total vehicle miles traveled, thereby offering significant potential to lower operating costs and overall delivery makespan.

Synchronizing two distinct vehicle classes characterized by disparate cruising speeds and operational constraints introduces a combinatorial complexity far exceeding that of classical vehicle routing problems. Scenarios in which drones can intercept the truck not only at discrete customer stops but also dynamically along road segments during transit require specialized mathematical formulations and synchronization strategies. This study extends the continuous en-route rendezvous paradigm to a multi-drone fleet and evaluates the efficacy of classical and contemporary metaheuristic algorithms, together with a classical discrete local-search baseline, within an event-driven framework designed to solve this extended problem.

The main contributions of this study are fourfold:

1. It relaxes the conventional node-restricted rendezvous assumption by proposing a synchronized routing formulation in which a multi-drone fleet can intercept the moving truck continuously at any point along its active route segments.

* Corresponding Author: samet.diri@kocaeli.edu.tr

2667-484X © This paper published in Kocaeli Journal of Science and Engineering is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

2. It introduces a single-pass, event-driven forward-decoding procedure that explicitly accommodates shared single-platform bottleneck constraints and incorporates protective ground-wait and segment-locking mechanisms, thereby guaranteeing schedule feasibility without the need for retroactive backtracking or penalty adjustments.
3. The proposed framework is systematically benchmarked across multiple problem scales by integrating a classical Genetic Algorithm (GA), Binary Particle Swarm Optimization (BPSO), and the recently introduced Binary African Vultures Optimization Algorithm (BAVOA), providing a rigorous statistical assessment of how algorithmic performance gaps depend on problem dimensionality.
4. It further benchmarks all three metaheuristics against a classical discrete Tabu Search (TS) baseline that searches the binary decision space directly through bit-flip neighborhoods rather than through a continuous-to-binary mapping, thereby isolating whether the observed performance patterns stem from the metaheuristic paradigm itself or from the representation each algorithm operates.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive review of the relevant literature and highlights the positioning of this study. Section 3 details the mathematical formulation, the event-driven decoding procedure, and the experimental design. Section 4 presents the computational results, along with a comparative discussion and statistical evaluations. Finally, Section 5 concludes the paper.

2. Literature Review

2.1. Development of the Truck-Drone Routing Problem

The truck-drone collaborative routing problem was first introduced by Murray and Chu[1] as the Flying Sidekick Traveling Salesperson Problem (FSTSP). In their foundational framework, a single truck operates in tandem with a single drone that serves at most one customer per sortie before rendezvousing with the truck at a predetermined customer node; the authors solved small-scale problem instances using a mixed-integer linear programming (MILP) formulation. Concurrently, Agatz et al.[2] examined this problem under the title of the Traveling Salesperson Problem with Drone (TSP-D) and introduced a dynamic programming-based split approach, demonstrating that drone dispatch schedules can be resolved efficiently once the truck's visiting sequence is fixed. Bouman et al.[3] subsequently extended this methodology to solve medium-sized instances using exact, scalable dynamic programming algorithms. Together, these seminal studies established the foundational mathematical formulations and benchmark datasets that have guided research in this domain over the subsequent decade.

In a comprehensive review, Macrina et al.[4] categorized drone-assisted vehicle routing problems across three primary structural configurations—single-truck single-drone, single-truck multi-drone, and multi-truck multi-drone systems—highlighting the rapid diversification of the field. Furthermore, Moshref-Javadi and Winkenbach[5] classified drone-based logistics models by application domain, emphasizing that realistic vehicle synchronization and precise energy management strategies remain underexplored frontiers in the existing literature.

2.2. Synchronization Approaches and En-Route Models

Classical FSTSP and TSP-D formulations share the restrictive assumption that a truck-drone rendezvous can occur exclusively at predefined customer or depot nodes. This assumption often impairs operational efficiency by forcing the truck to idle at fixed locations while awaiting drone recovery. Marinelli et al.[6] relaxed this constraint by introducing the en-route delivery concept, wherein a drone can intercept the truck continuously at any point along a road segment while the truck remains in motion, formalizing the synchronization problem analytically. Schermer et al.[7] tackled a similar en-route operational architecture using a hybrid variable-neighborhood tabu search algorithm, demonstrating that such flexible rendezvous mechanisms can be solved effectively via heuristic techniques. In parallel, Salama and Srinivas[8] proposed a formulation allowing non-fixed, flexible launch and recovery locations along the truck's path, showing that relaxing node-based constraints markedly enhances drone utilization.

While these en-route strategies are practically advantageous because they eliminate unnecessary truck idling, they substantially complicate solution methodology design, as temporal and spatial rendezvous coordinates must be computed dynamically along the truck's trajectory. Comparing discrete and continuous synchronization regimes, Moshref-Javadi et al.[9] confirmed that continuous en-route rendezvous models significantly reduce overall completion time, particularly across dense customer distributions.

2.3. Multi-Drone Extensions

Given that the single-drone assumption constitutes a major operational bottleneck in high-density logistics,

subsequent studies expanded the scope to multi-drone fleets. Murray and Raj[10] formulated the multiple Flying Sidekick Traveling Salesperson Problem (mFSTSP), in which the truck accommodates multiple drones capable of executing concurrent, independent sorties. Kitjacharoenchai et al.[11] embedded this structure into a two-echelon vehicle routing framework covering multi-truck and multi-drone operations. Luo et al.[12] introduced a multi-visit paradigm in which a single drone visits multiple customer locations per sortie, a concept subsequently generalized by Luo et al.[13] to a one-to-one pickup-and-delivery scheme. Similarly, Liu et al.[14] developed tailored heuristic methods for two-echelon collaborative routing, while Meng et al.[15] incorporated customer time windows into multi-visit pickup-and-delivery configurations.

A limited number of studies have concurrently addressed multi-drone fleets and continuous en-route rendezvous. Thomas et al.[16] proposed an en-route scheduling model for multi-drone delivery, showing that the simultaneous optimization of dispatch sequences and launch schedules exerts a decisive influence on solution quality. Han et al.[17] investigated multi-drone operations with flexible launch configurations, validating experimentally that such flexibility expands effective drone coverage and operational reach. Although these two studies represent the closest precedents to our work, both focus on specific solution heuristics without offering a systematic, scale-dependent performance evaluation of metaheuristics spanning distinct search representations (i. e., native binary versus mapped continuous-domain operators).

2.4. Solution Methods for the Problem

Owing to the NP-hard nature of truck-drone routing problems, exact solution methods such as mixed-integer linear programming (MILP), dynamic programming, and branch-and-bound algorithms can only guarantee optimality within tractable runtimes only for small instances, typically restricted to $n \leq 20$ customers. For medium- and large-scale instances, the literature relies predominantly on heuristic and metaheuristic approaches, including Adaptive Large Neighborhood Search (ALNS), Tabu Search, and various dynamic split procedures. Schermer et al.[18] benchmarked several heuristic algorithms across drone-assisted routing variants, demonstrating that heuristics offer substantial computational advantages over exact solvers as problem dimensionality grows. Recent comprehensive surveys confirm that this field remains highly active, identifying multi-drone coordination alongside dynamic and stochastic operational constraints as critical, underexplored research directions[19]. Furthermore, Zhou et al.[20] reviewed model taxonomies and solution paradigms, noting that while learning-based approaches are gaining traction, their practical scalability remains largely confined to smaller benchmark instances.

