A comparison of Artificial Bee Colony algorithm and the Genetic Algorithm with the purpose of minimizing the total distance for the Vehicle Routing Problem

Amel Mounia DJEBBAR[1], **Chérifa BOUDIA**[2]

[1] Graduate School of Economics of Oran, Algeria

[2] University Mustapha Stambouli of Mascara, Algeria

amel.djebbar@ese-oran.dz, cherifa.boudia@univ-mascara.dz

Abstract: Nowadays, the vehicle routing problem is one of the most important combinational optimization problems and it has received much attention because of its real application in industrial and service-related contexts. It is considered an important topic in the logistics industry and in the field of operations research. This paper focuses on the comparison between two metaheuristics namely the Genetic Algorithm (GA) and the Discrete Artificial Bee Colony (DABC) algorithm in order to solve the vehicle routing problem with a capacity constraint. In the first step, an initial population with good solutions is created, and in the second step, the routing problem is solved by employing the genetic algorithm which incorporates genetic operators and the discrete artificial bee colony algorithm which incorporates neighbourhood operators which are used for improving the obtained solutions. Experimental tests were performed on a set of 14 instances from the literature in the case of which the related number of customers ranges typically from 50 to 200, in order to assess the effectiveness of the two employed approaches. The computational results showed that the DABC algorithm obtained good solutions and a lower computational time in comparison with the GA algorithm. They also indicated that the DABC outperformed the state-of-the-art algorithms in the context of vehicle routing for certain instances.

Keywords: combinational optimization, logistics industry, operations research, metaheuristics, population.

1. Introduction

Route optimization is an excellent answer to planning problems. It has even become essential given the new challenges in the logistics industry, which require a combination of operational performance and reduced economic costs. The Vehicle Routing Problem (VRP) is an essential problem in logistics; it is stated as follows: given a set of vehicles and a set of clients, and assuming a fixed cost of traversing any client-client pair, find the path that reaches all locations at a minimum cost (Golden et al., 2008). This problem has numerous real-world applications, including disaster relief efforts, merchandise delivery to businesses, care visits, delivery in smart logistics, and others. The VRP is classified as a combinatorial optimization problem that can be considered as a fusion of traveling salesman problems and bin packing problem. Moreover, it is a basic problem arising from crew scheduling problems (Boyer et al., 2018). Combinatorial optimization is an essential sub-domain of computer science whose objective is to find the optimal solution under several constraints. So, VRP is known as an NP-hard problem (Archetti et al., 2011) in which the solution is hard to find and only a “good enough” solution will generally be obtained. Additionally, Lenstra & Rinnooy Kan (1981) studied its complexity and have found that the vehicle routing problem is NP-hard and therefore unsolved in polynomial time. The purpose of this work is to define the shortest possible routes for reaching a customer that can reduce the travel distance by planning the most efficient routes for all customers.

Several studies have approached the VRP, and they have proposed different methods that are classified as exact and heuristic methods. Exact methods comprise algorithms like branch and bound (Lawler & Wood, 1966), branch and price (Barnhart et al., 1998), and branch and cut (Gomory, 2010). To solve the problem, the exact methods divide it into sub-problems. However, they have a high level of time complexity; hence, it is difficult to solve NP-hard problems using exact algorithms. In fact, for solving optimization problems, the scientific community has mainly focused on metaheuristics, which can find quite good solutions with a competitive running time.

https://doi.org/10.33436/v32i3y202204

As a result, metaheuristic methods are increasingly being used in large-scale VRPs (Chen et al., 2010; Sharma & Saini, 2020). These include such algorithms as genetic algorithms, differential evolution algorithms, simulated annealing, tabu search, ant colony optimization, artificial bee colony, and particle swarm optimization. These algorithms can give satisfactory results with an acceptable execution time for different VRP problems (Akpinar, 2016). Furthermore, fast or slow convergence, where the solutions depend on the diversity of the initial population and different parameter values, may represent limitations for metaheuristics. Among the most challenging metaheuristic methods for solving the VRP problem, one can refer to the Genetic Algorithm (GA) and the Artificial Bee Colony algorithm (ABC). The two metaheuristics are well known for their efficiency in treating high-dimensional cost functions and offer a set of bio-inspired computing optimization algorithms (Wario et al., 2022). In addition, GA and ABC have been widely performed for several routing problems and are featured a great performance in many of these cases (Nazif & Lee, 2012). Indeed, GA is characterized by chromosomes, genes, population set, fitness function, selection, crossover, and mutation. The performance of genetic algorithms is largely influenced by the crossover and mutation operators. In its turn, the artificial bee colony metaheuristic is an optimization method activated by the attitude of nectar-collecting honey bees represented by the “waggle dance” (Karaboga et al., 2014). In the artificial bee colony algorithm, the artificial bees are distributed into three distinct groups: the employed bees, the onlookers, and the scouts. A bee that takes advantage of a food source is called an employed bee and the onlooker enjoys the dances of the employed bees. The duties entrusted to scout bees are to hunt for fresh food resources discretely in the neighborhood of the hive.

