Optimizing Electric Vehicle Routing: A Statistical Analysis of ACO, GA, and SA Algorithms

Arushee Thakur
SRM Institute of Science and Technology
Chennai, India

Diya Giri
SRM Institute of Science and Technology
Chennai, India

Swarup Panda
SRM Institute of Science and Technology
Chennai, India

R. Usharani
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur, Chennai

Abstract—In the world of electric vehicle (EV) navigation, the quest for the best routing algorithms is crucial for creating efficient and sustainable transportation options. Conventional methods like Dijkstra's Algorithm show limitations in scalability and adaptability in changing conditions, which require frequent adjustments. This study supports using Ant-Colony Optimization (ACO), Genetic Algorithm (GA), and Simulated Annealing (SA) as heuristic techniques to optimize EV routes. A comprehensive assessment of several optimization algorithms for solving the Electric Vehicle Routing Problem (EVRP) provided valuable information on their efficiency and effectiveness. This research evaluates Ant-Colony Optimization (ACO), Genetic Algorithm (GA), and Simulated Annealing (SA) as possible approaches to optimize the Electric Vehicle Routing Problem (EVRP). ACO stands out as the top contender, showing better precision and dependability in providing the best routing options for various EVRP situations. In spite of the extended computational duration, ACO stands out in pinpointing the most optimal travel distances, making it a strong option for EVRP optimization. Simulated Annealing shows respectable performance despite some variability, coming in behind Genetic Algorithm in ranking for finding the shortest routes. ACO's effectiveness in addressing convergence problems and providing eco-friendly transportation solutions makes it the top choice for optimizing EV routing. The provided visual representation explains how ACO works by showing the most efficient routes between set map points, strategically connected to charging stations for user convenience.

Index Terms—**Nature Inspired Computing Techniques, Ant Colony Optimisation, Genetic Algorithms, Simulated Annealing**

I. INTRODUCTION

Electric vehicles (EVs) present a sustainable option compared to conventional fossil fuel cars, positioning them as a promising advancement in the automotive sector. However, worries about the limited distance that electric vehicles can travel, known as "range anxiety," still hinder their widespread use [1]. It is essential to tackle these issues to increase the use of electric vehicles in everyday transportation situations.

This article introduces a new strategy for dealing with EV range anxiety by applying optimization techniques such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Simulated Annealing (SA) algorithms. [2] This research acknowledges that these algorithms are efficient in navigating complex networks, which makes them suitable for addressing the Electric Vehicle Routing Problem (EVRP). The goal is to enhance EV routes by integrating these algorithms with existing traffic and charging station information in order to develop a strong solution that considers both energy consumption and arrival time. [3]

Based on ants' foraging behavior, the ACO algorithm optimizes paths by simulating ant communication with pheromones, providing a distinctive approach . ACO algorithms imitate behavior to find the best routes with charging stations for electric cars, reducing range anxiety and improving travel efficiency. Furthermore, as stated by Rao et al. in 2019, the Genetic Algorithm and Simulated Annealing approaches offer unique but compatible ways to improve route efficiency when dealing with the EVRP. The aim is to offer useful information on how efficient and practical these optimization algorithms are in reducing EV range anxiety by evaluating and comparing their performance. Researchers carry out empirical analysis and experiments to prove the efficiency of these algorithms in enhancing EV route planning, aiming to promote sustainable transportation and diminish environmental effects. [4]

