An Adapted Cat and Mouse Based Optimizer for Solving the Capacitated Vehicle Routing Problem

Duvvuri Prathamesh, Neeraj Ramprasad, SN Omkar

ICOA 2025

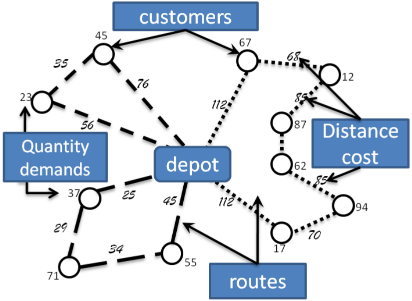
September 14, 2025

Table of Contents

Introduction

Capacitated Vehicle Routing Problem

1. CVRP : Extends the Traveling Salesman Problem to multiple vehicles with capacity limits, starting and ending at a single depot.
2. The objective is to design minimum-cost delivery routes that serve all customers while respecting vehicle capacities.
3. The main challenge is that CVRP is NP-hard; exact methods struggle with large instances, motivating heuristic and metaheuristic approaches.



Problem Description and Mathematical Modelling

Capacitated Vehicle Routing Problem (CVRP):

- Set of customers with known demand, served by a fleet of identical vehicles.
- Each vehicle starts/ends at a depot and cannot exceed capacity Q.
- Objective: minimize the **total travel cost (distance)**.

Mathematical Formulation:

$$\text{Minimize: } Z = \sum_{i=0}^{n} \sum_{j=0, j \neq i}^{n} \sum_{k=1}^{m} c_{ij} x_{ij}^{k}$$

$$\text{s.t. } \sum_{k=1}^{m} \sum_{i=0, i \neq j}^{n} x_{ij}^{k} = 1, \quad \forall j \text{ (each customer visited once)}$$

$$\sum_{j=1}^{n} \sum_{i=0, i \neq j}^{n} q_j x_{ij}^{k} \leq Q, \quad \forall k \text{ (capacity constraint)}$$

$$\sum_{i \in S} \sum_{j \in S, i \neq j} x_{ij}^{k} \leq |S| - 1, \quad \forall S \subset V \text{ (sub-tour elimination)}$$

Literature Review

Literature Review – Metaheuristics for CVRP

- Metaheuristics gained prominence for CVRP since the 1990s.
- Population-based, nature-inspired algorithms dominate the field.
- Key approaches:
  - PSO (1995/2009) – Models bird flocking, strong exploration, but weak diversity later
  - ACO (1999/2000) – inspired by ant foraging, effective with local search, but has poor scalability
  - Hybrids (2010s–2020s) - Firefly, Grey Wolf Optimizer, Enhanced Artificial Bee Colony

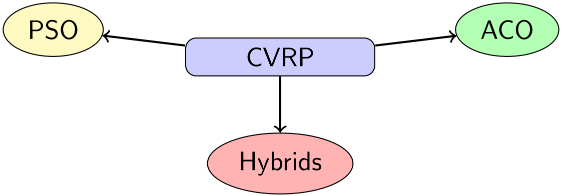


Figure: Major metaheuristic methodologies applied to CVRP

Cat and Mouse Optimization (CMBO) and Challenges for CVRP

What is CMBO?

- Population-based metaheuristic inspired by predator–prey dynamics.
- **Cats:** explore the search space by chasing solutions.
- **Mice:** exploit by escaping to safe havens (local refinement).
- Iterative process: update cats and mice $\rightarrow$ evaluate $\rightarrow$ best solution.

Challenges for CVRP:

- Originally designed for *continuous domains*.
- CVRP is *discrete* + *combinatorial* $\rightarrow$ route construction and vehicle assignment.
- Struggles with feasibility (capacity constraints) and avoiding sub-tours.

Improvements Required:

- Discrete solution representation (routes + vehicle loads).
- Feasibility-ensuring decoding mechanism.
- Problem-specific local search (e.g., 3-opt, route exchanges).
- Mechanisms to maintain diversity and escape local optima.