The principal contribution of this paper is to compare two metaheuristics, the genetic algorithm and the discrete artificial bee colony algorithm, for resolving the VRP, in order to capitalize on the benefits of both methods. Genetic and neighborhood operators were employed in order to further improve the solutions. An experimental study was carried out to show the competitiveness of the two algorithms. The remainder of this paper is organized as follows. The VRP is defined in Section 2. Section 3 illustrates different state-of-the-art methods employed for solving the VRP. Section 4 presents the GA and the proposed DABC algorithm in detail. Section 5 describes the VRP dataset and illustrates the computational experiments, and their interpretations. Section 6 provides the conclusion of this paper and possible future research directions.

2. Problem description

The main concern when solving an optimization problem is to solve the problem by satisfying all the constraints of the problem. On that principle, the objective function for VRP is defined to evaluate the obtained outcomes efficiently. VRP models use the fitness function to minimize the traveled distance and find the optimal solutions. The capacity of the vehicle is an added constraint that makes the VRP more real. This constraint ensures that a vehicle delivers demands within its capacity. The constraints of the vehicle routing problem are:

- A route begins and finishes at the depot;
- A customer is visited once and only once;
- A vehicle's load cannot exceed its capacity;
- All requests must be satisfied.

Figure 1 shows an example of this type of problem which involves 3 vehicles with 11 customers represented by nodes. Between two clients, the displayed distance is calculated as the Euclidean distance between the coordinates of the two nodes, and each client has a demand represented by a positive value.

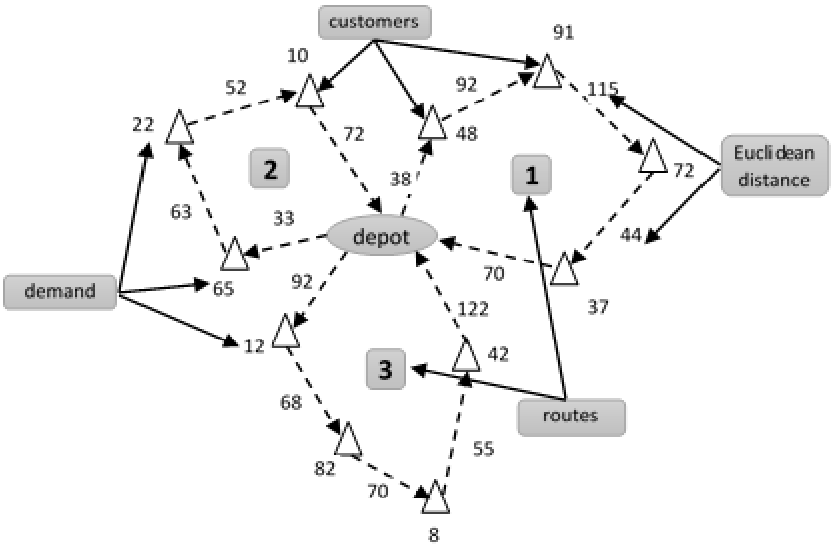


Figure 1. Example of a VRP solution with 11 customers and 3 routes

3. Literature review

The grand appearance and advancement of the routing problem has piqued the interest of many researchers, primarily due to its inherent combinatorial traits, making it very difficult to solve. Due to the complexities of VRP, metaheuristic optimization algorithms are usually employed for dealing with this type of challenge. For solving the vehicle routing problem with capacity constraint, Zhou et al. (2013) have applied the HBA-PR method, which is a hybrid bat algorithm with path re-linking. They obtained promising results with the proposed algorithm. Similarly, Chen et al. (2015) have hybridized a two-stage sweep algorithm, which was used for grouping all customers, and greedy search, to determine the shortest route for each vehicle for solving the VRP. Zhao et al. (2016) have implemented the discrete invasive weed optimization method to solve the VRP. They showed that the proposed algorithm was robust for discrete combinatorial optimization problems. To solve the vehicle routing problem with the objective of minimizing the traveled distance, Amous et al. (2017) proposed a variable neighborhood search approach based on different neighborhood methods for intensifying the search effort. In the same year, Hosseinabadi et al. (2017) introduced a new metaheuristic optimization algorithm based on group interactions and the law of gravity to solve the vehicle routing problem. They used two parameters, namely velocity and gravitational force in physics. The authors concluded that the proposed method could be a very efficient in order to solve the VRP with capacity constraint. Wei et al. (2018) have used a simulated annealing approach to solve the VRP. In a separate study, the GTS method is proposed, which combines the tabu search algorithm and the genetic algorithm for treating the vehicle routing problem. The authors concluded that the performance of the GTS is better than that of the genetic algorithm in terms of solutions. Helal et al. (2018) have used the theory of evidence to represent the uncertainty in customer requests. They have used the simulated annealing method for dealing with the vehicle routing problem with capacity constraint. Leggieri & Haouari (2018) have applied a novel metaheuristic method for the asymmetric capacitated vehicle routing problem that includes three sequential stages, reducing the problem size by eliminating unpromising arcs, deriving a satisfactory starting solution, and generating near-optimal solutions. For the purpose of determining the best path for waste collection, Hannan et al. (2018) adapted a particle swarm optimization algorithm. To improve routes, they used four local search algorithms: 2-Opt*, Or-Opt-1, 2-Opt, and Or-Opt. Their goal was to use a vehicle routing model with capacity constraints for minimizing total cost and total distance. Goel & Maini (2018) have hybridized the ant colony system that served as a basic framework and a firefly algorithm that was employed for exploring the search space for new solutions to solve the capacitated vehicle routing problem. Altabeeb et al. (2019) have implemented a hybrid Firefly Algorithm for a VRP that integrates local search algorithms and genetic approaches to FA. In their approach, initial fireflies were randomly generated under the capacity constraint. They have used the hamming distance to estimate the distance between two fireflies. They have also used Partially-matched crossover to produce two novel offsprings. They employed local search algorithms, improved 2-opt and 2-h-opt to improve the solution to the problem, and they also employed two types of mutation, inter-route swapping and intra-route swapping, to prevent premature convergence.