II. RELATED WORK

A. Temporal Multi-Objective Ant Colony Optimization for EV Routing

Zhang et al.(2016) [5] proposed a novel temporal multiobjective ant colony optimization (ACO) algorithm tailored for electric vehicle (EV) routing, addressing various drivers' requirements under stochastic traffic conditions. Their algorithm optimizes route length, traveling time, energy consumption, battery recycling lifetime, and cabin temperature while meeting constraints on arrival time and state of charge. Through simulation experiments and comparative analysis with existing ACO methods, their study demonstrates superior convergence performance and solution distribution,

showcasing the effectiveness of their approach in generating optimized

B. Genetic Algorithm for Two-Echelon EV Routing Problem

Zhang et al(2020) [6] developed a mathematical model for the two-echelon electric vehicle routing problem (2E-EVRP) and designed a heuristic algorithm, specifically targeting the challenges of high cargo load and high timeliness distribution in the logistics industry. Their genetic algorithm addresses both the multiple depot vehicle routing problem (MDEVRP) and the split delivery vehicle routing problem (SDVRP), aiming to optimize total path length. Through experimentation with a logistics company in Beijing, they validate the efficiency of their algorithm, showcasing significant optimization gains compared to traditional methods such as simulated annealing.

C. Multi-Objective Simulated Annealing for PRT Vehicle Management

Chebbi and Siala [7] proposed a multi-objective simulated annealing (MOSA) algorithm for routing electric vehicles with limited battery capacity in Personal Rapid Transit (PRT) systems. Their study focuses on minimizing both total energy consumption and the number of vehicles used, employing a Pareto-dominant-based fitness strategy to accept new solutions. Through computational analyses on various instances, they demonstrate the effectiveness of MOSA in optimizing routing plans for EVs within PRT systems, highlighting its potential for addressing complex multi-objective optimization problems in transit management.

D. ACO for EV Routing with Recharging Stations

Mavrovouniotis et al. [8] extended ant colony optimization (ACO) to address the unique challenges of the Electric Vehicle Routing Problem (EVRP), considering factors such as uncertain energy levels and the need for recharging stations. Their ACO-EVRP approach incorporates a look-ahead strategy to ensure feasible routes for EV fleets, as demonstrated through simulations on benchmark problems. By leveraging ACO's ability to adapt to dynamic environments, their study provides a robust solution for optimizing EV routing while considering the availability of recharging infrastructure.

E. Genetic Algorithm for Autonomous EV Min-Max Routing

Fazeli et al. [1] tackle the optimization of routing strategies for fleets of autonomous electric vehicles (AEVs) with limited battery capacity and sparse charging infrastructure. Their proposed genetic algorithm-based meta-heuristic targets the min-max AEV routing problem, aiming to minimize the maximum distance traveled while considering charging station availability. Through extensive computational analyses and simulations, they showcase the efficacy of their approach in addressing range anxiety and optimizing routing strategies for AEVs, highlighting its potential to enhance the efficiency and sustainability of autonomous electric transportation systems.

III. ELECTRIC VEHICLE ROUTING PROBLEM

A fully connected weighted graph $G = (A \cup F, L)$ models an EVRP instance, where $A = \{1,...,n\}$ is a set of n customers (nodes), $F = \{n + 1,...,n + s\} \cup \{0\}$ is a set of s energy recharging stations including a central depot 0,

$$L = \{(a,b)|a,b \in A\}$$

is a set of links. Each link (a, b) has a non-negative value $t_{ab} = R^+$ representing travel time between customers a and b, defined as

$$t_{ab} = d_{ab}/s_{ab}, \tag{1}$$

The distance (d_{ab}) and average speed (s_{ab}) for each arc $(a,b) \in L$ are given in kilometers and kilometers per hour, respectively. Each customer $a \in A$ has a non-negative demand δ_a for goods to be delivered by m vehicles, with an associated service time σ_a. The service time σ_a is proportional to the demand, with higher demand requiring more service time. The vehicle load of an EV, k, on arc (a,b) is denoted by l_{ab}^k ($0 \leq l_{ab}^k \leq Q$), where Q is the maximum vehicle capacity. The maximum service time for each EV is J minutes, determining maximum working hours. Each recharging station $a \in F$ has a waiting time w_a for possible wait times, with a constant battery recharging rate r for all charging stations.

Each electric vehicle (EV) arriving at a customer or recharging station has a battery level (represented as e_a^k) that ranges from 0 to the maximum capacity (B_C). The charging time (c_b^k) at a station $(a \in F)$ depends on the current energy level and the charging rate, calculated as $(B_C - e_a^k)/r$ for a full charge. All EVs start the day fully charged and loaded at the depot.