Among population-based metaheuristics, GA[21] and PSO[22] represent established paradigms widely applied to combinatorial routing problems. While GA operates naturally on discrete strings via genetic operators, canonical PSO was originally formulated for continuous search spaces. To address binary combinatorial settings, Kennedy and Eberhart[23] introduced Binary Particle Swarm Optimization (BPSO), which maps continuous velocity components into probability distributions via a sigmoid transfer function, enabling stochastic state-bit flips.

In parallel, adapting contemporary nature-inspired metaheuristics to discrete domains has attracted considerable attention. The African Vultures Optimization Algorithm (AVOA), introduced by Abdollahzadeh et al.[24], simulates the foraging, navigation, and scavenging interactions of vulture social hierarchies, exhibiting superior convergence capabilities across diverse engineering design problems[25]. Because AVOA is fundamentally formulated over continuous spaces, Balakrishnan et al.[26] developed discrete variants employing S-shaped and V-shaped transfer functions, demonstrating competitive performance against classical binary metaheuristics in high-dimensional feature selection tasks.

Beyond population-based metaheuristics, classical discrete local-search methods such as Tabu Search[27] operate directly on the native solution representation without any continuous-to-binary mapping layer. Because the present study's decision variable is itself binary, such a method provides a structurally relevant baseline against which the benefit of population-based search can be isolated; the discrete Tabu Search variant used for this purpose.

2.5. Research Gap

A critical synthesis of the literature reveals that continuous en-route rendezvous formulations[6-9] and multi-drone fleet extensions[10-15] have predominantly been investigated as disjoint modeling streams. Research efforts addressing both dimensions concurrently remain notably scarce, confined essentially to the frameworks proposed by Thomas et al.[16] and Han et al.[17]. However, these existing studies are structured around specific scheduling heuristics and do not provide a systematic, scale-dependent performance evaluation of metaheuristics operating across fundamentally different solution representations (i. e., native discrete operators versus mapped continuous-domain dynamics). Moreover, the adaptation and benchmarking of recent high-performing nature-inspired optimizers, such as AVOA, against classical paradigms within synchronized en-route

multi-UAV routing remains largely uncharted. This study directly bridges this gap by formulating a continuous en-route interception model for multi-drone fleets and conducting a rigorous, scale-dependent comparative analysis of population-based metaheuristics and a classical discrete Tabu Search baseline under an event-driven decoding framework.

3. Materials and Methods

3.1. Problem Definition

In the problem addressed in this study, a delivery fleet comprising a single truck and K homogeneous drones departs from a central depot, services the demands of n customers, and ultimately returns to the depot upon completion of the operation. The baseline visiting sequence of the truck is initialized via a pre-solved Traveling Salesperson Problem (TSP) tour. Customers assigned to drone delivery are dynamically bypassed by the truck, thereby shortening the vehicle's road trajectory. Each customer is served exactly once, either directly by the truck or via a dedicated drone sortie.

A drone sortie consists of three consecutive phases: (i) launch from the truck's current location, (ii) direct flight to the designated customer and delivery of the package, and (iii) return flight to a rendezvous point—either an intermediate customer node or continuously along an active road segment—on the truck's subsequent route for retrieval and docking. The mathematical model incorporates the following key operational constraints:

- The number of drones carried by the truck and operating concurrently in the airspace cannot exceed the fleet capacity K.
- Because the truck is equipped with a single takeoff and landing platform, minimum operational buffer times (T_{launch} and T_{dock}) are strictly enforced between consecutive launches, consecutive dockings, and launch-to-dock transitions to allow for battery replacement, parcel loading, and platform clearance.
- Drones can only transport parcels whose weights do not exceed a maximum payload threshold ($Q_{\max}$). Customers incompatible with drone delivery because of excessive parcel weight or volume, or spatial flight restrictions, must be served directly by the truck.
- The total net airborne flight duration of a single sortie cannot exceed the drone's maximum endurance limit ($E_{\max}$). If a drone arrives at the rendezvous location before the truck, it is assumed to perform safe ground waiting with its rotors stopped, thereby avoiding critical hovering battery consumption.

3.2. Mathematical Model

3.2.1. Decision Variable

The solution representation is an n-dimensional binary decision vector that determines each customer's service mode, defined in Eq. (1):

$$x = (x_1, \dots, x_n) \in \{0,1\}^n, \qquad x_j = \begin{cases} 1, & \text{if customer } j \text{ is served by a drone} \\ 0, & \text{otherwise (direct truck service)} \end{cases} \tag{1}$$

The reference truck visit sequence covering all customers, $R = (r_0, r_1, \dots, r_n, r_{n+1})$, where $r_0 = r_{n+1} = 0$ denotes the central depot, is constructed and fixed in advance using the Nearest Neighbor and 2-opt local search heuristics. The proposed metaheuristic searches exclusively over this binary vector x. The remainder of the model is a deterministic decoding procedure that evaluates any candidate solution x against the operational constraints and synchronization rules, converting it into a single fitness value $f(x)$.

3.2.2. Launch and Delayed Departure

The shared-platform constraint is enforced, as shown in Eq. (2), by deferring the launch time according to platform availability:

$$t_{launch} = \max\left(t_i,\ t_{last}^{launch} + T_{launch}\right), \quad t_i \leftarrow t_{launch} \tag{2}$$

where t_i denotes the ready time of the truck at launch point i, and t_{last}^{launch} represents the timestamp of the most recent launch operation. The drone's arrival time at the customer j is computed as $t_{arr} = t_{launch} + d(i,j)/v_d$, where $d(i,j)$ is the Euclidean distance between the launch point and the customer location, and v_d is the drone's cruising speed. The time at which the drone completes service and is ready to depart is $t_{dep} = t_{arr} + s_j$, where s_j is the service time at customer j.

To avoid battery depletion caused by hovering at the rendezvous point, the drone performs protective ground waiting at the customer site until the platform is clear for docking, resulting in the effective takeoff time shown in Eq. (3):

$$t_{takeoff} = \max\left(t_{dep},\ t_{last}^{dock} + T_{dock}\right) \tag{3}$$

3.2.3. Analytical Interception Equation

Assume the truck moves along road segment $[A, B]$ at a constant velocity vector $\vec{V}_t = \frac{B-A}{t_B - t_A}$ with scalar speed $v_t =$

$\|\vec{V}_t\|$. The time t^* at which a drone launched from customer j's location at time $t_{takeoff}$ with cruising speed v_d intercepts the truck is obtained in closed form as the root of a quadratic polynomial. Let $\tau = t - t_A$ denote the relative time normalized to the segment's start and let $\tau_{\min} = t_{takeoff} - t_A$ denote the segment-relative lower bound on the drone's departure time.

By time $\tau \geq \tau_{\min}$, the truck has reached position $A + \vec{V}_t\tau$, while the drone — airborne at constant speed v_d since $\tau_{\min}$ — has covered a straight-line distance of $v_d(\tau - \tau_{\min})$ from the customer location j. Interception requires these two quantities to coincide, i.e., the truck's position must lie exactly on the sphere of radius $v_d(\tau - \tau_{\min})$ centered at j in Eq. (4):

$$\left\|A + \vec{V}_t\tau - j\right\|^2 = v_d^2(\tau - \tau_{\min})^2 \quad (4)$$

The interception condition then takes the form shown in Eq. (5).