More recently, Thammano & Rungwachira (2021) have developed a hybridized ant colony system, a sweep algorithm, and a path relinking for obtaining a better solution from a pair of guiding and initial solutions to solve CVRP (Capacitated Vehicle Routing Problem). They used a sweep algorithm to create an initial population, and a revised ant colony system to create new generations of good solutions. Dalbah et al. (2021) have developed the Coronavirus Herd Immunity Optimizer (CHIO) to solve the CVRP. They have modified CHIO to use an efficient repair process in its initial stage and in the evolution loop in order to maintain the feasibility of CVRP solutions. They have demonstrated that the modified CHIO solved the vehicle routing problem with capacity constraint efficiently. Finally, Elshaer & Awad (2020) and Zhang et al. (2022) employed other metaheuristic in order to solve the VRP.

Even though the routing problem is a NP-hard combinatorial optimization problem, many researchers are still trying to find other algorithms to resolve the problem efficiently.

4. The proposed methodology

This section describes the application of the proposed GA and DABC approaches for the assignment of customers to routes, the scheduling of customers within the routes, and determining the optimal solution for the routing problem.

4.1. The representation of the solution of the initial population

A coding mechanism based on an integer string called the array representation is employed in order to represent the set of solutions to the problem. The size of the string is n+m+1, where the number of customers is represented by the variable n and the number of vehicles is represented by the variable m. There is also m+1 0s in the string representing the start of each vehicle route. The sequence between two 0s is the set of customers visited by a single vehicle. The string used in this way determines the set of routes of the problem, where each integer represents either the depot or a customer node. Using this encoding mechanism, neighborhood operators or genetic operators (Deb, 1999; García Nájera & Bullinaria, 2011) can be successfully applied to the problem.

The initial population was generated randomly. Each solution was constructed step-by-step using the Initialization algorithm. This algorithm was constructed by affecting one customer at a time on one of the m routes. The selection of the customer is done randomly. The customer is then moved to a position that minimizes the total distance. The initialization algorithm is repeated until all clients are affected. A total of t initial solutions is generated by the employed procedure. The initialization algorithm for the initial population is defined as follows:

Algorithm 1 Initialization Algorithm

```
Let m = 0
Let list that contains all nodes
Repeat
Initialize a route r with 0
m = m+1
For all the routes r do
A node $n_i$ is selected at random from the list
```

```
Delete the node
Calculate the distance between $n_i$ and the other nodes
Assign $n_i$ to the nearest vehicle
Update the position of the vehicle with the location of the client
Then add it to the corresponding route r
Add the routes to the solution list
Until all the solutions of the population are generated.
```

4.2. The formulation of the objective function and fitness function

The objective function of the proposed vehicle routing problem lies in minimizing the value of a solution. Then it is converted into a fitness function, which defines the quality of a solution. Function (1) gives the equation for calculating the total distance traveled by all vehicles, where d_{ij} represents the distance between i and j and was calculated using Euclidean distances and x_{ij} is a binary variable, $x_{ij} = 1$ when arc (i, j) exists and otherwise $x_{ijk} = 0$. The fitness function of the solution is given in equation (2).

