Review Article

Engineering: Open Access

Genetic Algorithm vs Firefly for a Single-Truck, Single-Drone Vehicle Routing Problem

Janith Thenura Akuratiya Gamge*

Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka

***Corresponding Author**
Janith Thenura Akuratiya Gamge, Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka.

Submitted: 2025, Oct 15; **Accepted:** 2025, Nov 16; **Published:** 2025, Nov 24

Citation: Gamge, J. T. A. (2025). Genetic Algorithm vs Firefly for a Single-Truck, Single-Drone Vehicle Routing Problem. *Eng OA, 3*(11), 01-04.

Abstract
We study a practically motivated variant of the Vehicle Routing Problem with Drones (VRP-D): a single truck collaborates with a single drone to serve customers. On each truck edge ($i \rightarrow j$), at most one drone sortie may be launched at i, visit exactly one customer k, and rendezvous at j. The edge time is the maximum of the truck's travel and service time versus the drone's sortie time (including launch and recovery). We compare two popular metaheuristics—Genetic Algorithm (GA) and Firefly Algorithm (FF)—against a simple baseline (Nearest-Neighbor + 2-opt with greedy drone assignment). We implement a reproducible experiment suite, parse parameters and a 20-customer instance from provided files, and report best-time (makespan) performance over multiple random seeds. On the 20-customer instance, FF tends to achieve lower total time than GA (Mann–Whitney: $z \approx -1.73$, $p \approx 0.083$; Cliff's $\delta \approx -0.531$, large, negative favors FF). The baseline offers a fast, competitive reference. We release code, datasets, and plots to support replication.

Keywords: VRP with Drones, TSP-D, Last-mile Logistics, Genetic Algorithm, Firefly Algorithm, Metaheuristics, Non-Parametric Statistics

1. Introduction

Demand for fast and low-cost last-mile delivery motivates hybrid solutions in which a truck launches a small unmanned aerial vehicle (drone) to serve some customers while the truck continues its route. This gives rise to the Traveling Salesman Problem with Drone (TSP-D) and related VRP-D variants, which have been actively studied in the last decade. We consider a minimal yet operationally meaningful variant—one truck, one drone, fly-return with rendezvous at the next truck node—and benchmark two widely used metaheuristics: Genetic Algorithm (GA) and Firefly Algorithm (FF).

1.1. Contributions

1. We formalize a clear single-truck, single-drone variant with explicit timing and feasibility rules.
2. We implement discrete GA and FF with a shared greedy drone-assignment layer that keeps the search space over truck permutations only.
3. We provide a fully reproducible experiment harness (instances, parameters, seeds, scripts) and a statistical comparison (Mann–Whitney U and Cliff's δ).
4. We report initial evidence that FF can outperform GA on the studied instance, with a simple baseline offering a strong reference.

2. Related Work

Early formalizations of truck-and-drone routing include the Flying Sidekick Traveling Salesman Problem (FS-TSP) and the TSP-D, along with exact and heuristic approaches. Reviews and later models cover richer synchronization and fleet settings. Metaheuristics such as GA are longstanding tools in VRP, while the Firefly Algorithm [4, 5] is a nature-inspired method shown effective on multimodal search. We build on this foundation, focusing on a compact, reproducible benchmark; see, e.g., [1, 2, 3] and general GA–VRP surveys.

3. Problem Definition

We define a single-truck, single-drone fly-return with rendezvous at next truck node variant.

- **Customers:** set $C = \{1, . . . , N\}$, depot node 0 at (0, 0).
- **Truck route:** $\pi = [0, \pi_1, . . . , \pi_N , 0]$ visits each customer at most once.
- **Drone sorties:** On each truck edge $(i \to j)$ (consecutive nodes in π), at most one drone sortie is allowed: launch at i, visit exactly one customer $k \in C$ not served by truck, and rendezvous at j.
- **Times:** truck travel time is Euclidean distance divided by truck speed; truck service time applies at each non-depot truck stop. Drone sortie time includes launch, drone travel $i \to k \to j$, service at k, and recovery. Edge duration is max !truck($i \to j$) + service at j, drone sortie".