Squaring both sides of Eq. (4) and grouping the terms with respect to τ directly yields the quadratic polynomial formulated in Eq. (5):

$$a\tau^2 + b\tau + c = 0 \quad (5)$$

The coefficients of Eq. (5) are derived as shown in Eq. (6).

$$\begin{cases} a = v_t^2 - v_d^2 \\ b = 2\left((A - j) \cdot \vec{V}_t + v_d^2\tau_{min}\right) \\ c = \| A - j \|^2 - v_d^2\tau_{min}^2 \end{cases} \quad (6)$$

Here, j denotes the customer's coordinate vector, A denotes the segment's starting coordinate, $\| \cdot \|$ denotes the Euclidean norm, and $(\cdot)$ denotes the standard vector inner product. The quadratic coefficient a reflects the relative-speed regime ($a = 0$ when $v_t = v_d$, in which case Eq. (5) degenerates to a linear equation solved directly for τ); the linear coefficient b combines the truck's initial positional offset from the customer, projected onto its direction of travel, with the drone's head start $\tau_{\min}$; and the constant term c is the squared initial separation between the truck's segment-start position and the customer, net of the distance the drone can already cover during $\tau_{\min}$. Among the real roots of the polynomial, the smallest τ satisfying the feasibility conditions in Eq. (7) is selected.

$$\tau \geq \tau_{\min} \quad \text{and} \quad 0 \leq \tau \leq t_B - t_A \quad (7)$$

The smallest root satisfying these conditions determines the interception time $t^* = t_A + \tau$. Selecting the earliest valid root ensures that the drone docks at the first moment it meets the truck, preventing it from overtaking the vehicle and either hovering or turning back. If no valid root exists on the segment under examination (due to a negative discriminant or roots falling outside it), the interception search advances to the next consecutive segment along the truck's route.

3.2.4. Segment-Locking Mechanism

Once a drone sortie is validated and accepted, every intermediate customer node situated between the launch location and the beginning of the rendezvous segment $[A, B]$ is locked for mandatory truck visitation, as formalized in Eq. (8).

$$\mathcal{L} \leftarrow \mathcal{L} \cup \{k \in R: k \text{ lies between the launch point and the rendezvous segment}\}, \quad \forall k \in \mathcal{L}: x_k \leftarrow 0 \quad (8)$$

Here, $\mathcal{L}$ denotes the set of customer nodes locked to direct truck service. This dynamic locking mechanism prevents subsequent assignments from altering the forward trajectory and timing profile assumed during the analytical interception calculation. Consequently, it guarantees that the schedule evaluation proceeds as a single-pass, strictly forward decoding procedure that is deterministic, requires no backtracking, and exhibits an $O(n)$ computational complexity.

3.2.5. Objective Function

The optimization objective is to minimize the total operational makespan, defined as the time when all customer demands have been fulfilled and both the truck and all airborne drones have safely returned to the central depot. For a given binary decision vector x, the fitness value $f(x)$ is evaluated as the maximum between the truck's depot arrival time and the completion time of the final drone docking operation, as defined in Eq. (9):

$$f(x) = \max\left(t_{truck}^{depot}(x),\ t_{last}^{dock}\right) \quad (9)$$

where $t_{truck}^{depot}(x)$ represents the truck's arrival timestamp at the depot, and t_{last}^{dock} denotes the instant the final drone completes recovery and docking on the truck or at the depot.

Accordingly, the overall optimization model is formulated in Eq.(10) .

$$0{,}1\, f(x) \quad (10)$$

3.3. Solution Method

3.3.1. Event-Driven Forward Decoding Procedure

A candidate solution vector $x \in \{\{0,1\}\hat{}1\}^n$ is decoded along the truck's reference route in chronological sequence

via a single forward pass. For each customer j encoded as $x_j = 1$, four operational feasibility constraints are evaluated sequentially:

1. **Payload and Eligibility (K1):** The customer's demand does not exceed the maximum payload threshold ($q_j \leq Q_{\max}$), and the customer location is physically and spatially accessible for drone landing.
2. **Fleet and Concurrency Availability (K2):** The number of drones concurrently operating in the airspace at the planned launch time does not exceed the fleet capacity ($K_{\text{active}} < K$).
3. **Interception Feasibility (K3):** A valid rendezvous location satisfying the analytical interception quadratic equation exists along the remaining segments of the truck's forward route.
4. **Flight Endurance (K4):** The sortie's net airborne flight duration does not exceed the drone's maximum endurance limit ($T_{flight} \leq E_{\max}$).

If any of these conditions are violated, the customer is automatically reassigned to direct truck service ($x_j \leftarrow 0$). Through this deterministic repair mechanism, every candidate solution generated in the metaheuristic search space is mapped directly onto a physically feasible schedule without requiring penalty-factor calibrations or constraint-violation penalties.

3.3.2. Genetic Algorithm

The classical binary-coded GA framework incorporates tournament selection, uniform crossover, bit-flip mutation, and an elitism strategy to maintain population diversity while propagating top-performing solutions across generations. Because the decision space is fundamentally defined over $\{0,1\}^n$, it natively aligns with the chromosome representation of the GA; consequently, genetic crossover and mutation operators are executed directly without requiring any intermediate continuous-to-discrete space mapping.

3.3.3. Binary Particle Swarm Optimization

In the binary PSO variant introduced by Kennedy and Eberhart[23], continuous velocity components are mapped onto a probability value within 0,1 via the sigmoid transfer function defined in Eq.(11) .

$$S(v_{id}^t) = \frac{1}{1 + e^{-v_{id}^t}} \tag{11}$$

A particle's subsequent binary position is updated through stochastic thresholding against a standard uniform random variable $r \sim U(0,1)$, as formalized in Eq. (12).

$$x_{id}^{t+1} = \begin{cases} 1, & \text{if } r < S(v_{id}^t) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Particle velocity vectors are updated using canonical PSO equations driven by distances to the personal best (p best) and swarm global best (g_{best}) positions.

3.3.4. Binary African Vultures Optimization Algorithm

The African Vultures Optimization Algorithm (AVOA), proposed by Abdollahzadeh et al.[24], models the navigation, foraging, and competitive scavenging behaviors of vultures, dynamically transitioning between global exploration and local exploitation phases based on an individual satiety rate. To adapt the continuous search operators of AVOA to binary combinatorial optimization, the S-shaped sigmoid transfer function approach proposed by Balakrishnan et al.[26] was incorporated. This mechanism maps continuous step-size velocities into the probability domain, enabling discrete updates of decision variables in binary space.

3.3.5. Discrete Tabu Search

To verify whether the observed performance patterns stem from the optimization paradigms themselves or from the continuous-to-binary transfer functions, a discrete Tabu Search (TS)[27] was integrated as a native combinatorial baseline operating directly on $x \in \{0,1\}^n$ via single-bit-flip neighborhoods. To enforce strict evaluation budget parity with the population-based algorithms ($N = 30$ evaluations per iteration across $t_{\max} = 100$ steps), TS adopts a candidate list strategy within a single trajectory: at each iteration, N candidate neighbors are generated, and the search transitions deterministically to the best candidate—even if it degrades the objective value—to escape local optima. An otherwise tabu bit-flip is accepted exclusively under the aspiration criterion if it strictly improves upon the best makespan found so far.