The goal is to minimize total travel time by finding the smallest set of EVs that visits each customer once, starting and ending at the depot, satisfying their demands. A complete solution for EVRP involves a permutation of nodes (customers and recharging stations) and includes the routes for all EVs [9].

IV. PROPOSED METHODOLOGY

This paper discusses multiple approaches to address the electric vehicle (EV) routing issue, such as the GA, SA, and PSO algorithms, before advocating for the Ant Colony Optimization (ACO) algorithm.

A. Genetic algorithms

JH Holland developed genetic algorithms (GA) in Michigan, USA, in the 1960s, categorizing them as evolutionary algorithms. GA is based on evolution through natural selection. It consists of three components: selection, crossover, and mutation. The population of solutions is generated, with a selection of solutions based on fitness to become parents for offspring generation through crossover. Offspring undergo a mutation process for population diversification. With better solutions having a higher probability of being chosen for

crossover and mutation maintaining some parent features, new generations are assumed to provide improved results. [10] This iterative process is continued through generations to find the optimal solution.

TABLE I
BASIC GA PARAMETER VALUES

Attribute	Value
Mutation Rate	0.06
Population size	2000
Population generation	100

Algorithm 1 GA-VRP

1: pop ← InitializePopulation(popsize)
2: g ← 0
3: while (g ¡ maxgen and term not satisfied) do
4:     EvalFitness(pop)
5:     selected ← Selection(pop)
6:     offspring ← Crossover(selected)
7:     mutated ← Mutation(offspring)
8:     pop ← Replace(pop, mutated)
9:     g ← g + 1
10: end while
11: OUTPUT: Best solution found in the population

The fitness function in GAs for the VRP evaluates a solution's quality by minimizing total distance, balancing workload, and meeting capacity constraints. Fitness is determined by the total distance traveled per vehicle, the customer visit sequence considered. [11]

$$Fitness = (\alpha * TotalDist) + (\beta * WorkImbalance) + (\gamma * CapViolation)$$

where α, β, γ are weight coefficients. Total distance is the sum of distances between visited customers. Workload imbalance is measured by workload standard deviation or max difference. Capacity violation is demand exceeding vehicle capacity.

B. Simulated annealing

Kirkpatrick et al. devised the Simulated Annealing (SA) method, which is a meta-heuristic approach inspired by solids' annealing process. This method includes heating a material and then gently cooling it to generate a strong crystalline structure. In optimisation, the feasible solution is the system, with the energy state being the target function. [12] The optimum solution is linked to the lowest energy state. SA can avoid local optima, which slows convergence, and it is a memory-less

method that does not require information from the search process. In each iteration, the algorithm generates a random neighbour and accepts the neighbour solution if it improves the objective function. Otherwise, it will use the metropolis criteria to determine whether to accept the answer. [13] This criterion employs a Boltzmann distribution-based acceptance probability that is determined by the goal value's current temperature (T0) and energy degradation (DE). [14]

TABLE II
SA PARAMETERS VALUES

Attribute	Value
Initial temperature	1000
Cooling rate	0.003

Algorithm 2 SA-VRP
1: Initialize T
2: Initialize π
3: while (term condition not satisfied) do
4:     for i from 1 to max iter per T do
5:         Perturb π to obtain π new
6:         Calculate $\Delta f = f(\pi new) - f(\pi)$
7:             if (Δf ¡ 0) or (exp(-Δf / T) ¿ rand(0, 1)) then
8:                 Accept $\pi \leftarrow \pi$ new
9:         end if
10:     end for
11:     Update T: T ← cool(T)
12: end while
13: OUTPUT: Best solution π

The acceptance probability is calculated as, P

$$= \exp((F(curr) - F(new))T,$$

where T represents the temperature, allowing worse solutions to be accepted more frequently at higher temperatures and less as the temperature decreases.

