ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 03 March 2024

Green Vehicle Routing Model Using Flamingo Search Algorithm (FSA)

[1] Kathleen Iza Monzales, [2] Jay Tejada, [3] Christine Pena

[1][2][3] Department of Computer, Information Sciences, and Mathematics, University of San Carlos, Cebu City, Philippines.
E-mail: [1] 20100869@usc.edu.ph, [2] 20103300@usc.edu.ph, [3] cfpena@usc.edu.ph

Abstract— ***This paper presents a unique approach to the Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges (GVRP-MTPR) using the Flamingo Search Algorithm (FSA). GVRP-MTPR is a variation of the traditional Vehicle Routing Problem (VRP) characterized by the need for efficient routes considering the use of electric vehicles, multiple charging technologies, and partial recharges of the green vehicles. FSA is a swarm intelligence optimization algorithm that is inspired by the behaviors of flamingos, mainly the Foraging and Migrating behaviors. FSA has previously demonstrated excellent performance in a diverse set of tasks such as push-pull circuit problems, path planning problems, and network intrusion detection systems. In our proposed methodology, we use the Flamingo Search Algorithm (FSA) to tackle GVRP-MTPR. Our proposed model generates an initial set of solutions that is further optimized using the foraging and migrating behaviors of FSA. The model was tested on a dataset of 60 instances of varying customer and vehicle counts, order distributions, and topologies. Key metrics such as cost, fitness, number of iterations, and execution time are used in evaluating the performance of the model. The results highlight the competitiveness of the Flamingo Search Algorithm in addressing GVRP-MTPR, offering insights for the optimization of green vehicle routing in logistics and transportation operations.***

Index Terms— ***Flamingo Search Algorithm, Foraging Behavior, Green Vehicle Routing Problem with Multiple Technologies and Partial Recharges, Migrating Behavior***

I. INTRODUCTION

The vehicle routing problem (VRP) is a complex optimization challenge that arises in various industries requiring efficient transportation operations. It involves determining optimal routes and schedules for a fleet of vehicles to deliver goods or services to a set of geographically dispersed locations while minimizing costs and maximizing operational efficiency. As businesses increasingly face demands for faster delivery, reduced costs, and minimized environmental impact, the need for advanced VRP solutions becomes evident.

The traditional VRP poses numerous challenges due to its inherent complexity, which exponentially grows with the number of customers, vehicles, and constraints. Factors such as varying customer demands, time windows, vehicle capacities, and multiple depots further complicate the optimization process. Failing to address these challenges adequately can lead to inefficient route planning, increased transportation costs, excessive fuel consumption, and extended delivery times, negatively impacting a company's competitiveness and profitability. Thus, there is a pressing need to develop novel VRP algorithms and techniques that can effectively address these challenges.

The search for ever more efficient algorithms to implement vehicle routing models is constant and in recent years, nature-inspired algorithms have gained significant attention as effective tools for solving complex optimization problems. One such algorithm is the Flamingo Search Algorithm (FSA), a metaheuristic algorithm inspired by the unique characteristics and behavior of flamingos. Developed and published in 2021, FSA shows promise in addressing various optimization challenges. It is shown to have the best results in single peak and multipeak tests among Swarm Intelligence Algorithms from Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Grey Wolf Optimization (GWO), and Tunicate Swarm Algorithm(TSA). It is also shown to have a high accuracy rate with a fast convergence speed in the aforementioned tests. FSA has been experimented to solve several simulation experiments: push-pull circuit problem, path planning problem, and network intrusion detection system and were proven to be useful for the said problems. In this research, FSA will be applied on the Vehicle Routing Problem and will be examined for its applicability on this problem.

Through this research, the researchers will contribute to the development of efficient logistics systems for electric vehicle deliveries by developing a vehicle routing model that uses the Flamingo Search Algorithm (FSA) to optimize their routes.