Adapted Cat and Mouse Based Optimizer

Population Initialization & Customer Clustering

Population Initialization

- Population size:

$$n_{population} = \max\left(50,\ \min(100,\ 2 \times n_{customers})\right)$$

- Each solution encoded as a vector:
  - **Customer Priorities** $\rightarrow$ order of service.
  - **Vehicle Reference Points** $\rightarrow$ coordinates guiding route assignment.
- Random uniform initialization ensures diverse starting solutions.

Customer Clustering

- Customers grouped with **K-Means**, number of clusters:

$$K = \lceil \sqrt{n} \rceil$$

- Reduces combinatorial complexity by focusing on sub-routes.
- **Dynamic update:** Clusters adjusted every 50 iterations to avoid stagnation.

Fitness Evaluation & Sorting

Fitness Function

- Measures quality of each solution through 2 parameters: total route distance for a solution, penalties for each unserved customer in a solution.

$$Fitness = Total\_distance + 10000 \times (\text{unserved customers})$$

- Ensures all customers are served within vehicle capacity.

Sorting into Roles

- Rank population by fitness (best $\rightarrow$ worst).
- **Mice (Top 50%)**: high-quality solutions, used for exploitation.
- **Cats (Bottom 50%)**: weaker solutions, explore new areas.

Fitness Sharing

- Adjusts fitness to penalize similar solutions within clusters.
- Maintains **diversity** of solutions.
- Prevents premature convergence to poor local optima.

F. Solution Update and Integration & G. Iteration Transition

Solution Update (Core Phase):

- **Update Cats:** Cluster-based refinement of subroutes.

$$s_{\text{new}}[l] = s_{\text{cat}}[l] + r \cdot (s_{\text{mouse}}[l] - l_{\text{rand}} \cdot s_{\text{cat}}[l]) \cdot \left(1 + \alpha \frac{t}{n_{\text{iter}}}\right)$$

  - $r \in [0, 1]$, $l_{\text{rand}} \in \{1, 2\}$, $\alpha = 0.5$
  - Random swaps & temperature-based acceptance $\Rightarrow$ maintain diversity

- **Solution Integration:** Aggregate cats from clusters $\Rightarrow$ unified population:

$$S = S_{\text{mice}} \cup S_{\text{cats}}$$

  Preserves cluster-optimized subroutes while retaining elite mice.

- **Update Mice:** High-quality solutions refined:

$$s_{\text{new}} = s_{\text{mouse}} + r \cdot (pop[l] - l_{\text{rand}} \cdot s_{\text{mouse}}) \cdot sign$$

  Accepted only if fitness improves; mutations (prob. p_m) avoid stagnation.

H. Local Search Operators

Variable Neighborhood Descent (VND)

- Refines routes using 6 operators.
- Ensures thorough neighborhood search
- Accepted only if distance reduces and capacity remains feasible.
- **Intra-route:**
  - Insert: relocate a customer.
  - Swap: exchange two customers.
  - Reverse: flip a segment.
  - 3-Opt: three-edge exchange.
- **Inter-route:**
  - Insert: move customer to another route.
  - Swap: exchange across routes.

Destroy-and-Repair

- **Destroy:** Remove up to $k = \max(5, \lfloor n/10 \rfloor)$ customers, guided by correlation:

$$R_{ij} = \frac{1}{(d_{ij}/\text{maxd}) + V_{ij} + 10^{-6}}$$

where d_{ij} = distance, maxd = max pairwise distance, $V_{ij} = 0$ (same route), 1 (different routes). Targets closely located customers across routes for disruption.

- **Repair:** Reinsert customers via cheapest insertion heuristic, ensuring feasibility. If infeasible, retain original; adjust vehicle count if needed.