$$f(x) = \sum_{n}^{i,j=1} x_{ij} * d_{ij} \tag{1}$$

$$fitness = \frac{1}{f(x)} \tag{2}$$

4.3. The Discrete Artificial Bee Colony algorithm

4.3.1. Principle

The Discrete Artificial Bee Colony (DABC) is an iterative algorithm (Ng et al., 2017; Khan & Maiti, 2019). The initialization phase consists in the generation of a set of random solutions; each solution represents a food source that is assigned to each employed bee. Then, for each iteration, each employed bee will find a new food source based on the old food source using neighborhood operators. The quantity of nectar that represents the fitness value of the novel food source is then evaluated. If the novel solution (food source) contains more nectar than the old solution (food source, the new one will replace the old one. Once the employed bees complete the above steps, the nectar information solutions have been shared with the onlooker bees. Each onlooker chooses a solution using the roulette selection mechanism. Next, each onlooker finds a food source near her selected food source using neighborhood operators and calculates the amount of nectar from the nearby food source. An employed bee abandons a food source if its value is worse than the previous food source. After a predetermined number, this solution becomes a scout bee and searches for a novel food source at random. Once the scout has found a new solution, the scout is again involved as an employed bee. A new iteration of the DABC algorithm begins once a food source is allotted to every employed bee. The algorithm is reiterated until a given condition is satisfied. The steps of the DABC algorithm can be summarized as follows:

Algorithm 2 DABC

```
Initialize all parameters,
while iter < max_iter do
 Employed bees phase
    For every employed bee do
```

```
            Apply randomly the neighborhood operators cited in section 4.3.2,
            Apply greedy selection,
        Onlooker bees phase
            For every onlooker bee do
                Apply randomly the neighborhood operators cited in section 4.3.2,
                Apply greedy selection,
        Scouts bee phase
            Generate new food sources to replace the rejected food sources.
        Memorize the best food source.
    End of while.
```

4.3.2. Neighborhood operators

A neighborhood operator is employed in order to obtain a new solution *nvx* from the current solution *x* in the employed bee phase and the onlooker phase of the DABC algorithm. The neighborhood operators are employed to improve the solutions and the convergence of the algorithm (Yang, 2014). Thus, a faster convergence was achieved, and higher quality solutions were obtained (Yu & Lin, 2016). In this paper, instead of using a single operator, a combination of three operators is used. When a new solution *nvx* is required, a neighborhood operator is randomly selected in order to be applied to solution *x* from the set of the selected neighborhood operators. Different combinations of neighborhood operators have been tested, and it has been concluded that random swaps, random swaps of subsequences, and random swaps of reversed subsequences can give the most effective results.

a. Random swaps

This operator randomly selects the positions *i* and *j*, with $i \neq j$ in the integer string, and permutes the client located in position *i* with the client located in position *j*.

b. Random swaps of subsequences

In this operator, two subsequences of clients and deposits of random length are selected and swapped.

c. Random swaps of reversed subsequences

In this operator, two subsequences of customers and deposits of random length are chosen and swapped. Then, each of the swapped subsequences can be inversed with a probability of 50%.

4.3.3. Solution selection

In each iteration of the DABC algorithm, every onlooker bee chooses a solution (food source) by using the selection probabilities related to food sources for each solution, and a solution x_i is selected based on the correlative. The probability of choosing solution x_i is given in equation (3), where $fitness_i$ represents the fitness of the solution x_i .

$$P_i = \frac{fitness_i}{\sum_{i=1}^{n} fitness_i} \qquad (3)$$

4.4. The genetic algorithm

4.4.1. Principle

The genetic algorithm was designed by Holland (1992). It uses the idea of genetics and the "survival of the fittest" to produce near-optimal solutions to problems such as the traveling salesman problem, machine scheduling problems, vehicle routing problems, and many others. A genetic algorithm can be divided into several subparts namely representation, initialization, fitness function evaluation, selection mechanism, recombination operators that include crossover and mutation, and termination. A new child is integrated into the population if his chromosomes have a better fitness value than the current chromosome to be substituted. There are various genetic operators in the literature (Jie-sheng et al., 2011; Kamkar et al., 2010), and in this paper the ordered crossover (OX) according to (Karakatič & Podgorelec, 2015) was chosen along with the swap mutation for its easy implementation. The order crossover produces two children from two parents, and the swap mutation allows for solution diversification. The steps of GA are given in Algorithm 3.

Algorithm 3 Genetic Algorithm

```
Initialize population and parameters
Evaluate the fitness function
while iter < max_iter do
  Select the subset of the population by Roulette wheel selection
  Apply OX operator
  Apply swap mutation operator
  Evaluate the fitness function
  The chromosomes with the worst fitness value are removed;
End of while.
```

4.4.2. The selection method

Selection is a step, which consists in choosing the part of the population that will be reproduced in the next step. It consists in selecting the chromosomes that have the highest probabilities. In our work, we used the roulette wheel selection method, in which an individual's chances of being selected are proportional to his fitness value; thus, selection can be imagined as a spinning roulette wheel, with each individual taking up an amount of space on the roulette wheel based on their fitness.