Flipped bit indices are subsequently locked for a dynamic tabu tenure $T_{\text{tenure}} = \text{round}(\alpha \cdot n + \beta)$ (with $\alpha = 0.1$ and $\beta \sim U(1,5)$) to suppress short-term cycling. Because exactly N candidate solutions are evaluated per iteration, TS operates under an identical cumulative budget of $N \times t_{\max} = 3000$ fitness evaluations, ensuring that any empirical performance divergence isolates search-operator efficacy rather than computational budget asymmetry.

3.4. Experimental Design

3.4.1. Benchmark Dataset

To assess the computational efficacy of the proposed model and solution algorithms, this study adopts the publicly available geometric benchmark datasets established by Agatz et al.[2] and Bouman et al.[3], which are widely recognized in the truck-drone routing literature. Although these coordinate profiles were originally tailored for discrete, node-restricted TSP-D settings, they have been adapted here to accommodate the continuous en-route interception multi-drone paradigm. Consequently, the resulting makespan values cannot be benchmarked directly against the node-constrained optimal baselines reported in the seminal works. Nonetheless, employing an established spatial layout from the literature rather than arbitrary synthetic distributions ensures high experimental reproducibility and strengthens external validity.

Fifteen benchmark instances were drawn from the publicly available 'uniform' coordinate family of the Agatz/Bouman repository, comprising five instances per scale tier selected strictly as the first five files (by instance index) within each target customer-count range. Because the source benchmark encodes customer coordinates and travel-cost factors without explicit service durations, a fixed service time of 0.3 time units—consistent with the temporal scale of the derived vehicle speeds—was assigned to each customer. Drone eligibility was governed directly by the dataset's built-in parcel-weight and no-fly (#NOVISIT) directives, where available, and the #MAXFLY directive was utilized to calibrate the drone endurance limit $E_{\max}$. The truck's fixed reference route R for each instance was generated prior to the metaheuristic executions via the deterministic Nearest Neighbor and 2-opt procedure, ensuring that all four algorithms optimized over an identical, seed-independent baseline trajectory.

The empirical evaluation is conducted across 15 problem instances categorized into three distinct dimensional scales with five instances per scale tier:

- **Small scale ($n \approx 19$):** Commensurate with classical FSTSP benchmark sizes[1], utilized primarily for model verification and baseline parametric calibration.
- **Medium scale ($n \approx 49$):** Reflective of customer densities typical of real-world urban delivery zones, serving as the core benchmark for comparative algorithmic performance.
- **Large scale ($n \approx 99$):** Designed to evaluate the scalability limits and computational robustness of the proposed event-driven decoding framework under high spatial customer densities.

3.4.2. Parameter Settings

Truck and drone cruising speeds are derived as the inverse of the per-unit-distance travel cost coefficients provided in the source benchmark, establishing a physically consistent temporal scale. To ensure a strictly impartial comparative evaluation, all four metaheuristics were configured with identical population sizes (or candidate list sizes) and equivalent cumulative fitness evaluation budgets. Furthermore, BPSO and BAVOA share the identical S-shaped sigmoid transfer function for continuous-to-binary mapping, whereas TS evaluates an equivalent number of candidate configurations per iteration ($N = 30$) to preserve evaluation parity. The termination criterion was established at $t_{\max} = 300$ iterations—yielding a cumulative budget of 9,000 function evaluations per independent run—providing a sufficiently expansive search horizon to guarantee full algorithmic convergence across both population-based and trajectory-based operators, particularly in high-dimensional instances. To eliminate stochastic bias, each algorithm was executed across 30 independent runs per instance using dedicated pseudo-random seeds. The operational and algorithmic parameters are summarized in Table 1.

Table 1. Operational and algorithmic parameter configurations

Parameter	Symbol	Value / Description
Maximum concurrent active drones	K	2
Platform launch preparation time	T_{launch}	0.5 time units
Platform docking buffer time	T_{dock}	0.5 time units
Drone flight endurance	$E_{\max}$	Instance-specific maximum-range derivative (default: 15)
Swarm / Population size	N	30
Maximum number of iterations	$t_{\max}$	300
Independent runs	—	30 per algorithm per instance
Binarization transfer function	$S(v)$	S-shaped sigmoid (shared by BPSO and BAVOA)
TS tabu tenure	θ	$\max(5, \lfloor 0.1n \rfloor)$ iterations (TS only)

3.4.3. Performance Metrics and Statistical Evaluation

For each problem instance, the arithmetic mean of the final fitness (makespan) values obtained across 30 independent runs, denoted as $\bar{f}_{A,s}$ for algorithm A and instance s, was computed. Because problem instances vary considerably in customer count and spatial dispersion, their resulting makespan values span different orders of magnitude; directly averaging raw makespan outputs would disproportionately bias aggregate conclusions toward larger instances. To eliminate scale bias and provide a normalized basis for comparative assessment, two scale-independent performance metrics were utilized, as formalized in Eqs. (12) and (13).

$$\text{AvgRank}(A) = \frac{1}{S}\sum_{s=1}^{S} \text{rank}_s(A) \tag{13}$$

$$\text{AvgGap}(A) = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{\bar{f}_{A,s} - \min_{A' \in \mathcal{A}} \bar{f}_{A',s}}{\min_{A' \in \mathcal{A}} \bar{f}_{A',s}}\right) \times 100 \tag{14}$$

where $\text{rank}_s(A)$ denotes the rank of algorithm A on instance s based on its average makespan ($1 =$ best), S represents the total number of benchmark instances ($S = 15$), and $\mathcal{A} = \{\text{GA,BPSO,BAVOA,TS}\}$ is the set of competing algorithms. The metric $\text{AvgGap}(A)$ quantifies the average relative percentage deviation of an algorithm A from the best mean objective value achieved among the compared algorithms on each instance. Accordingly, a value of $\text{AvgGap}(A) =$ 0% indicates that the algorithm A achieved the lowest mean makespan among the methods evaluated in this benchmark suite, and must not be construed as a certified global optimum, as no exact solver (e.g., MILP) or analytical lower-bounding procedure was implemented to guarantee mathematical optimality for these instances.

The statistical significance of overall performance differences among the algorithms was examined via the nonparametric Friedman test, while post hoc pairwise comparisons were evaluated using the Wilcoxon signed-rank test. Computational experiments were conducted on a workstation equipped with an Intel Core i7-10750H CPU and 16 GB of RAM. All algorithms were implemented in Python (ver. 3.10.21), and statistical tests were carried out using the scipy.stats library. To ensure the strict reproducibility of the empirical findings, a predefined sequence of fixed pseudorandom seeds was assigned across all independent runs.

4. Results and Discussion

4.1. Comparative Analysis by Problem Scale

To evaluate the scalability and search efficacy of the investigated algorithms across varying instance complexities, the average performance ranks, relative percentage gaps, and nonparametric Friedman test statistics were computed for each scale tier, as summarized in Table 2. Mean CPU execution time per run, broken down by the same scale tiers, is reported separately in Table 3.