Results and Analysis

Results: Evaluation Setup

- **Parameter Settings:**
  - Iterations: $n_{iterations} = 1500$, Restart limit: $n_{restart} = 100$, VND calls: $n_{vnd} = 40$.
  - Control parameters: $\alpha = 0.5$, $i_{restart} = 5$.
  - Problem-specific: Capacity, number of vehicles (m), and customers taken directly from CVRPLIB.
- **Implementation Environment:** Python 3.13 in VS Code, Windows 11 (64-bit), Intel Core i7-10510U @ 1.80 GHz, 8 GB RAM.
- **Evaluation Protocol:**
  - Each instance tested across 10 independent runs.
  - Statistical measures: *Best*, *Worst*, and *Average* objective function values.
  - Quality metric: Percentage Deviation (*Dev.%*) from Best-Known Solutions (BKS).
- **Comparison:** Compared ACMBO performance with the best solutions obtained by ACO and PSO over 10 runs across diverse CVRP instances, highlighting ACMBO's competitiveness against other state-of-the-art algorithms.

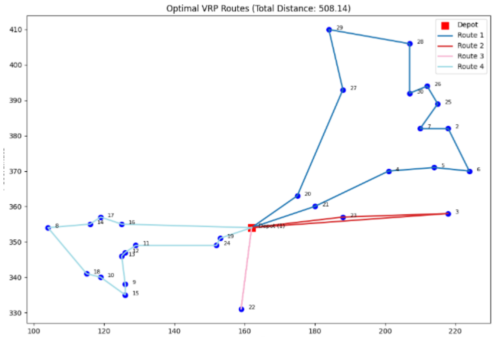


Figure: E-n30-k4 with 30 customers

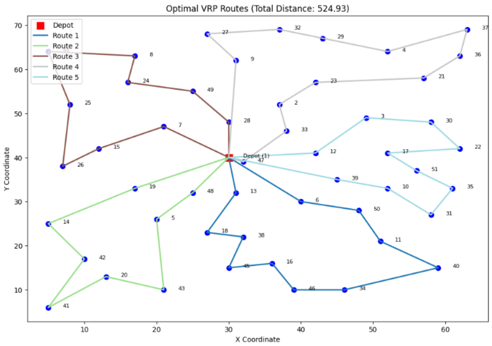


Figure: E-n51-k5 with 51 customers

In both these cases, ACMBO achieves equal or better solutions than the best-known solutions (BKS) from CVRPLIB, demonstrating that the algorithm performs very effectively for instances with uniformly distributed customers.

Results: Large-scale instances with Non-uniformly distributed Customers

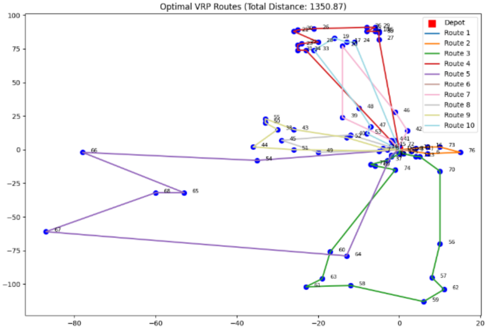


Figure: Tai75b with 75 customers

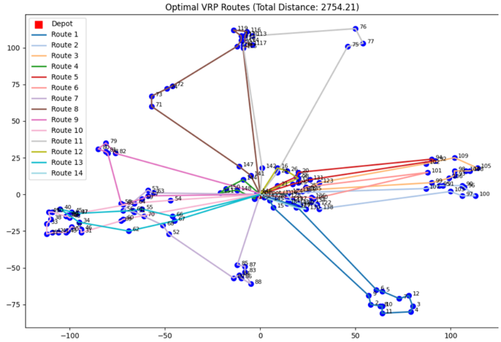


Figure: Tai150b with 150 customers

For large-scale instances with non-uniformly distributed customers, the graphs demonstrate that the ACMBO algorithm consistently achieves solutions within a 1% deviation from the best-known solutions.