4.4.3. The crossover operator

Various studies have been conducted to understand crossover operators theoretically (Doerr et al., 2013). The solutions are improved by the crossover operator in the population and make the convergence of the algorithm easier. In this paper, the OX operator was used. It creates offspring solutions within a subspace limited by the parents, and if the parents have smaller fitness values than the offspring solutions, the parent solutions are replaced by the offspring solutions in the new population. The order crossover is constructed in such a way that a part of the first parent is copied to the chromosomes of the offspring and the remaining nodes are positioned in the offspring in the same order as they were in the other parent (Puljić & Manger, 2013). The steps for the order crossover are the following:

1. Select two random cut points for each parent.
2. Consecutive nodes between two cut points in parent 1 are copied into child 1, and consecutive nodes between two cut points in parent 2 are copied into child 2.

3. From child 1, the rest of the genes are copied in the order they appear in parent 2, starting after the second cut point (excluding a certain value if it was already inserted) until all positions are filled.
4. From child 2, the rest of the genes are copied in the order they appear in parent 1, starting after the second cut point (excluding a certain value if it was already inserted) until all positions are filled.

4.4.4. The mutation operator

For the sake of simplification, mutation is a random permutation in a solution to obtain a novel solution. The advantage of a mutation is that it preserves and diversifies the population. In the context of the proposed method, a swap mutation was chosen which consists in selecting two positions randomly on the chromosome and exchanging their values.

5. Computational results

The environment of work is Python 3.7.4, which is a powerful language ideal for scripting and fast application development in several fields on most platforms; Jupyter Notebook was chosen, which is an interactive computational environment. The computations were carried out using a personal computer with Intel Core i5 2.60 GHz CPU and 4 GB of RAM. The personal computer runs on the Windows 8 operating system. The two metaheuristics were tested on 14 well-known benchmark instances created by Christofides et al. (1979). The instances are set as follows: depot, number of customers (50-199), and the number of vehicles (5-18).

The main purpose of the experiment was to compare the proposed GA approach to the DABC approach. The first experimental scenarios are dedicated to studying the impact of the different parameters on the efficiency of the two algorithms: the genetic algorithm and the discrete artificial bee colony algorithm. Robust parameterization is required for both algorithms so that they can perform on the different datasets. The parameters are studied for tended to cover a small number of solutions from the problem search space, which esulted in a high speed of convergence. For the DABC, in order to select the best parameterization, trials were conducted on four elements: pop (population size or number of food sources), number of iterations, and limit (number of times the solution is dropped). Based on (Karaboga et al., 2014), the total number of employed bees and the total number of onlooker bees were the same. It was found that the colony size (i.e., the total number of employed and onlooker bees) is equal to 50, which can provide a satisfactory convergence speed for the search space. The maximum number of iterations is set to 200*n and the limit is 50*n. For the GA, the population size was set to 25 and the maximum number of iterations to 500*n. For the proposed GA approach, the iterations are repeated for each dataset where the best value of the objective function and the best computation time are reported. In addition, the proposed DABC algorithm is repeated five times (each run is iterated 500*n) for each dataset for which the best value of the objective function and the average computation time are reported.

Table 1. Computational results for the benchmark datasets of (Christofides et al., 1979)

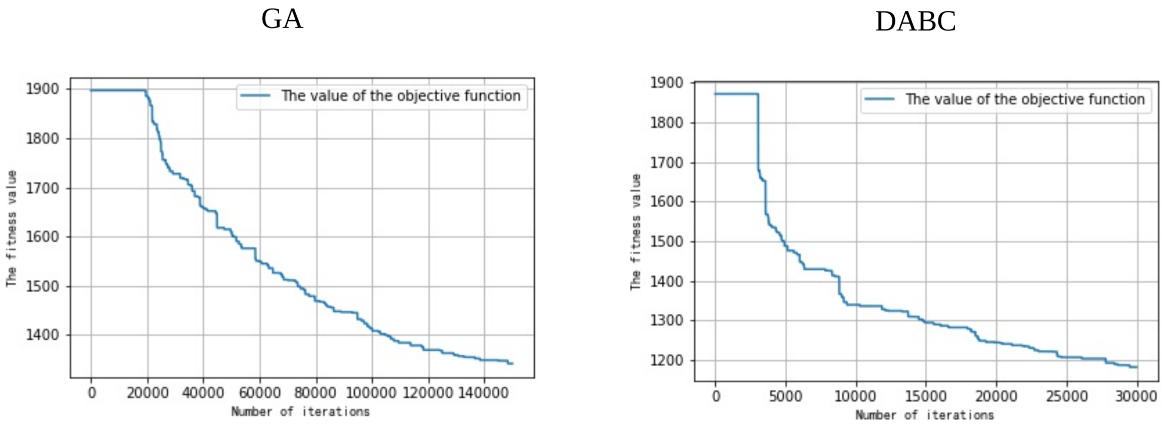
Instance	No. of cust.	No. of routes	Q	No. of iterations	Objective function (distance)		Computation time	
					GA	DABC	GA	DABC
CMT1	50	5	160	25000	573.89	531.18	26.50	7.50
CMT2	75	10	140	37500	977.97	879.72	60.55	16.40
CMT3	100	8	200	50000	1064.68	880.84	158.20	24.80
CMT4	150	12	200	75000	1360.82	1167.91	282.90	54.40
CMT5	199	17	200	99500	1913.07	1536.47	379.70	103.70
CMT6	50	6	160	25000	593.66	**537.83**	29.20	7.30
CMT7	75	11	140	37500	950.99	**884.82**	239.70	16.20