II. REVIEW OF RELATED LITERATURE

A. Vehicle Routing Problem

Vehicle Routing Problem(VRP) is a combinatorial optimization for generating an optimal set of routes for a fleet of vehicles [5]. It was later improvised by Clarke and Wright in 1964 using the Savings Algorithm, an effective greedy algorithm. VRP offers a wide range of direct applications in business. VRP routing tool vendors frequently assert that their products may save customers between 5% - 30%. VRP is concerned with the service of a delivery company, from one or more depots to a set of customers through a set of vehicles with designated drivers moving on a given road network, such that all the customers' demands are met while the global transportation cost (monetary, distance, fuel usage, etc.) is minimized. Graphs can be used to describe the road network used, where the arcs are the roads, and the vertices are the nodes (depot and customers). Each arc has its own designated cost, which is usually the length or travel time [1]. Another variant of the Vehicle Routing Problem that recently emerged is the Green Vehicle Routing Problem that involved the use of Electric Vehicles (EV) instead of the manual vehicles. The use of electronic vehicles is a promising solution for reducing costs and pollution caused by transportation and mobility operations.

B. Swarm Intelligence

Swarm Intelligence (SI) was first introduced by Gerardo Beni and Jing Wang in 1989 as the collective behavior of decentralized, self-organized systems, natural or artificial. SI systems typically consist of a population of boids or simple agents that interact with one another and their surroundings on a local level. The inspiration often comes from nature, especially biological systems [10]. Swarm Intelligence Optimization is described as an important paradigm itself with a wide range of applications. Most applications are in the industry of engineering, medicine, and business, whether to minimize cost and energy consumption, or to maximize profit, output, performance, and efficiency [9].

C. Flamingo Search Algorithm

Flamingo Search Algorithm (FSA) was presented by Zhiheng and Jianhua in 2021. In their paper, they presented a new swarm intelligence optimization algorithm inspired by the two distinctive behaviors of flamingos: migratory and foraging. The proposed algorithm was built with global exploration and local exploitation capabilities for the optimization of algorithms with the use of mathematical models [18].

The main optimization ideas of FSA model are as follows: Flamingos communicate with each other.

The population of the flamingos are unaware of the where the abundance of food is located in the current search area, they update the location of each flamingo based on the knowledge shared with each other.

The rules of updating each flamingo are based on the behavior of the flamingo.

The two main behavior of the FSA model are as follows:

1. Foraging Behavior

a. Communicative Behavior

The flamingos that have the abundance of food in the population call the other flamingos to spread their location information and influence the position changes of other flamingos in the population.

b. Beak Scanning Behavior

If the current area of the flamingo is abundant in food, then the flamingo is encouraged to scan the area more carefully.

c. Bipedal Mobile Behavior

When the flamingos forage, while scanning the area with their beaks, their claws move toward where the food is most abundant in the flamingo population

2. Migrating Behavior

When the food is scarce in the present foraging area, the flamingo population migrates towards the next area where food is more abundant.

D. Other Approaches

In the paper, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges" by Felipe et al. (2014), presented the different optimization techniques to solve a variant of vehicle routing problem. The two search methods used in this paper are Local Search and Simulated Annealing.

1. Local Search

Deterministic Local Search is based on an iterative improvement of the solutions, usually defined through neighborhood structures. This paper utilized 4 local search operators as follows:

a. Recharge Relocation

If a feasible solution's route visits at least one recharge station, it can be improved by ideally relocating a single recharging station. Following this logic, the operator Recharge Relocation is designed to locate, if necessary (and if possible), the ideal location of a single recharge point along a route, without changing the sequence of visits to the consumers.

b. 2-opt

The 2-opt algorithm works as follows: take 2 arcs from the route, reconnect these arcs with each other and calculate new travel distance.

c. Reinsertion

Customers are relocated from one route to another with the purpose of eliminating routes, eliminating recharges or just decreasing energy consumption.

d. Local Search Combinations

combination of local search operators

2. Simulated Annealing

The Simulated Annealing Framework utilizes a modified version of reinsertion as the basic move of that metaheuristic. The customer to be removed is selected at random with the probabilities proportional to the savings produced by the removal. The route in which it is reinserted.