Table 2. Average ranking, average percentage gap (Rank / %gap), and Friedman test results stratified by problem scale

Scale Tier	GA	BPSO	BAVOA	TS	Friedman $\chi 2$	p-value
Small ($n \approx 19$)	**1.00 / 0.00**	**1.00 / 0.00**	1.60 / 0.0018	**1.00 / 0.00**	3.00	0.392
Medium ($n \approx 49$)	1.40 / 0.001	3.00 / 0.17	4.00 / 0.57	**1.00 / 0.00**	14.62	**0.0022**
Large ($n \approx 99$)	1.60 / 0.048	4.00 / 2.77	3.00 / 1.73	**1.40 / 0.032**	13.56	**0.0036**
Overall (15 instances)	1.33 / 0.016	2.67 / 0.98	2.87 / 0.77	**1.13 / 0.011**	26.07	**< 0.0001**

Table 3. Mean CPU execution time per run (seconds), by algorithm and problem scale

Scale Tier	GA	BPSO	BAVOA	TS
Small ($\boldsymbol{n \approx 19}$)	9.70	9.33	10.10	9.53
Medium ($\boldsymbol{n \approx 49}$)	15.79	14.96	16.90	15.32
Large ($\boldsymbol{n \approx 99}$)	23.06	21.62	25.40	23.06

The progression of the mean algorithmic rankings across the evaluated problem tiers is visualized in Figure 1.

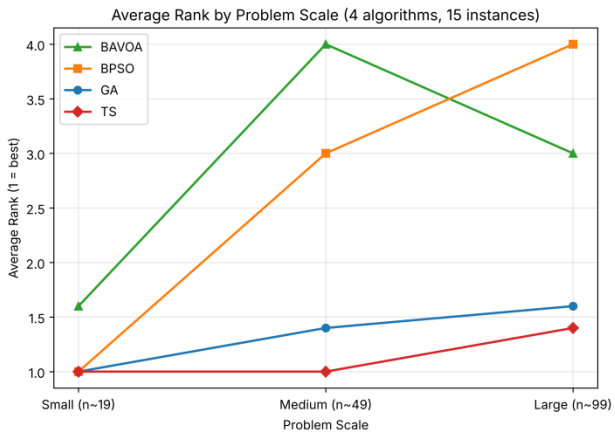


Figure 1. Trajectory of average rankings for algorithms across increasing problem scales (1="best")

For small-scale problem instances ($n \approx 19$), the relative compactness of the solution space allowed GA, BPSO, and TS to consistently converge to identical best-found makespan values across all independent runs (AvgRank $= 1.00$, AvgGap $= 0.00\%$). BAVOA achieved competitive solution quality at this scale, exhibiting a marginal average gap of only 0.0018% (AvgRank $= 1.80$). The non-parametric Friedman test confirms that these minor variations are statistically indistinguishable ($\chi^2 = 3.000,\ p = 0.392 > 0.05$).

When scaling to the medium tier ($n \approx 49$), the combinatorial expansion of the decision space renders performance divergences among the algorithms statistically significant ($\chi^2 = 14.617,\ p = 0.0022 < 0.05$). At this scale, TS attains the lowest average rank across all four algorithms (AvgRank $= 1.00$, AvgGap $= 0.00\%$), narrowly outperforming GA (AvgRank $= 1.40$, AvgGap $= 0.001\%$); notably, both native-discrete algorithms substantially outperform BPSO (AvgGap $= 0.17\%$) and BAVOA (AvgGap $= 0.57\%$).

For large-scale instances ($n \approx 99$), GA and TS remain closely matched and jointly dominant (AvgRank $= 1.60$ and 1.40; AvgGap $= 0.048\%$ and 0.032%, respectively). Conversely, BPSO's search capability deteriorates substantially as dimensionality expands, dropping to the lowest rank (AvgRank $= 4.00$) with an average gap of 2.77%, while BAVOA trails with an average gap of 1.73%.

Considering all 15 instances concurrently, the omnibus Friedman test ($\chi^2 = 26.068,\ p < 0.0001$) confirms that the divergence in search capability is strongly scale-dependent; crucially, this statistical gap demarcates the two native-discrete paradigms (GA, TS) from the continuous-domain, sigmoid-binarized metaheuristics (BPSO, BAVOA), rather than isolating a single dominant algorithm. Furthermore, runtime evaluations reveal that this performance advantage was not achieved at the expense of disproportionate computational overhead: at the largest scale, GA and TS exhibit virtually identical mean per-run CPU times (23.06 s vs. 23.06 s), fully consistent with the equal-evaluation-budget design. In contrast, BAVOA incurred the highest computational latency across all scales while delivering inferior solution quality.

4.2. Pairwise Statistical Comparisons

To rigorously examine the pairwise differences in overall algorithmic performance across the 15 benchmark instances, post-hoc non-parametric Wilcoxon signed-rank tests were performed on the instance mean makespan values. The resulting test statistics (W) and asymptotic p-values are summarized in Table ***4***.

Table 4. Pairwise Wilcoxon signed-rank test results across 15 benchmark instances

Comparison Pair	Wilcoxon Statistic (W)	p-value	Statistical Significance (α=0.01)
GA vs. TS	10.000	0.499	Not significant ($p > 0.05$)
GA vs. BAVOA	0.000	0.0033	Statistically significant ($p < 0.01$)
GA vs. BPSO	0.000	0.0051	Statistically significant ($p < 0.01$)
TS vs. BAVOA	0.000	0.0033	Statistically significant ($p < 0.01$)
TS vs. BPSO	0.000	0.0051	Statistically significant ($p < 0.01$)
BAVOA vs. BPSO	21.000	0.286	Not significant ($p > 0.05$)

The Wilcoxon signed-rank test results confirm that the performance differential between GA and TS is not statistically significant ($W = 10.000,\ p = 0.499$): across the 15 benchmark instances, neither native-discrete method establishes a definitive statistical dominance over the other. By contrast, both GA and TS individually outperform BAVOA ($p = 0.0033$ in both cases) and BPSO ($p = 0.0051$ in both cases) at the stringent $\alpha = 0.01$ significance threshold. Crucially, the zero test statistics ($W = 0.000$) demonstrate that this dominance is absolute across every non-tied instance. This outcome is directly attributable to the high representational fidelity that native binary search operators—both GA's recombination/mutation mechanics and TS's bit-flip neighborhood moves—provide when navigating discrete decision landscapes, irrespective of whether the underlying architecture is population-based or trajectory-driven.

Conversely, the pairwise comparison between BAVOA and BPSO yields no statistically significant divergence ($W = 21.000,\ p = 0.286$). This indicates that both continuous-domain metaheuristics encounter comparable structural bottlenecks during the transfer-function-based binarization process, confirming that the choice between their continuous-space foraging dynamics is secondary to the performance penalty imposed by sigmoid-induced saturation.

4.3. Convergence Behavior and Solution Stability

To evaluate the search dynamics of the algorithms and detect potential susceptibility to premature convergence, the trajectory of the best-so-far fitness value across successive iterations was recorded. Figure 2 illustrates the mean convergence profiles computed over 30 independent runs for a representative large-scale benchmark instance ($n \approx 99$). GA exhibits rapid initial convergence, characterized by a steep descent trajectory, reaching high-quality solution basins within the early generations due to its native binary representation and elitist selection strategy. TS, by contrast, starts with a comparatively higher makespan and converges more gradually—a direct consequence of its single-bit-flip neighborhood, which navigates the Boolean hypercube along single coordinate projections rather than via population-wide recombination. However, TS steadily closes the gap with GA across the extended iteration horizon, with both curves converging to statistically indistinguishable final makespan values by iteration 300. This trajectory provides empirical justification for establishing the termination criterion at $t_{\max} = 300$ generations: a more restrictive evaluation horizon would prematurely truncate the trajectory-based search, obfuscating the asymptotic competitiveness of discrete local search against genetic recombination. Meanwhile, BAVOA demonstrates an effective capability in escaping local optima during intermediate iterations, driven by its hunger-rate-regulated global exploration phase. Conversely, BPSO displays a sluggish, stepwise convergence trajectory that plateaus substantially above the GA/TS basin; this premature stagnation stems from stochastic bit oscillations and probability saturation induced by the sigmoid transfer function.