C. Particle Swarm Optimisation

PSO was introduced by Kennedy and Eberhart in 1995 as a population-based metaheuristic inspired by the behavior of organisms like birds and fish searching for food. Each particle in the population has its own position and speed, sharing information with others to adjust accordingly. [15] Optimization is achieved through adjustments based on individual (Pbest) and global best (Gbest) positions. The position and speed of each particle is updated in the process.

Algorithm 3 PSO-VRP

1: Init particles P
2: Init global best solution gbest
3: while (term not satisfied) do
4:     for each particle p in P do
5:         Update vel and pos:
6:             $v_p = w * v_p + c1 * rand() * (pbest_p - x_p)$
                $+ c2 * rand() * (gbest - x_p)$
7:             $x_p = x_p + v_p$
8:         Evaluate fitness
9:         if (fitness of p ¡ fitness of gbest) then
10:            Update gbest: gbest = p
11:        end if
12: end while
13: OUTPUT: Best solution π

D. Ant Colony Optimisation

The ACO algorithm is utilized in the proposed methodology to solve the EV routing problem, with specified objective functions, constraints, and parameters. The ACO technique is inspired by the foraging behavior of ant colonies, introduced by Marco Dorigo in the 1990s. Ants prefer community survival over individual species, communicating through sound, touch, and pheromones. Pheromones are chemical compounds emitted by ants to trigger a social response within the same species. The ACO algorithm efficiently finds near-optimal solutions in complex problems at a low
computational cost. [16]

Algorithm 4.1 ACO-EVRP

1: Set t = 0
2: InitPheromTrails(τ_0)
3: while (term not satisfied) do
4:     ConstructSol
5:     Find π_{ab} (Best solution for current iteration)
6:         If(Objective value of π_{ab} < Objective value of π_{bs})
7:             Update π_{bs} with π_{ab}
8:     EndIf
9:     PheromUpdate
10:        Increment t: t = t + 1
11: EndWhile
12: OUTPUT: π_{bs}

Defining the objective function, constraints, and parameters is essential when applying the ACO algorithm to the EV routing problem. The objective is to decrease travel time and energy consumption, influenced by various factors. Constraints guarantee the EV reaches its destination with charge remaining and meets user preferences. Parameters include particle

number, iteration limit, and exploration/exploitation coefficients.

TABLE III
BASIC ACO PARAMETER VALUES

Parameter	Value
alpha	−0.2
beta	9.6
pheromone resistance	0.3
initial pheromone	0.8
number of ants	2048

The configuration sets up essential global variables for simulating an EVRP by defining parameters like 'shortest path', 'shortest distance', 'cars', 'chargers', 'distances', 'pheromones', and 'ants', along with associated values like 'num ants', 'dst power', 'pheromone _power', 'evaporation rate', and 'pheromone intensity'. It also loads images for cars and chargers, specifies screen size, and sets frame rate. The function 'load image rgb' is described for BGR to RGB image conversion. 'spawn ants' generates ant agents ensuring the presence of chargers and cars, initializes charger slots, and assigns attributes for ants like 'distanceTraveled', 'carsVisited', 'chargerSlots', 'location', and 'nodesVisited', adding them to the ant list for simulation.

Algorithm 4.2 SpawnAnts

1: Function spawn ants
2: If |C| > 0 and |V | > 0
3:     S ← {i : 2 for i in range(|C|)}
4:     For _ in range(N)
5:         l← {'t' : $c',' i'$: random integer between 0 and |V |}
6:         $a ← \{'d' : 0.0,' v' : \{\},' s' : S$.copy(),
            $'l' : 1,' n' : [l.copy()]\}$
7:     Add a to A
8:     EndFor
9: EndIf
10: EndFunction

The ResetShortestPath function is a crucial part of the Electric Vehicle Routing Problem (EVRP) algorithm as it initializes important variables for the simulation to progress. Setting 'shortest_path' to None clears any previous shortest path information, ensuring a fresh start for the algorithm. [17] Additionally, initializing

'shortest_distance' to infinity provides a reference point for newly calculated distances. This proactive approach prepares the EVRP algorithm for success by offering a clean slate for optimal routing solutions to be developed.