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 11 Issue 03 March 2024

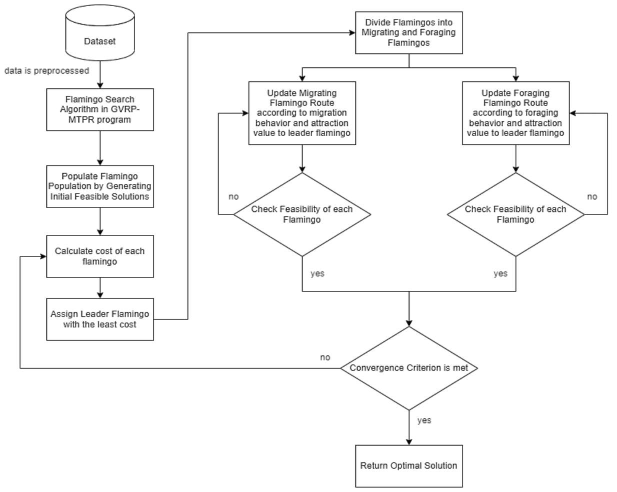

Fig. 1. Flowchart of Flamingo Search Algorithm in GVRP

is selected at random with uniform probability, to introduce diversification, and the optimal insertion point is computed.

Table I. Average Cost and Running Time of Algorithms

	Average Cost				Average Time			
	2O	RR	R_1^1	48A	2O	RR	R_1^1	48A
N=100	88.03	92.34	66.42	64.96	<1	<1	2	57
N=200	155.41	161.72	114.65	111.48	<1	<1	15	348
N=400	279.84	287.62	208.31	203.53	<1	<1	91	198

Table I shows the result of the comparison between algorithms 2-opt (2O), Recharge Relocation (RR), 1,1 Reinsertion (R_1^1) and 48A algorithm (48A) with respect to their average cost and average time. It shows that the 1,1 Reinsertion algorithm has the lowest average cost per customer while both O2 and RR have the lowest average time.

III. DESIGN AND METHODOLOGY

A. Dataset

The dataset used in the proposed research was taken from the paper by Felipe et al. [8], "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges". Each dataset contains the following technical data, as follows:

Number of customers (N): 100, 200, 400

Number of vehicles available: N/4

Battery capacity: 20 KWh (equivalent to an autonomy of 160 km)

Energy consumption: 0.125 KWh/km

Average speed: 25 km/h

Vehicle capacity: 2300 kg

Maximum route duration: 8 h

Service time (minutes)

Available technologies:

Slow (S): 3.600 KWh/h and cost 0.160 €/KWh (conventional household technology, only used at the depot for night recharge)

Medium (M): 20.000 KWh/h and cost 0.176 €/KWh (following the CHAdeMO protocol, available at some recharge stations)

Fast (F): 45.000 KWh/h and cost 0.192 €/KWh (following a wireless protocol, available at some recharge stations)

Fixed cost of recharge: 2.270 €/cycle

The dataset considers two configurations: (1) The depot is centrally located and there are nine recharge stations including the depot, (2) and the depot is located at a corner with 5 recharge stations including the depot. There are three (3) sets of twenty (20) instances per set containing 100, 200, and 400 customer nodes, respectively. The dataset has a total of 60 instances. All these instances are available at Doble TSP Multiples Pilas (ucm.es).

Table II. Dataset Composition

	Number of Instances	Number of Customers in each instance
N100	20	100
N200	20	200
N400	20	400

B. Algorithm Design

The general overview of the flow of the system, shown in Fig. 1 is detailed as follows:

1. Dataset are preprocessed and populated in the data structures of the model

The dataset instance will be preprocessed into the data structures of the model for it to be usable in the program.