CMT8	100	9	200	50000	965.03	883.22	115.20	26.30
CMT9	150	14	200	75000	1321.86	**1154.72**	82.25	53.80
CMT10	199	18	200	99500	1813.92	1544.04	226.60	99.50
CMT11	120	7	200	60000	1477.94	1209.87	47.80	34.40
CMT12	100	10	200	50000	882.64	827.47	34.90	25.40
CMT13	120	11	200	60000	1231.99	**1161.02**	54.18	36.30
CMT14	100	11	200	50000	997.46	**836.05**	755.60	27.80

The computational results for the fourteen chosen datasets are illustrated in Table 1. The columns from left to right represent, the number of clients, the number of routes, the capacity Q, the number of iterations, the objective function for GA and DABC, and the computation time in minutes for GA and DABC, respectively. The best results for the objective function obtained by the DABC approach among the fourteen chosen instances are higher than those obtained by the GA approach. However, for large problem instances, the value of the objective function increased for both methods. The obtained computation time for DABC is much better than for GA. Nevertheless, the proposed methods require a longer computation time for large problem instances. The increasing computation time for the proposed approaches can be justified by the fact that they simultaneously create a routing sequence and affect the customers' routes in DABC and GA iterations. The proposed DABC approach outperformed the GA approach in certain instances, which were written in bold in Table 1.

Figure 2 shows the change in the obtained solutions for GA and DABC approaches for the dataset CMT4 as an example. The performance of the two proposed approaches has gradually improved until the last iterations of the two algorithms for all cases, as it is shown in the graphs below, where the x-axis indicates the number of iterations and the y-axis represents the fitness value. The global best values for GA are stable for the first iterations and begin to drop slowly in the later iterations and continue to drop. By contrast, the global best values for DABC drop very fast for the first iterations, and drop gradually in the later iterations, and continue to drop fast. The other problem instances display the same convergence for the two approaches. For the CMT4 instance, the GA solution decreased slowly from the beginning, which shows that the algorithm did not reach the optimal solution due to the size of the instance since it belongs to the set of large-scale instances. On the other hand, the optimal value of the objective function for the DABC dropped rapidly in the first iterations and decreased slowly in the following iterations until it became stable after 25,000 iterations.

Besides, the GA for the VRP usually proves to have a slow convergence speed, and therefore it is easy to trap local best solutions (Qing-dao-er-ji & Wang, 2012). For this reason, the implementation of the DABC algorithm in optimization problems is bound to reach a high precision in addition to reducing the execution time and generating high-level results.


GA

DABC

Figure 2. The change in the obtained solutions for the CMT4 dataset

Figure 3 depicts the optimal solution for the CMT1 dataset as a graph of vertices and edges using AG and DABC. The vertex represents the customer, and the concatenation of the edges determines the routes of the vehicle.

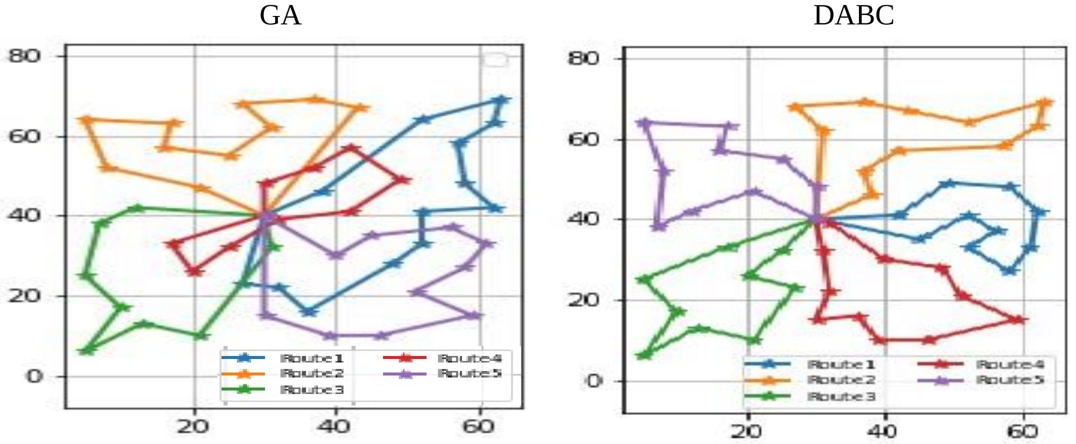


Figure 3. Best routes for the CMT1 dataset

The circle in the middle of the graph represents the depot. Each colored route represents a vehicle route that starts from and ends at the same depot. However, in Figure 3 it can be seen that when the number of vehicles and the total number of clients increases, the complexity of the problem increases and the network plot also becomes more complex.