In the following SaveShortestPath function, the algorithm utilizes an iterative method to enhance the shortest path calculation during the simulation. By looping through each ant in the environment, the function evaluates their traveled distance and compares it with the current shortest distance. If an ant's path is shorter, shortest_path is updated with their visited nodes and shortest_distance is refreshed with the new minimal distance. This iterative approach helps the algorithm adjust to changing route conditions, refining its understanding of efficient paths. The SaveShortestPath function is crucial in guiding the EVRP algorithm towards an optimal solution, supporting efficient route planning in complex logistics.

Algorithm 4.3 CalculateProbabilities

1: Functioncalculate probabilitesD,P

2:    $D \leftarrow EnsureNonZero(D)$

3:    $des \leftarrow \left(\frac{1}{D}\right)^d \times P^p$

4:    return Normalize(des) 5:
EndFunction

In addition to technical aspects, the suggested approach will take into account user experience. The goal is to develop a user-friendly and intuitive interface for the navigation application, enabling users to input their destination, choose charging stations, and modify their route in real-time easily. Through a blend of advanced optimization strategies and a user-focused approach, a navigation solution can be developed to address EV range anxiety and improve the driving experience. [1]

V. RESULTS AND DISCUSSIONS

The assessment of multiple optimization algorithms for addressing the Electric Vehicle Routing Problem (EVRP) provided valuable insights into their performance and effectiveness. Ant Colony Optimization (ACO) stood out as the top performer in terms of identifying the most efficient travel distance. [10] Despite its longer execution time compared to other algorithms, ACO excelled in finding the shortest route and maintained competitive performance. Its exceptional accuracy in solving optimal distance between cities makes it a strong choice for EVRP optimization. Similarly, ACO's reliable performance even under convergence conditions highlights its effectiveness in solving routing problems with different levels of complexity.

TABLE IV
ACO ALGORITHM

Map Name	Nodes	Time(s)	Distance	Accuracy
Map A	127	29	124651.524	0.95435
Map B	52	5	721.098	0.95724
Map C	535	64	1510.321	0.97425

TABLE V
GA ALGORITHM

Map Name	Nodes	Time(s)	Distance	Accuracy
Map A	127	4	419224.465	0.95435
Map B	52	7	11551.312	0.95724
Map C	535	28	9466.576	0.97425

TABLE VI
SA ALGORITHM

Map Name	Nodes	Time(s)	Distance	Accuracy
Map A	127	0.3	265289.381	0.95435
Map B	52	0.9	10768.876	0.95724
Map C	535	16	6471.753	0.97425

Although Genetic Algorithm (GA) ranked second in finding the shortest route, it had slower execution times, making it less ideal for real-time EVRP optimization. [18] Simulated Annealing (SA) exhibited impressive performance in terms of execution times and convergence behavior, but struggled to consistently find the shortest distance, especially with increasing nodes. This indicates that SA may need more iterations to converge, limiting its practical use in time-sensitive EVRP scenarios. [14]