2. Initialize flamingos by generating initial feasible solutions

A flamingo population was initialized by generating feasible solutions. Each flamingo will represent one solution(route) in the population.

3. Calculate the cost of each flamingo

For each flamingo in the population, the total amount of money spent for the solution was calculated, by taking into consideration the technology used for the recharging of the battery of the vehicle in order for the vehicle to successfully go through the whole route.

4. Assign the flamingo with the least cost as the Leader Flamingo

Find the flamingo with the least amount of cost in the population and assign it as the Leader Flamingo.

5. Divide the flamingos into 2 groups: foraging and migration

Divide the flamingo in the population into two groups based on the distinct behaviors of flamingos.

6. Update the flamingos based on the behavior of their group, and attraction value to the Leader Flamingo.

Each of the flamingos(solution) are updated according to the behavior they belong to.

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 11 Issue 03 March 2024

Operators to be used for the Foraging Behavior:

a. *Balance Vehicle Routes*: Given a flamingo, find the shortest and longest route and balance them by moving a set of nodes from the longest route to the shortest where their new size differences should not be larger than 1.
b. *Balance Charging Stations*: For each vehicle, find the shortest and longest sub-route, then move the Charging Station from the shortest sub-route to the middle of the longest sub-route.

Operators to be used for the Migratory Behavior:

a. *Route Rebuild:* The routes of the current flamingo's vehicles are compared to the routes of the vehicles in the leader flamingo. The common sub-routes between the current flamingo and best flamingo are retained while the remaining nodes are redistributed throughout the vehicles of the current flamingo.

7. Check the feasibility of each of the updated flamingos.

Check whether the newly updated flamingos exceed the restrictions of the problem: Vehicle Capacity and Maximum Route Duration. If the feasibility is violated, go back to 6, if not, go to 8.

8. Check if the convergence criterions is met

Check whether all the flamingos have passed the threshold value of the convergence criterion. Refer to Section 4.8.3 for the calculation of the convergence criterion. If all them have passed, proceed to 9, if not, go back to 3.

9. Return Most Optimal Solution

Once all the flamingos have converged into a single point in the solution space, the Leader Flamingo of the last iteration is returned to the user.

C. Verification, Validation, and Testing

The output of the model was the most optimal route from the given instance of the dataset. The verification and validation of the model was measured by these four metrics: Cost, Fitness, Number of Iterations, and CPU time of the optimal solution. Each of the metrics were calculated as follows.

a. Cost

The cost was calculated by getting the total amount of money spent for the recharge of the vehicle for the entire route. We get D_j^i which is the distance between nodes i and j in km. Given the coordinates of each nodes in the route, we calculate the distance between nodes by using the Euclidean Distance formula, refer to (1) where x_i *and* y_i and x_j *and* y_j are x and y coordinates of node i and j respectively.

$$D_j^i = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \qquad (1)$$

The proponents then computed for the energy consumption of the travel, refer to (2) given that E is the energy consumption (0.125 KWh/km) and E_j^i is the energy consumption between nodes i and j

$$E_j^i = \frac{D_j^i}{E} \qquad (2)$$

The proponents then got the total cost spent for every recharge station in the route, including the fixed cost of recharge, refer to (3). Given that C_t is the cost of recharge using technology t, and FC is the fixed cost of recharge (2.270 €/cycle). TC_i is the total cost (in euro) spent in recharge station i.

$$TC = \sum_{i=0}^{r} (TC_i + FC) \qquad (3)$$

b. Fitness

The proponents then got the total cost spent for every recharge station in the route, including the fixed cost of recharge. Given that D_j^i is the distance between nodes i and j in km, A_s is the average speed (25km/h), and n is the total number of nodes in the route. The proponents got the time spent traveling between nodes i and j, refer to (4).