5. Conclusion

The vehicle routing problem consists in finding the optimal paths for a number of vehicles with the constraint of maintaining the vehicle capacities, in which the vehicles are responsible for delivering goods to a number of clients. This problem is considered a NP-hard one due to its complex nature. This is why metaheuristic optimization algorithms are widely used to deal with this type of challenge. In this article, two metaheuristics, namely GA and DABC, are compared for solving the VRP. This paper used a heuristic for generating good initial solutions, then GA, and the DABC for building new generations of good solutions. Finally, the solutions were improved by employing genetic and neighborhood operators. In order to measure the effectiveness of the two algorithms, 14 datasets for VRP were employed, with the purpose of showing their impact on the GA and DABC performance when VRP was solved. The selected and reasonable values of the parameters lead to the best configuration for the two algorithms and they were determined through preliminary experiments. A comparative study was carried out for the two proposed approaches. Computational experiments on benchmark datasets demonstrated the effectiveness of the two proposed approaches. However, they needed a longer computation time for large problem instances. In addition, DABC outperformed AG regarding the value of the optimal solution and execution time. Furthermore, the proposed DABC approach obtained excellent results for some instances in terms of solution quality in comparison with the results obtained by Christofides et al. (1979). While one obtained better solutions than the existing ones, unfortunately, it was not possible to find optimal solutions to some instances of the problem. The application of the proposed approach to other variants of VRP and its comparison with other algorithms, such as the ant colony optimization algorithm to solve VRP, are other possible subjects for future research.

REFERENCES

1. Akpinar, S. (2016). Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications,* 61, 28-38.

2. Altabeeb, A. M., Mohsen, A. M. & Ghallab, A. (2019). An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing,* 84, 105728.

3. Amous, M., Toumi, S., Jarboui, B., Eddaly, M. (2017). A variable neighborhood search algorithm for the capacitated vehicle routing problem. *Electronic Notes in Discrete Mathematics,* 58, 231-238.

4. Archetti, C., Feillet, D., Gendreau, M., Grazia Speranza, M. (2011). Complexity of the VRP and SDVRP. *Transportation Research Part C - Emerging Technologies,* 19(5), 741-750.

5. Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., Vance, P. H. (1998). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research,* 46(3), 316-329.

6. Boyer, V., Ibarra-Rojas, O. J. & Ríos-Solís, Y. Á. (2018). Vehicle and Crew Scheduling for Flexible Bus Transportation Systems. *Transportation Research Part B: Methodological,* 112, 216–229.

7. Chen, M.-H., Chang, P.-C., Chiu, C.-Y., Annadurai, S. P. (2015). A hybrid two-stage sweep algorithm for capacitated vehicle routing problem. *International Conference on Control, Automation and Robotics*, 195-199.

8. Chen, P., Huang, H. & Dong, X.-Y. (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications,* 37(2), 1620-1627.

9. Christofides, N., Mingozzi, A. & Toth, P. (1979). *The vehicle routing problem*. Chichester; New York: Wiley.

10. Dalbah, L. M., Al-Betar, M. A., Awadallah, M. A., Zitar, R. A. (2021). A modified coronavirus herd immunity optimizer for capacitated vehicle routing problem. *Journal of King Saud University - Computer and Information Sciences*, 10.1016/j.jksuci.2021.06.013.

11. Deb, K. (1999). An introduction to genetic algorithms. *Sadhana,* 24(4), 293-315.

12. Doerr, B., Johannsen, D., Kötzing, T., Neumann, F., Theile, M. (2013). More effective crossover operators for the all-pairs shortest path problem. *Theoretical Computer Science,* 471, 12-26.

13. Elshaer, R. & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering,* 140, 106242.

14. García Nájera, A. & Bullinaria, J. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research,* 38(1), 287-300.

15. Golden, B. L., Raghavan, S., Wasi, E. (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer LLC.

16. Goel, R. & Maini, R. (2018). A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *Journal of Computational Science,* 25, 28-37.

17. Gomory, R. E. (2010). Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem. In Jünger, M. et al. (Eds.). *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art.* Berlin, Heidelberg: Springer. pp. 77-103.

18. Hannan, M. A., Akhtar, M., Begum, R. A., Basri, H., Hussain, A., Scavino, E. (2018). Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm. *Waste Management,* 71, 31-41.

19. Helal, N., Pichon, F., Porumbel, D., Mercier, D., Lefèvre, É. (2018). The capacitated vehicle routing problem with evidential demands. *International Journal of Approximate Reasoning,* 95, 124-151.

20. Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press.