Beyond aggregate convergence trends, the algorithmic robustness and run-to-run reproducibility of final solution quality are essential for practical operational deployment. A variance analysis on large-scale instances ($n \approx 99$) shows that the two native-discrete methods again form a distinct, lower-variance cluster: the average run-to-run standard deviation of the final makespan is $\sigma \approx 1.30$ for GA and $\sigma \approx 1.41$ for TS, compared with $\sigma \approx 3.35$ for BPSO and $\sigma \approx 3.33$ for BAVOA. This confirms that both GA and TS sustain reliable exploitation toward the best-found solution basin and exhibit minimal sensitivity to initial-state stochasticity, whereas the two continuous-domain, sigmoid-binarized methods are markedly more erratic across independent runs.

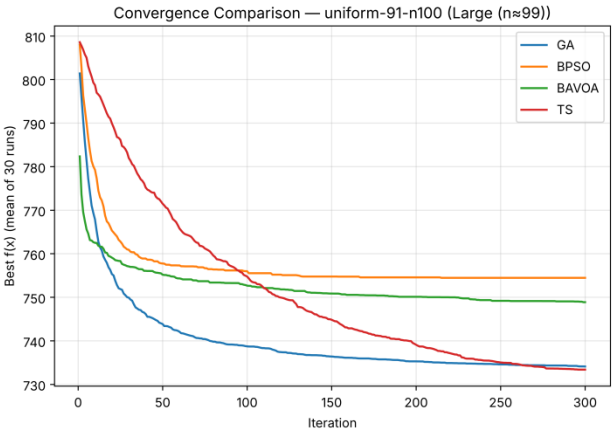


Figure 2. Average over 30 independent runs convergence profiles of algorithms on a representative problem instance.

Figure 3 complements this run-level analysis with an across-instance perspective: it presents box plots of the relative percentage gap, $\mathrm{Gap}_s(A) = 100 \times \left(\bar{f}_{A,s} - \min_{A'} \bar{f}_{A',s}\right) / \min_{A'} \bar{f}_{A',s}$, attained by each algorithm across all 15 benchmark instances. This visualization demonstrates not only how closely each algorithm approaches the best-found solution on individual instances, but also how consistently it performs across problem topologies that differ substantially in scale and spatial customer dispersion.

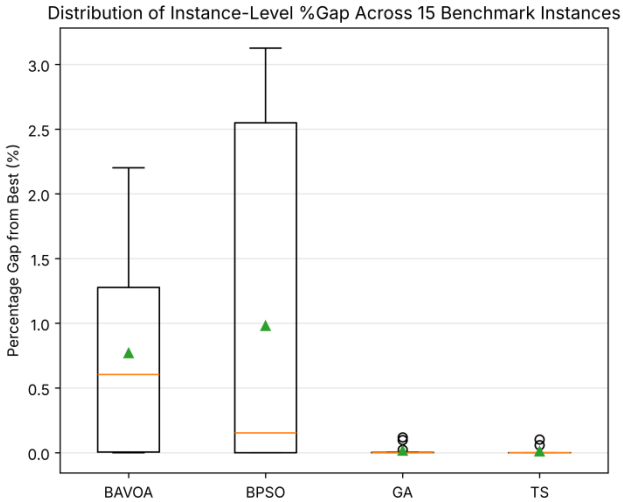


Figure 3. Distribution, across the 15 benchmark instances, of each algorithm's percentage gap from the best solution found on that instance.

Consistent with Table 2, GA and TS form a tight, near-zero-gap cluster across nearly all benchmark instances, whereas BPSO and BAVOA exhibit both higher median gaps and substantially wider interquartile ranges, reflecting their pronounced sensitivity to instance-specific spatial topologies.

To elucidate the spatio-temporal coordination inherent in the en-route interception model, a representative delivery schedule optimized by GA is depicted in Figure 4. Given

that GA and TS achieve statistically indistinguishable solution quality, GA serves as an illustrative exemplar of the native-discrete solution topology. The visualization highlights the truck's primary route, drone flight paths, serviced customer locations, and dynamic rendezvous coordinates along active road segments. Geographically dispersed customers situated away from the primary transport corridor are delegated to drone sorties, while the truck traverses the main arterial network. Crucially, the truck intercepts returning drones en route rather than remaining idle at discrete customer nodes. This synchronized coordination substantially diminishes vehicle dwell times, directly securing the observed reductions in total operational makespan.

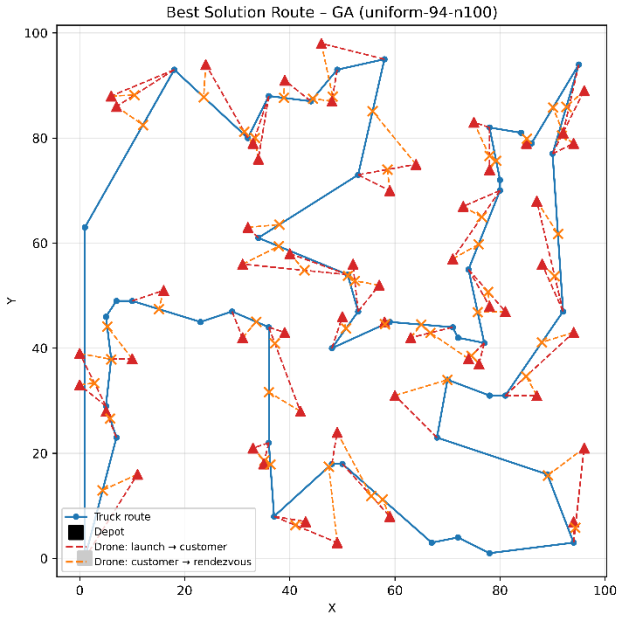


Figure 4. Representative GA-optimized delivery schedule illustrating the truck's primary route, drone launch/rendezvous points, and the en-route interception geometry along active road segments.

4.4. Discussion

The experimental results demonstrate that the search capability and scalability of metaheuristic optimization algorithms are strictly conditioned by problem dimensionality and, more fundamentally, by whether the search operators act directly on the problem's native binary representation. For small-scale instances ($n \approx 19$), the compactness of the combinatorial solution space and the well-defined basin of attraction surrounding the best-found solution allow all four evaluated methods to reliably converge to the same best-found makespan value. However, as problem dimensionality increases ($n \geq 49$), the exponential expansion of the discrete search space (2^n) coupled with highly nonlinear temporal and spatial synchronization constraints, makes the algorithmic search operators, as well as their structural compatibility with the underlying decision topology, the decisive performance determinant.