$$T_j^i = \frac{D_j^i}{A_s} \qquad (4)$$

The proponents then got the total time spent for the recharge stations, where r is the total number of recharge stations in a route, E is the total energy consumption of a trip, T_{it} is the time spent for the recharge of recharge station i using technology t, and R_t is the energy rate of recharge station using technology t. The proponents then got the summation of the time spent between all nodes and their service time present in the entire route, refer to (5). Given that S_i is the service time of node i.

$$F = \sum_{x=0}^{n} [(T_j^i)x + S_i] + \sum_{x=1}^{r} T_{it} \qquad (5)$$

c. Number of Iterations

The convergence criterion refers to the condition that determines when the algorithm has converged to an acceptable solution or when it should terminate. In this research, the proponents based off the definition of the convergence criterion by Querin, et al. from the book "Discrete Method of Structural Optimization" where i is the current iteration number(>10), OF_i is the objective function value(cost) in the ith iteration, and $\mathcal{E}_i$ is the convergence value of the objective function in the ith iteration.

$$\varepsilon_i = \frac{\left|\sum_{i-9}^{i-5} OF_i - \sum_{i-4}^{i} OF_i\right|}{\sum_{i-4}^{i} OF_i} \qquad (6)$$

The convergence value was then compared to the threshold value of 85% for maximum precision. If the convergence value was below the threshold value, the program continues to optimize the solution. If the convergence value was equal or greater than the threshold value, convergence criterion was met, hence the program terminates, and the best solution

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 03 March 2024

becomes the most optimal solution.

d. CPU Execution Time

CPU time refers to the number of seconds spent to execute the entire program in a specific instance of the dataset. CPU time was provided by the IDE used by the proponents when the program terminates.

IV. RESULTS AND ANALYSIS

This chapter illustrates the results of the different processes taken in order to achieve the goals set forth by the researchers at the beginning of this study. Furthermore, it further expounds on the different methodologies and environments used in obtaining such results.

a. Preliminary Results

Table 3 presents the results of the FSA runs enumerating the metrics previously elaborated, grouped and averaged according to the customer node count of the instances. It shows the average metrics of the instances when grouped by their customer node count. It shows that cost grows exponentially by a factor of 2 as the customer node count increases by a factor of 2. Average fitness follows the same trend where a doubling of customer node size corresponds to a rough doubling of fitness value. The same trend applies to average iterations. The average CPU times on the other hand, showed a fivefold increase when the customer node count was doubled from 100 to 200, then it showed an increase of a factor of 10 when the customer node count increased from 200 to 400.

Table III. Average Metric Cost

n	Ave. Cost (Euros)	Ave. Fitness (Minutes)	Ave. Iterations	Ave. CPU Time (Seconds)
100	57.07682	1646.0775	13.4	0.9259
200	114.15895	3228.369	24.5	5.82865
400	228.3429	6543.039	46	51.759

b. Cost

Table IV compares the result of cost for FSA to other contemporary approaches which were included in Felipe's study. It shows that FSA produced the most optimal solution cost-wise on instances with 100 customer nodes. It also shows that FSA produced the 2nd most optimal solution at 200 nodes. Lastly, it shows that at 400 nodes, FSA produced the 3rd most optimal solution.

Table IV. Ave. Cost Comparison Between FSA and Other Algorithms

N	FSA	2O	RR	R_1^1	48A
100	**57.08** **0.00%**	88.03 +54.22%	92.34 +61.77%	66.42 +16.36%	64.96 +13.81%
200	114.16 +2.40%	155.41 +39.41%	161.72 +45.07%	114.65 +2.84%	**111.48** **0.00%**
400	228.34 +12.19%	279.84 +37.49%	287.62 +41.32%	208.31 +2.35%	**203.53** **0.00%**

In the instances of 100 nodes, FSA has the least cost of roughly € 57.07682, followed by R11, 48A, 2O, and RR having the greatest cost. For instances with 200 nodes, 48A has the least cost of 111.48 followed by FSA, R11, 2O, and RR having the greatest cost. For instances with 400 nodes, 48A has the least cost of 203.53 followed by R11, FSA, 2O, and RR having the greatest cost of 287.62.

c. Fitness

Figure 2 shows the progression of the leader flamingo's fitness (total service time) over the iterations of running FSA for N100 instances (100 customer nodes). It shows that the fitness of the leader flamingo is generally constant over the run of FSA, if not reduced by a negligible amount.