21. Hosseinabadi, A. A. R., Rostami, N. S. H., Kardgar, M., Mirkamali, S., Abraham, A. (2017). A new efficient approach for solving the capacitated Vehicle Routing Problem using the Gravitational Emulation Local Search Algorithm. *Applied Mathematical Modelling,* 49, 663-679.

22. Jie-sheng, W., Chang, L. & Ying, Z. (2011). *Solving capacitated vehicle routing problem based on improved genetic algorithm*. Chinese Control and Decision Conference (CCDC), May 2011. 60-64.

23. Kamkar, I., Poostchi, M. & Akbarzadeh Totonchi, M. R. (2010). A Cellular Genetic Algorithm for Solving the Vehicle Routing Problem with Time Windows. In Gao, X.-Z. et al. (Eds). Soft *Computing in Industrial Applications. Advances in Intelligent and Soft Computing*, Berlin, Heidelberg: Springer. pp. 263-270.

24. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 42(1), 21-57.

25. Karakatič, S. & Podgorelec, V. (2015). A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, 27, 519-532.

26. Khan, I. & Maiti, M. K. (2019). A swap sequence based Artificial Bee Colony algorithm for Traveling Salesman Problem. *Swarm and Evolutionary Computation,* 44, 428-438.

27. Lawler, E. L. & Wood, D. E. (1966). Branch-And-Bound Methods: A Survey. *Operations Research*, 14(4), 699-719. *JSTOR.*

28. Leggieri, V. & Haouari, M. (2018). A matheuristic for the asymmetric capacitated vehicle routing problem. *Discrete Applied Mathematics*, 234, 139-150.

29. Lenstra, J. K. & Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks,* 11(2), 221-227.

30. Nazif, H. & Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling,* 36(5), 2110-2117.

31. Ng, K. K. H., Lee, C.K.M., Zhang, S. Z., Wu, K., Ho, W. (2017). A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion. *Computers & Industrial Engineering,* 109, 151-168.

32. Puljić, K. & Manger, R. (2013). Comparison of eight evolutionary crossover operators for the vehicle routing problem. *Mathematical Communications*, 18(2), 359-375.

33. Qing-dao-er-ji, R. & Wang, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research,* 39(10), 2291-2299.

34. Sharma, R. & Saini, S. (2020). Heuristics and Meta-Heuristics Based Multiple Depot Vehicle Routing Problem: A Review. In: 2*020 International Conference on Electronics and Sustainable Communication Systems (ICESC),* July 2020 (pp. 683-689).

35. Thammano, A. & Rungwachira, P. (2021). Hybrid modified ant system with sweep algorithm and path relinking for the capacitated vehicle routing problem. *Heliyon*, 7(9), e08029.

36. Wario, F., Avalos, O. & Gálvez, J. (2022). Chapter 11 - Bio-inspired algorithms. In: Torres García, A. A. et al. (Eds). *Biosignal Processing and Classification Using Computational Learning and Intelligence*. Academic Press. pp. 225-248.

37. Wei, L., Zhang, Z., Zhang, D., Leung, S.C.H. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research,* 265(3), 843-859.

38. Yang, X.-S. (2014). *Nature-Inspired Optimization Algorithms*. 1st edition, London, Elsevier.

39. Yu, V. F. & Lin, S.-Y. (2016). Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. *International Journal of Production Research,* 54(2), 526-549.

40. Zhang, H., Ge, H., Yang, J., Tong, Y. (2022). Review of Vehicle Routing Problems: Models, Classification and Solving Algorithms. *Archives of Computational Methods in Engineering,* 29, 195-221.

41. Zhao, Y., Leng, L., Qian, Z., Wang, W. (2016). A Discrete Hybrid Invasive Weed Optimization Algorithm for the Capacitated Vehicle Routing Problem. *Procedia Computer Science,* 91, 978-987.

42. Zhou, Y., Xie, J. & Zheng, H. (2013). A Hybrid Bat Algorithm with Path Relinking for Capacitated Vehicle Routing Problem. *Mathematical Problems in Engineering,* 2013, e392789.

Amel Mounia DJEBBAR, PhD in Computer Science from the University of Science and Technology of Oran Mohamed-Boudiaf, Algeria. Presently, she is working as a Professor at the Graduate School of Economics of Oran, Algeria. She carries out research in the fields of Optimization, Artificial Intelligence, Operational Research, Big Data and machine learning techniques using C++ and Python.

Chérifa BOUDIA obtained her Ph.D. in Computer Science in 2020 at the University of Oran 1 Ahmed Ben Bella, Algeria. Currently, she is an Associate Professor at the University Mustapha Stambouli of Mascara, Algeria. She carries out research in the fields of Artificial intelligence and Education. She has been focusing on e-learning methods for the Java programming language, collaborative e-learning, e-learning technology, e-learning systems and developing a collaborative platform for e-learning at the University of Oran 1 Ahmed Ben Bella.