The central empirical finding of this study is that this determinant is the representation an algorithm operates, rather than any single algorithm's inherent superiority. GA and TS represent fundamentally distinct search paradigms, with the former operating as a population-based evolutionary method with crossover and elitism, and the latter functioning as a single-trajectory local search governed by a tabu list. Nevertheless, both act directly on the binary decision vector $x \in \{0,1\}^n$ without an intermediate continuous mapping, and both achieve statistically indistinguishable top performance at every scale tested. Conversely, BPSO and BAVOA are structurally distinct nature-inspired metaheuristics that both rely on an S-shaped sigmoid transfer function to project continuous positions into binary assignments. Consequently, both are statistically significantly outperformed by GA and TS alike, while exhibiting no significant difference from each other. This two-against-two separation, which cuts across population-based versus trajectory-based search and classical versus recently introduced nature-inspired paradigms, provides considerably stronger evidence for the representation-fidelity hypothesis than a single-algorithm comparison could offer, as it is replicated by two structurally independent discrete methods rather than resting on GA alone.

The sigmoid-based indirect mapping shared by BPSO and BAVOA introduces two pronounced structural deficiencies:

- Identical velocity and continuous position vectors can probabilistically map to divergent binary strings across iterations, injecting uncontrolled stochastic noise into the search trajectory and disrupting the retention of elite solutions.
- The asymptotic properties of the sigmoid function cause large-magnitude velocity and position updates to rapidly saturate probability values toward extreme limits. This saturation phenomenon severely depletes swarm diversity, particularly in high-dimensional settings ($n \approx 99$), forcing premature convergence into suboptimal local basins.

The consistently minimal variance and low standard deviation exhibited by both GA and TS across 30 independent runs provide robust empirical validation of this structural explanation. Notably, TS required the full 300-iteration budget to close its initial gap with GA, indicating that native discrete local search is not inherently faster to converge than population-based search, but rather that, given an equivalent evaluation budget, it ultimately reaches comparably strong solutions. These observations align

strictly with the foundational premise of the No Free Lunch (NFL) theorem[28], which establishes that no single metaheuristic possesses universal dominance across disparate problem domains and that optimization efficacy depends intrinsically on how well an algorithm's inductive biases and operators match the problem's topological representation. This study substantiates this principle through two independently structured discrete algorithms rather than relying on a single baseline.

From an applied logistics and decision-support standpoint, these findings offer critical design guidelines. The event-driven forward decoding procedure, reinforced by protective ground-wait and segment-locking mechanisms, deterministically resolves physical feasibility for every candidate vector without requiring artificial penalty-function tuning. Consequently, the observed performance disparities stem entirely from the fundamental exploration and exploitation efficiencies of the search operators over the raw binary space, rather than from artifacts of constraint handling. Contrary to the prevailing assumption in applied optimization that recently introduced nature-inspired optimizers inherently outperform classical paradigms, this study methodologically demonstrates, through both a discrete trajectory baseline and a population-based evolutionary method, that alignment between representation fidelity and operator mechanics is far more decisive than either algorithmic novelty or the population-based versus trajectory-based distinction.

Finally, the concurrent evaluation using AvgRank, AvgGap, and mean CPU execution time provided an unbiased and standardized framework across 15 topologically diverse benchmark instances. This metric integration effectively eliminated scale-magnitude distortions and zero-variance anomalies that arise from single-instance comparisons, enabling a statistically rigorous validation of algorithmic divergence through nonparametric hypothesis testing, while the CPU-time results confirmed that this divergence was not an artifact of unequal computational budgets.

4.5. Limitations and Future Work

While the en-route interception multi-drone delivery model and the event-driven decoding architecture developed in this study demonstrate significant computational and operational efficacy, several structural limitations warrant discussion, outlining directions for future research.

In the current formulation, the truck's baseline delivery sequence is fixed in advance using Nearest Neighbor and 2-opt local search heuristics to preserve computational tractability and maintain the single-pass $O(n)$ complexity of the decoding procedure. Future research could explore integrated two-stage or hybrid metaheuristic frameworks, such as coupling with Adaptive Large Neighborhood Search (ALNS), wherein truck-visitsequences and drone dispatch decisions are optimized simultaneously.

The operational model assumes that each drone sortie serves exactly one customer before rendezvousing with the truck. Given ongoing advancements in UAV payload capacity, incorporating multi-drop sorties, where a drone delivers parcels to multiple geographically clustered customers within a single flight, could substantially enhance operational flexibility and energy efficiency.

Vehicle cruising speeds, customer service times, and battery discharge rates are treated deterministically, with energy consumption modeled as a linear function of flight duration. Integrating stochastic parameters reflective of real-world operational environments, such as payload-dependent non-linear battery depletion, dynamic aerodynamic wind vectors, and traffic-induced road speed variations, would further enhance the model's practical fidelity.

The current problem setting is confined to a single truck and a homogeneous drone fleet. Generalizing the formulation to multi-truck logistics systems that support cross-truck docking, wherein a drone launches from one vehicle and recovers onto a different, intersecting mobile platform, represents a promising extension for dense regional distribution networks.

Finally, this study evaluated solution quality relative to the best-found metaheuristic schedules without calculating theoretical lower bounds through exact mathematical programming. Developing exact branch-and-cut algorithms or matheuristic decomposition procedures to formally benchmark continuous en-route rendezvous solutions on small-to medium-sized instances represents another valuable research trajectory.

5. Conclusion

This study addressed an extended last-mile delivery framework coordinating a conventional truck with a multi-drone fleet, wherein drones are capable of intercepting the moving truck continuously along road segments. Unlike classical node-restricted TSP-D variants, an analytical, continuous-time rendezvous geometry was formulated, eliminating the requirement for the truck to idle at fixed customer nodes. This operational scheme was coupled with a protective ground-wait strategy to mitigate the physical bottleneck imposed by the single launch-and-recovery platform, alongside a dynamic segment-locking mechanism that safeguards forward schedule feasibility. Consequently, this study introduces a deterministic, single-pass, event-

driven decoding architecture that maps any candidate binary decision vector into a physically valid delivery schedule with $O(n)$ computational complexity, entirely circumventing external penalty functions or computationally intensive backtracking routines.

The proposed optimization framework was benchmarked across 15 standard reference instances spanning small ($n \approx 19$), medium ($n \approx 49$), and large ($n \approx 99$) dimensional scales, comparing a Genetic Algorithm (GA), Binary Particle Swarm Optimization (BPSO), the Binary African Vultures Optimization Algorithm (BAVOA), and a classical discrete Tabu Search (TS) baseline under an equal function-evaluation budget. Non-parametric statistical tests, specifically the Friedman and Wilcoxon signed-rank tests, confirmed that algorithmic performance divergences widen significantly as problem scale increases, reaching high statistical significance ($p < 0.01$). Crucially, this divergence separates two distinct classes of algorithms rather than isolating a single winner. GA and TS, both operating via native discrete binary search operators, achieved statistically indistinguishable top performance, recording average ranks of 1.33 and 1.13 as well as average percentage gaps of 0.016% and 0.011% respectively (Wilcoxon $p = 0.499$). Both methods significantly and consistently outperformed the sigmoid-binarized BPSO and BAVOA ($p \leq 0.0051$ in every pairwise comparison), which, in turn, did not differ significantly from one another. This advantage was achieved without disproportionate computational overhead, as the mean CPU execution time per run for GA and TS was statistically indistinguishable at every problem scale, confirming that all four algorithms were evaluated under an equivalent computational budget.