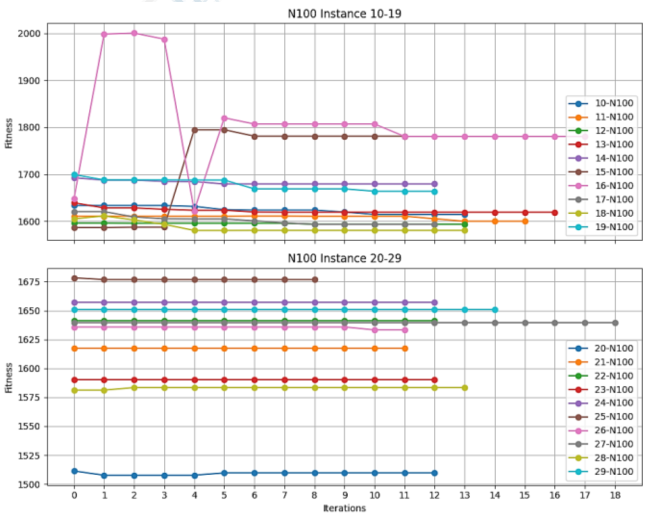

Figure 2. Fitness vs. Iteration with 100 customer nodes

Figure 3 shows the progression of the leader flamingo's fitness over the iterations of running FSA for N200 instances (200 customer nodes). It shows that for instances 10-19, FSA was able to generally reduce the leader flamingo's fitness by a noticeable amount. For instances 20-29, the leader flamingo's fitness remains constant.

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 03 March 2024

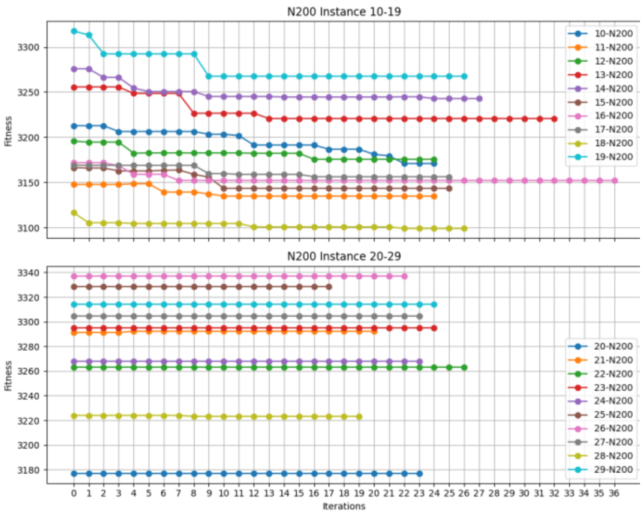

Figure 3. Fitness vs. Iteration with 200 customer nodes

Figure 4 shows the progression of the leader flamingo's fitness over the iterations of running FSA for N400 instances (400 customer nodes). It shows that the fitness of the leader flamingo is generally constant over the run of FSA, if not reduced by a negligible amount.