- **Feasibility:** (i) each customer served exactly once (truck or drone); (ii) drone capacity and endurance respected; (iii) truck maximum driving time respected; (iv) one drone sortie per edge.
- **Objective:** minimize total completion time (makespan). We also report truck/drone distances and a simple proxy cost = $\alpha d_{\text{truck}} + \beta d_{\text{drone}}$.

Simplification. Truck capacity is treated as a global bound at the depot (the truck carries all goods from depot), appropriate for the small instances considered, and stated explicitly to keep the model compact.

4. Methods

4.1. Data and Parameters

We parse a 20-customer instance (coordinates, demands) from a text file and experiment parameters from a parameter file (truck/drone speeds, service/launch/recovery times, endurance, max driving time, and cost factors). Units are harmonized to minutes and miles per minute; the depot is fixed at (0, 0).

4.2. Greedy Drone Assignment (Shared Layer)

For a given truck route π, we decide at most one drone customer per edge greedily. For each edge $(i \to j)$, among drone-eligible customers not on the truck route, we choose the k that yields the largest positive time saving while respecting endurance and capacity. This provides a fast, consistent decoding from permutations to full truck-and-drone schedules.

4.3. Baseline (NN + 2-opt)

Construct a truck-only tour with Nearest-Neighbor, improve via 2-opt, then apply the greedy drone assignment. This yields a simple, deterministic baseline of reasonable quality in seconds.

4.4. Genetic Algorithm (GA)

Solutions are permutations of customers; the decoded route is [0, perm, 0] and drone sorties are assigned greedily as above. We use tournament selection, order crossover (OX), and swap mutation, with elitism. Stopping uses a fixed number of generations.

Default hyperparameters (20-customer): population = 40–60, generations = 40–60, crossover rate = 0.9, mutation rate = 0.2, elitism = 2.

4.5. Firefly Algorithm (FF, Discrete)

We adapt FF to permutations. Brightness is inverse fitness (lower time ⇒ brighter). Distance between permutations is the Hamming distance over positions. Attractiveness $\beta = \beta_0 e^{-\gamma r2}$. The move-toward operator repeatedly applies targeted swaps that align the current permutation with a brighter neighbor; with small probability α, a random swap adds exploration. Stopping uses a fixed number of iterations.

Default hyperparameters (20-customer): swarm size = 40–60, iterations = 40–60, $\alpha = 0.3$, $\beta_0 = 1.0$, $\gamma = 0.01$.

4.6. Evaluation Metrics and Statistics

Primary metric is best-so-far total time (lower is better). We run multiple random seeds per algorithm and summarize with median/mean/min/max. Since normality is not assumed, we use Mann–Whitney U (two-sided) and report U , z, p. We also report Cliff's delta with conventional magnitude thresholds (negligible/small/medium/large).

4.7. Reproducibility

We provide code, data, and scripts to (i) parse inputs; (ii) run GA/FF/Baseline across seeds;
(iii) compute statistics; (iv) plot GA vs. FF convergence (median + IQR). Artifacts are listed in Appendix A.

5. Experiments

5.1. Setup

- **Instance:** 20 customers from the provided file; parameters from the parameter file (truck speed ≈ 35/60 mi/min; drone speed ≈ 50/60 mi/min; drone capacity = 5; endurance = 30 min; launch/recovery = 1 min each; truck max driving time = 8 h).
- **Runs:** GA and FF with 8 seeds each (demo scale); Baseline with 5 seeds.
- **Compute:** commodity desktop; Python implementation.