These empirical findings deliver critical theoretical and methodological insights for discrete combinatorial optimization. The joint, structurally independent success of GA and TS indicates that representation fidelity, defined by operating directly on the Boolean hypercube without an intermediate continuous mapping, is the primary driver of search quality in this problem class, rather than any property specific to population-based search, evolutionary operators, or algorithmic novelty. Conversely, continuous-domain metaheuristics relying on stochastic sigmoid transfer functions introduce uncontrolled variance and suffer from rapid probability saturation as dimensionality grows (2^n), prematurely depleting search diversity regardless of the specific nature-inspired dynamics driving their continuous-space updates. In strict accordance with the No Free Lunch theorem, these results substantiate that aligning search operator mechanics with the problem's underlying decision topology is substantially more critical than pursuing algorithmic novelty in isolation.

From an applied logistics and decision-support perspective, this study provides concrete empirical evidence that continuous en-route synchronization significantly curtails total delivery makespan. The proposed analytical rendezvous formulation and event-driven decoding procedure deliver an optimization backbone that is scalable, computationally lightweight, and operationally dependable for high-density, real-time last-mile distribution networks. Furthermore, the availability of two structurally distinct, equally effective native-discrete solution methods gives practitioners substantial implementation flexibility without compromising solution quality.

Declaration of Ethical Standards

The author of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Conflict of Interest

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author declares that generative artificial intelligence tools were utilized solely for language editing and translation support during the preparation of this manuscript. The intellectual content, analysis, and conclusions remain entirely the responsibility of the author. All outputs produced by generative AI were critically reviewed and revised to ensure accuracy, clarity, and compliance with academic standards.

References

[1] Murray CC, Chu AG. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies. 2015;54:86-109. Available from: https://doi.org/10.1016/j.trc.2015.03.005

[2] Agatz N, Bouman P, Schmidt M. Optimization approaches for the traveling salesman problem with drone. Transportation Science. 2018;52(4):965-981. Available from: https://doi.org/10.1287/trsc.2017.0791

[3] Bouman P, Agatz N, Schmidt M. Dynamic programming approaches for the traveling salesman problem with drone. Networks. 2018;72(4):528-542. Available from: https://doi.org/10.1002/net.21864

[4] Macrina G, Pugliese LDP, Guerriero F, Laporte G. Drone-aided routing: A literature review. Transportation Research Part C: Emerging Technologies. 2020;120:102762. Available from: https://doi.org/10.1016/j.trc.2020.102762

[5] Moshref-Javadi M, Winkenbach M. Applications and research avenues for drone-based models in logistics: A classification and review. Expert Systems with Applications. 2021;177:114854. Available from: https://doi.org/10.1016/j.eswa.2021.114854

[6] Marinelli M, Caggiani L, Ottomanelli M, Dell'Orco M. En route truck–drone parcel delivery for optimal vehicle routing strategies. IET Intelligent Transport Systems. 2018;12(4):253-261. Available from: https://doi.org/10.1049/iet-its.2017.0227

[7] Schermer D, Moeini M, Wendt O. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. Computers & Operations Research. 2019;109:134-158. Available from: https://doi.org/10.1016/j.cor.2019.04.021

[8] Salama MR, Srinivas S. Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. Transportation Research Part E: Logistics and Transportation Review. 2022;164:102788. Available from: https://doi.org/10.1016/j.tre.2022.102788

[9] Moshref-Javadi M, Hemmati A, Winkenbach M. A comparative analysis of synchronized truck-and-drone delivery models. Computers & Industrial Engineering. 2021;162:107648. Available from: https://doi.org/10.1016/j.cie.2021.107648

[10] Murray CC, Raj R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transportation Research Part C: Emerging Technologies. 2020;110:368-398. Available from: https://doi.org/10.1016/j.trc.2019.11.003

[11] Kitjacharoenchai P, Min BC, Lee S. Two echelon vehicle routing problem with drones in last mile delivery. International Journal of Production Economics. 2020;225:107598. Available from: https://doi.org/10.1016/j.ijpe.2019.107598

[12] Luo Z, Poon M, Zhang Z, Liu Z, Lim A. The multi-visit traveling salesman problem with multi-drones. Transportation Research Part C: Emerging Technologies. 2021;128:103172. Available from: https://doi.org/10.1016/j.trc.2021.103172

[13] Luo Z, Gu R, Poon M, Liu Z, Lim A. A last-mile drone-assisted one-to-one pickup and delivery problem with multi-visit drone trips. Computers and Operations Research. 2022;148:106015. Available from: https://doi.org/10.1016/j.cor.2022.106015

[14] Liu Y, Liu Z, Shi J, Wu G, Pedrycz W. Two-echelon routing problem for parcel delivery by cooperated truck and drone. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2021;51(12):7450-7465. Available from: https://doi.org/10.1109/TSMC.2020.2968839

[15] Meng S, Chen Y, Li D. The multi-visit drone-assisted pickup and delivery problem with time windows. European Journal of Operational Research. 2024;314(2):685-702. Available from: https://doi.org/10.1016/j.ejor.2023.10.021

[16] Thomas T, Srinivas S, Rajendran C. Collaborative truck multi-drone delivery system considering drone scheduling and en route operations. Annals of Operations Research. 2024;339(1):693-739. Available from: https://doi.org/10.1007/s10479-023-05418-y

[17] Han B, Chi T, Sun Z. Collaborative delivery problem of trucks and multiple drones: An en-route operation with flexible launch positions. International Transactions in Operational Research. 2026;33:4072-4109. Available from: https://doi.org/10.1111/itor.70045

[18] Schermer D, Moeini M, Wendt O. Algorithms for solving the vehicle routing problem with drones. In: Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Part I. Cham: Springer; 2018. p. 352-361. Available from: https://doi.org/10.1007/978-3-319-75417-8_33

[19] Han X, Shen J. Vehicle routing problem with trucks and drones collaboration: A structured literature review. Applied Computing and Intelligence. 2025;5(2):348-358. Available from: https://doi.org/10.3934/aci.2025020

[20] Zhou J, Yi J, Yang Z, Pu H, Li X, Luo J, Gao L. A survey on vehicle–drone cooperative delivery operations optimization: Models, methods, and future research directions. Swarm and Evolutionary Computation. 2025;92:101780. Available from: https://doi.org/10.1016/j.swevo.2024.101780

[21] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press; 1975.

[22] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks. 1995. p. 1942-1948. Available from: https://doi.org/10.1109/ICNN.1995.488968

[23] Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. 1997. p. 4104-4109. Available from: https://doi.org/10.1109/ICSMC.1997.637339

[24] Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering. 2021;158:107408. Available from: https://doi.org/10.1016/j.cie.2021.107408

[25] Zheng R, Hussien AG, Qaddoura R, Jia H, Abualigah L, Wang S, Saber A. A multi-strategy enhanced African vultures optimization algorithm for global optimization problems. Journal of Computational Design and Engineering. 2023;10(1):329-356. Available from: https://doi.org/10.1093/jcde/qwac135

[26] Balakrishnan K, Dhanalakshmi R, Seetharaman G. S-shaped and V-shaped binary African vulture optimization algorithm for feature selection. Expert Systems. 2022;39(10):e13079. Available from: https://doi.org/10.1111/exsy.13079

[27] Glover F. Tabu Search—Part I. ORSA Journal on Computing. 1989;1(3):190-206. Available from: https://doi.org/10.1287/ijoc.1.3.190

[28] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1997;1(1):67-82. Available from: https://doi.org/10.1109/4235.585893