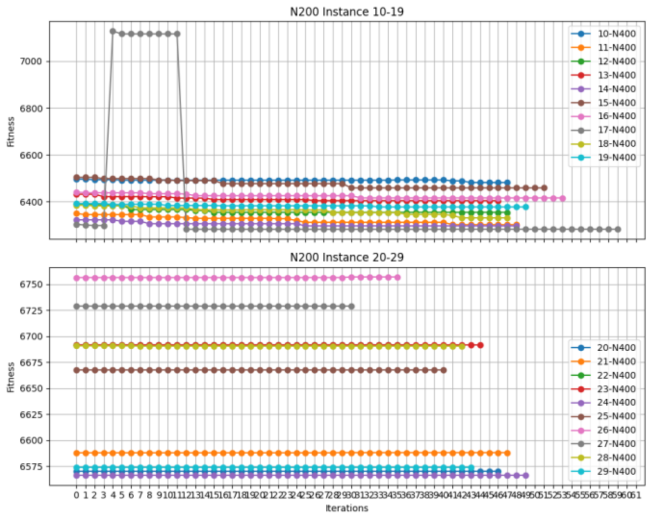

Figure 4. Fitness vs. Iteration with 400 customer nodes

d. CPU Execution Time

Table V shows the comparison of the CPU execution times between FSA and the approaches presented in Felipe et al.'s study. It shows that both 2O and RR has the fastest execution time in all the instances followed by FSA, R11 , and 48A. FSA also has a fast execution time for instances with 100 nodes. It also shows that for instances with 200 customer nodes, 2O and RR both run with the lowest execution time, followed by FSA, then followed by R11, with 48A coming last. The same order follows for instances with 400 customer nodes.

Table V. Ave. CPU Execution Time Comparison Between FSA And Other Algorithms

	FSA	2O	RR	R_1^1	48A
N=100	<1	<1	<1	2	57
N=200	6	<1	<1	15	348
N=400	52	<1	<1	91	198

V. CONCLUSIONS AND RECOMMENDATIONS

The Vehicle Routing Problem (VRP) poses a multifaceted optimization challenge crucial for industries seeking to enhance transportation efficiency. The escalating demands for swift deliveries, cost minimization, and environmental sustainability necessitate innovative VRP solutions. This study addresses the significance of devising novel approaches to tackle VRP complexities, considering factors like diverse customer demands and multiple constraints. Leveraging the Flamingo Search Algorithm (FSA), inspired by flamingo behavior, holds promise for overcoming optimization challenges and contributes to the advancement of efficient logistics systems.

This research concentrates on the application of FSA to optimize vehicle routing logistics. Given the dual behaviors of FSA—Foraging and Migratory—this paper introduces operators emulating these behaviors to enhance routing problem optimization: 2 operators for Foraging and 1 for Migratory. The dataset used is sourced from Felipe et al., and this study endeavors to compare its outcomes with those of Felipe et al., utilizing four metrics: Average Cost, Fitness, Number of Iterations, and CPU Execution Time.

The findings reveal that FSA exhibits competitive performance compared to contemporary approaches (FSA, 2O, RR, R11, 48A) in the literature. FSA proves to be the most cost-effective solution, outperforming other approaches in instances with 100 customer nodes. For instances with 200 and 400 customer nodes, FSA remains competitive, with a maximum of 12.19% higher cost than the optimal approach for the instance. In terms of fitness, FSA minimally impacts the leader flamingo's fitness. Regarding CPU execution time, FSA ranks among the most efficient approaches, boasting the third shortest execution time at most.

While this study delves into the primary behaviors of flamingos, we recommend exploring the three distinct foraging characteristics—Communicative, Beak Scanning, and Bipedal mobile behaviors—when implementing operators. Furthermore, experimenting with parameter values, such as the Convergence criterion and the Ratio of dividing flamingos into foraging and migratory groups, is advised. Future research could also investigate additional operators to mimic migratory behavior, thereby expanding the scope and depth of insights into optimizing VRP using FSA.