5.2. Results (20-Customer)

Statistical test (GA vs FF): Mann–Whitney $z \approx -1.73$, $p \approx 0.083$ (two-sided). Cliff's $\delta \approx -0.531$ ⇒ large magnitude; negative favors FF because lower is better. Convergence curves (median with IQR) show FF tends to reach strong solutions earlier on this instance, while GA exhibits larger variability and occasional outliers.

alg	median	mean	min	max	count
baseline	76.76	77.08	76.76	78.08	5
ff	80.98	81.08	78.67	83.40	8
ga	83.84	83.38	76.09	87.29	8

Table 1: Summary (best time, minutes). Lower is Better

5.3. Larger Instances (Quick Checks)

We include reproducible synthetic instances of 50 and 100 customers (uniform square; integer demands 1–5) and append small-budget runs as sanity checks (for publication-grade inference, we recommend ≥ 20 seeds and proportionally larger iteration budgets).

alg	median	mean	min	max	count
baseline	318.95	316.63	310.94	320.00	3
ff	729.10	728.71	716.19	738.54	5
ga	751.55	751.46	726.45	778.38	5

Table 2: Customers_50 Summary (best time, minutes)

alg	median	mean	min	max	count
baseline	524.45	523.32	520.65	524.85	3
ff	1640.01	1640.84	1627.22	1656.12	4
ga	**1599.94**	**1608.34**	**1566.67**	**1666.80**	**4**

Table 3: Customers_100 Summary (best time, minutes)

6. Discussion

The discrete FF with a targeted swap move and Hamming-distance attractiveness performs competitively and, on the 20-customer instance, tends to dominate GA under our settings. The baseline is close enough to be a practical reference and is very fast to compute. The greedy per-edge drone assignment decouples the complicated truck–drone synchronization from the metaheuristic search, yielding robust decoding and consistent feasibility checks; however, it can miss complex multi-edge trade-offs.

When might GA win? GA's population dynamics and recombination can shine with larger iteration budgets, hybrid local search (e.g., 2-opt/3-opt), or more structured crossover tailored to routing. In our quick runs, GA occasionally produced the best single outlier solution (best min), suggesting potential with deeper tuning.

7. Threats to Validity

- **Internal:** We used a specific greedy drone assignment; other decoders (or joint truck–drone neighborhoods) may change relative rankings.
- **External:** Results are from a single 20-customer instance (plus small-budget larger demos). Different geographies, demand distributions, or timing parameters may alter behavior.
- **Statistical power:** Demo runs used modest seed counts; we recommend ≥ 20–30 seeds for publication-grade inference.
- **Simplifications:** Global truck capacity and single-drone/ one-sortie-per-edge assumptions keep the model compact; richer variants exist (multiple drones, time windows, flexible rendezvous).

8. Conclusion and Future Work

We presented a compact, reproducible comparison of GA and FF on a single-truck, single-drone VRP-D variant. The discrete FF with a simple move-toward operator outperformed GA in median best time on our 20-customer instance, with a strong baseline providing context. Future work includes (i) hybrid GA↔FF operators and local search; (ii) dynamic scenarios (e.g., on-the- fly customer changes); (iii) richer synchronization (multi-sortie per edge, flexible rendezvous); and (iv) extended benchmarks with public instance sets.

Acknowledgments

(Optional) Acknowledge support, feedback, or dataset providers.

References

1. Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies, 54*, 86-109.
2. Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science, 52*(4), 965-981.
3. Poikonen, S., Golden, B., & Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing, 31*(2), 335-346.

4. Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg.

5. Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg.

Appendix A. Reproducibility Artifacts

- **Code & data:** vrpd_final.zip
- **Results:** results_ga_ff_baseline.csv
- **Plots:** plots/conv_ga_vs_ff_20.png, plots/baseline_20.png
- **Scripts:** scripts/run_experiments.py, scripts/stats.py, scripts/plot_convergence.py, scripts/gen_instances.py, scripts/run_quick_large.py

Appendix B. Extended Results (50 & 100 customers; relaxed max drive-time)

See Tables above for summary statistics over quick sanity-check runs.