ACO's superior performance in determining the most efficient travel distance and its ability to overcome convergence challenges make it the top choice for EVRP optimization. [19] By utilizing pheromone trails and effectively exploring solution space, ACO delivers unparalleled accuracy and reliability in providing optimal routing solutions for various levels
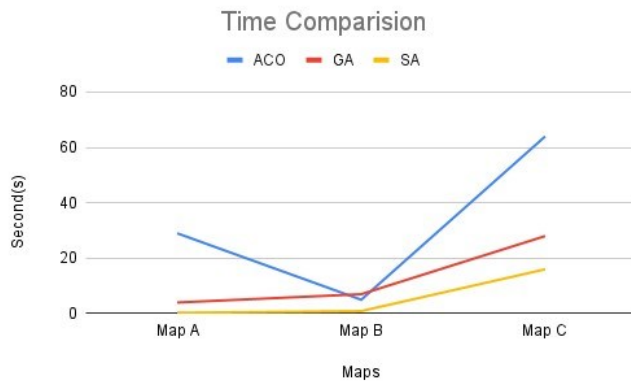
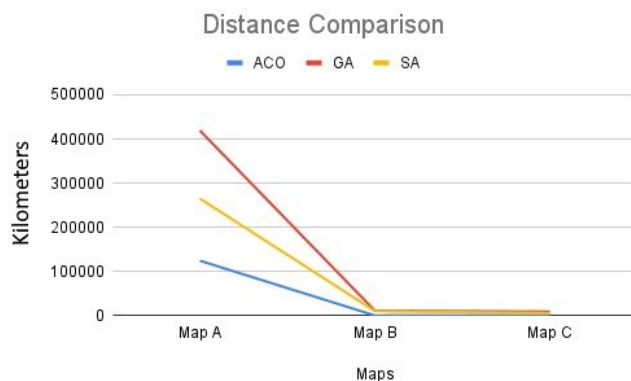Fig. 1. Evolution of GA, ACO, and SA findings over time(s) for Maps A, B, and C

Fig. 2. Evolution of GA, ACO, and SA results across distance (km) for Maps A, B, and C.

of complexity in EVRP instances. The accompanying graph demonstrates how ACO algorithms function, showing the closest and most optimized route between two map coordinates. The red lines represent the optimized path for the best route, while the black lines display all possible routes. The red lines also feature the highest number of charging stations along the optimized path. [16]
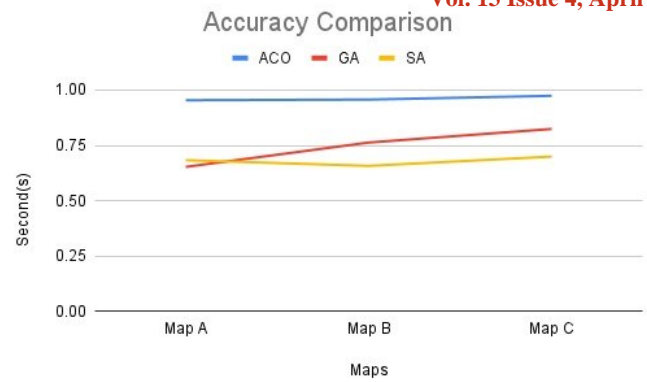

Fig. 3. Evolution of GA, ACO, and SA accuracy over time (seconds) for Map A, Map B, and Map C.

VI. CONCLUSION

The user will input their desired destination location. Given the car's current position, we can determine the general direction it should head towards. Initialisation particles for ACO will be based on routes in this direction. The car's current charge level will be requested to decide if a charging stop is needed, impacting the cost function. Particles will be adjusted based on actual distance and turns taken. Each particle's cost will consider total distance, proximity to charging stations, and preference for straight roads. Once the most optimal path cost stabilizes, it will be manually checked for user destination and charging station proximity. If checks fail, a new seed solution will be used to generate alternative paths, reducing compute time for mobile and on-car devices.

The final path will be displayed on Google Maps for user convenience.

The results offer helpful guidance for professionals and scholars in logistics and optimization, aiding in choosing suitable algorithms for tackling intricate routing issues in electric vehicle operations. Future studies could investigate combining methods and improving algorithms to boost ACO's efficiency and scalability in solving extensive EVRP cases, advancing sustainable transportation planning and management.

REFERENCES

[1] S. S. Fazeli, S. Venkatachalam, and J. M. Smereka, "Efficient algorithms for autonomous electric vehicles' min-max routing problem," arXiv preprint arXiv:2008.03333, 2020.

[2] F. Semet, P. Toth, and D. Vigo, "Chapter 2: Classical exact algorithms for the capacitated vehicle routing problem," in Vehicle Routing: Problems, Methods, and Applications, Second Edition. SIAM, 2014, pp. 37–57.

[3] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," European journal of operational research, vol. 59, no. 3, pp. 345–358, 1992.