REFERENCES

[1] Archetti, C.; Speranza, G.; Vigo, D. (2014). "Vehicle routing problems with profits". In Toth, P.; Vigo, D. (eds.). Vehicle Routing: Problems, Methods, and Applications (Second ed.). pp. 273–297. Retrieved February 25, 2023, from doi:10.1137/1.9781611973594.ch10

[2] Bishop, J.M., (1989) Stochastic Searching Networks, Proc. 1st IEE Int. Conf. on Artificial Neural Networks, pp. 329-331, London, UK. Retrieved February 25, 2023.

[3] Byjus. (n.d.). Datasets. https://byjus.com/maths/data-sets/

[4] Cars, planes, trains: where do CO2 emissions from transport come from? (2020, October 6). Our World in Data. https://ourworldindata.org/co2-emissions-from-transport

[5] Dantzig, George Bernard; Ramser, John Hubert (1959, October). "The Truck Dispatching Problem" (PDF). Management Science. 6 (1): 80–91. Retrieved February 25, 2023, from doi:10.1287/mnsc.6.1.80

[6] Definition of algorithm. (2023). In the Merriam-Webster Dictionary. https://www.merriam-webster.com/dictionary/algorithm

[7] Dorigo, M., & Stutzle, T. (2004). Ant Colony Optimization. Retrieved February 25, 2023.

[8] Felipe, Á., Ortuño, M. T., Righini, G., & Tirado, G. (2014). A heuristic approach for the Green Vehicle Routing problem with multiple technologies and partial recharges. Transportation Research Part E: Logistics and Transportation Review, 71, 111–128. https://doi.org/10.1016/j.tre.2014.09.003

[9] Hassanien, A. E., & Emary, E. (2018, December 12). Swarm intelligence: Principles, advances, and applications: Aboul El. Taylor & Francis. Retrieved February 25, 2023, from https://www.taylorfrancis.com/books/mono/10.1201/9781315222455/swarm-intelligence-aboul-ella-hassanien-eid-emary

[10] Hu, J., Turgut, A. E., Krajnik, T., Lennox, B., & Arvin, F. (2020, August 21). Occlusion-based Coordination Protocol Design for Autonomous Robotic Shepherding Tasks. IEEE explore. Retrieved February 25, 2023, from https://ieeexplore.ieee.org/abstract/document/9173524

[11] Parsopoulos, K. E.; Vrahatis, M. N. (2002). "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization". Natural Computing. 1 (2–3): 235–306. doi:10.1023/A:1016568309421

[12] Tanase, G. C. (2021). The Increasing Adoption of Electric Vehicles (EVs) in The Last-Mile Delivery Operations. ideas.repec.org. https://ideas.repec.org/a/rdc/journl/v12y2021i3p33-37.html

[13] Tang, J., Liu, G., & Pan, Q. (2021, July 27). A review on representative swarm intelligence algorithms. IEEEexplore. Retrieved February 25, 2023, from https://ieeexplore.ieee.org/abstract/document/9498989

[14] Toth, P.; Vigo, D., eds. (2002). The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. Vol. 9. Philadelphia: Society for Industrial and Applied Mathematics. Retrieved February 25, 2023.

[15] Touati-Moungla, N., & Jost, V. (2011). Combinatorial optimization for electric vehicles management. Renewable Energy & Power Quality Journal. https://doi.org/10.24084/repqj09.504

[16] United Nations Environment Programme [UNEP]. (2022). Emissions Gap Report 2022. In United Nations Environment Programme (No. 978-92-807-3979–4). https://www.unep.org/resources/emissions-gap-report-2022

[17] Wang, Y., Zhe, J., Wang, X., Sun, Y., & Wang, H. (2022, May 31). Collaborative multi depot vehicle routing problem with dynamic customer demands and time windows. MDPI. Retrieved February 25, 2023, from https://www.mdpi.com/2071-1050/14/11/6709

[18] Zhiheng, W., & Jianhua, L. (2021, June 18). Flamingo Search Algorithm: A new swarm intelligence. IEEEexplore. Retrieved February 25, 2023, from https://ieeexplore.ieee.org/abstract/document/9459769/