Result Analysis of ACMBO

- Across 36 benchmark instances, ACMBO obtains the **optimal BKS** in 7 cases and achieves **improved solutions** in 2 cases with deviations of –2.49% and –4.87%.
- For **small-scale instances** ($n < 25$), deviations remain at 0.00%, indicating consistent recovery of benchmark values.
- For **medium-sized problems** (30–70 customers), deviations are mostly contained within 0.2–2.0%, with a few cases extending to 3.52%.
- For **large-scale instances** ($n \geq 100$), deviations range between 1.0% and 2.61%, confirming stable performance across increasing instance size.
- In comparative evaluation over 35 instances, ACMBO produces the **best or equal-best solutions** in 29 cases, corresponding to approximately 83% coverage.
- Relative to ACO, ACMBO consistently yields **lower objective values**, with notable gaps such as 92 units in B-n67-k10 and 442.77 units in Tai75a.
- Relative to PSO, ACMBO outperforms or matches in 32 cases; PSO records marginally better solutions in only 3 cases, with differences limited to 3–5 units.

Conclusion

Conclusion

- The ACMBO algorithm demonstrates strong effectiveness in solving the Capacitated Vehicle Routing Problem (CVRP), consistently delivering **high-quality and near-optimal solutions**.
- It successfully **achieves or surpasses best-known solutions (BKS)** in multiple benchmark instances, while maintaining **low deviations** across small, medium, and large problem sizes.
- In non-uniform clustered customer scenarios (Tai series), ACMBO outperforms other algorithms significantly, demonstrating its adaptability to irregular and challenging problem structures.
- Comparative analysis confirms ACMBO's **superior stability and accuracy** over established metaheuristics such as PSO and ACO, establishing its competitive advantage.
- The algorithm's adaptive mechanisms, customer segmentation, and local search strategies contribute to its **scalability and robustness**, making it effective across diverse benchmark sets.

Future Work

Future Work

- **Refinement for Large-Scale CVRP:**
  Enhance scalability and precision to further reduce deviations in very large customer instances, ensuring consistent near-optimal solutions.

- **Extension to VRP Variants:**
  Apply ACMBO to *electric, time-window, multi-depot, and heterogeneous vehicle routing problems* to evaluate adaptability under diverse and realistic logistical constraints.

- **Real-World Applications:**
  Test ACMBO in domains like *e-commerce delivery optimization* and *urban waste collection*, incorporating real-time data and dynamic constraints to validate operational utility.

REFERENCES

1. G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, Oct. 1959.
2. C. Contardo and R. Martinelli, "A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints," *Discrete Optimization*, vol. 12, pp. 129–146, May 2014.
3. M. Dehghani, Š. Hubálovský, and P. Trojovský, "Cat and mouse based optimizer: A new nature-inspired optimization algorithm," *Sensors*, vol. 21, no. 15, p. 5214, Jul. 2021.
4. T. J. Ai and V. Kachitvichyanukul, "A Particle Swarm Optimization for the Capacitated Vehicle Routing Problem," *Int. J. of Logistics and SCM Systems*, vol. 2, no. 1, pp. 50–55, 2007.
5. B. Bullnheimer, R. F. Hartl, and C. Strauss, "Applying the ant system to the vehicle routing problem," in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Springer, 1999, pp. 285–296.
6. Z. H. Ahmed et al., "An enhanced ant colony system algorithm based on subpaths for solving the capacitated vehicle routing problem," *Symmetry*, vol. 15, no. 11, p. 2020, Nov. 2023.
7. A. S. Alfa, S. S. Heragu, and M. Chen, "A 3-opt based simulated annealing algorithm for vehicle routing problems," *Computers & Industrial Engineering*, vol. 21, no. 1–4, pp. 635–639, 1991.
8. V. H. S. Pham, V. N. Nguyen, and N. T. N. Dang, "Hybrid whale optimization algorithm for enhanced routing of limited capacity vehicles in supply chain management," *Scientific Reports*, vol. 14, no. 1, p. 793, Jan. 2024.
9. L. Korayem, M. Khorsid, and S. S. Kassem, "Using grey wolf algorithm to solve the capacitated vehicle routing problem," in *IOP Conf. Series: Materials Science and Engineering*, vol. 83, no. 1, p. 012014, 2015.

Thank You