[4] B. L. Golden, S. Raghavan, and E. A. Wasil, The vehicle routing problem: latest advances and new challenges. Springer Science & Business Media, 2008, vol. 43.

[5] S. Zhang, Y. Luo, and K. Li, "Multi-objective route search for electric vehicles using ant colony optimization," in 2016 American Control Conference (ACC). IEEE, 2016, pp. 637–642.

[6] Y. Zhang, S. Zhou, X. Ji, B. Chen, H. Liu, Y. Xiao, and W. Chang, "The mathematical model and an genetic algorithm for the two-echelon electric vehicle routing problem," in Journal of Physics: Conference Series, vol. 1813, no. 1. IOP Publishing, 2021, p. 012006.

[7] O. Chebbi and J. Chaouachi, "Multiple-objective simulated annealing optimization approach for vehicle management in personal rapid transit systems," in Telematics-Support for Transport: 14th International Conference on Transport Systems Telematics, TST 2014, Katowice/Krakow/Ustro´ n, Poland, October 22-25, 2014. Selected Papers 14´ . Springer, 2014, pp. 284–293.

[8] M. Mavrovouniotis, G. Ellinas, and M. Polycarpou, "Ant colony optimization for the electric vehicle routing problem," in 2018 IEEE Symposium series on computational intelligence (SSCI). IEEE, 2018, pp. 1234–1241.

[9] S. R. Ait Haddadene, N. Labadie, and C. Prodhon, "Bicriteria vehicle routing problem with preferences and timing constraints in home health care services," Algorithms, vol. 12, no. 8, p. 152, 2019.

[10] J. Long, Z. Sun, P. M. Pardalos, Y. Hong, S. Zhang, and C. Li, "A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem," Information Sciences, vol. 478, pp. 40–61, 2019.

[11] Y. Tsujimura and M. Gen, "Entropy-based genetic algorithm for solving tsp," in 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111), vol. 2. IEEE, 1998, pp. 285–290.

[12] E. Alba and B. Dorronsoro, "Solving the vehicle routing problem by using cellular genetic algorithms," in European Conference on Evolutionary Computation in Combinatorial Optimization. Springer, 2004, pp. 11–20.

[13] F. Y. Vincent, H. Susanto, P. Jodiawan, T.-W. Ho, S.-W. Lin, and Y.T. Huang, "A simulated annealing algorithm for the vehicle routing problem with parcel lockers," IEEE Access, vol. 10, pp. 20764–20782, 2022.

[14] C. Wang, F. Zhao, D. Mu, and J. W. Sutherland, "Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows," in Advances in Production Management Systems. Sustainable Production and Service Supply Chains: IFIP WG 5.7 International Conference, APMS 2013, State College, PA, USA, September 9-12, 2013, Proceedings, Part II. Springer, 2013, pp. 170–177.

[15] Y. Marinakis, M. Marinaki, and A. Migdalas, "Particle swarm optimization for the vehicle routing problem: a survey and a comparative analysis," in Conference LOT 2014: Logistics, optimization and transportation 01/09/2014-02/09/2014. Springer, 2018, pp. 1163–1196.

[16] H. I. Calvete, C. Gale, and M.-J. Oliveros, "Ant colony optimization´ for solving the vehicle routing problem with delivery preferences," in Modeling and Simulation in Engineering, Economics and Management: International Conference, MS 2012, New Rochelle, NY, USA, May 30June 1, 2012. Proceedings. Springer, 2012, pp. 230–239.

[17] G. S. Er and V. Dhir, "Open vehicle routing problem by ant colony optimization," International Journal of Advanced Computer Science and Applications, vol. 5, no. 3, 2014.

[18] D. Gong and X. Ruan, "A hybrid approach of ga and aco for tsp," in Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788), vol. 3. IEEE, 2004, pp. 2068–2072.

[19] M. Dorigo and T. Stutzle, "Aco algorithms for the traveling salesman¨ problem," Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, 1999.