Chapter

Comparative Study of Algorithms Metaheuristics Based Applied to the Solution of the Capacitated Vehicle Routing Problem

Fernando Francisco Sandoya Sánchez,
Carmen Andrea Letamendi Lazo
and Fanny Yamel Sanabria Quiñónez

Abstract

This chapter presents the best-known heuristics and metaheuristics that are applied to solve the capacitated vehicle routing problem (CVRP), which is the generalization of the TSP, in which the nodes are visited by more than one route. To find out which algorithm obtains better results, there are 30 test instances used, which are grouped into 3 sets of problems according to the position of the nodes. The study begins with an economic impact analysis of the transportation sector in companies, which represents up to 20% of the final cost of the product. This case study focuses on the CVRP for its acronym capacitated vehicle routing problem, analyzing the best-known heuristics such as Clarke & Wright and sweep, and the algorithms GRASP and simulated annealing metaheuristics based.

Keywords: vehicle routing problem, VRP, CVRP, heuristics, metaheuristics, Clarke & Wright, sweep heuristic, GRASP, simulated annealing

1. Introduction

Logistics as a science has its origins in the military area; the transportation of weapons, food, and men at the service was coordinated through it. With the passage of time, the concept began to be applied in the business field, and for a long period of time, the logistics function was considered as a habitual, operational, and necessary activity to take the products from the seller to the buyer [1]. A little later, starting in the 1950s, a cycle of growth and constant demand increase was experienced throughout the world, which caused the production and sales capacity to exceed the companies' ability to distribute products. Thus, in those years, delivering orders on time became a problem due to poor compliance. Then, in 1980, the concept of response time was created, which is the union between the concept of physical distribution and material management; specialists realized that the faster the response time to the customer, the more the profitability of the company increased.

As the concepts were changing, the methods as well and the companies looked for ways to become efficient; in this way they expanded the activities related to logistics and determined that one of the heaviest items is transportation costs, representing on average, between 10% and 20% of the final cost of the product or service [2].

Although transport decisions are expressed in a variety of ways, the main ones are mode selection, route design, vehicle programming, and shipment consolidation [1]. In relation to the route design problem, the problem is commonly known as vehicle routing problem (VRP). Both the companies that own the transport service as part of their processes and the companies that provide the service seek to optimize resources within the route selection, since a good selection brings savings in time, resources such as fuel, maintenance of the fleet, salaries, and improvements, among others, in service indicators as a promise of product delivery.

The VRP can be considered as the natural extension of the TSP, in the sense that unlike the TSP, in the VRP we consider that the vehicles, or the agents in charge of providing a service to the nodes, have a limited capacity; therefore, most likely the entire route cannot be made through a single route, with a single vehicle that leaves and returns to the storage, traveling all the nodes, but to respect the restriction of the limited capacity of the vehicles so. In general, several routes are required, or what is the same, the solution of the VRP will be a set of Hamiltonian cycles that start from the deposit and such that each node is traveled only once.

2. Vehicle routing problem

The vehicle routing problem (VRP) consists in determining a set of routes for a fleet of vehicles that depart from one or more warehouses to meet the demand of several geographically dispersed customers [3].

The VRP objective is to meet the demand of the customers, optimizing some objective, which is generally the total cost involved in the routes, which is affected by the vehicular congestion of large cities, the high-energy consumption of cargo vehicles, and other factors.

Since the VRP problem is a generalization of the TSP, and knowing that the TSP is of the NP-hard problem class [4], it is concluded that the VRP is also a difficult problem of the NP-hard class.

The VRP model has many classifications by the different characteristics that can be included or considered in it. The most basic version is reflected with the CVRP capability (for the acronym of capacitated vehicle routing problem). The CVRP has the following assumptions:

The fleet of vehicles is homogeneous, that is to say all cargo vehicles have the same characteristics:

I. The demand is known in advance, that is, the quantity to be delivered for each client is known; this means that the demand is deterministic.

II. Each vehicle will carry the entire delivery to customers, prohibiting the distribution of fractional or partial loads that would later be completed by another vehicle.

III. All vehicles in the fleet have exactly the same load capacity.

IV. The starting point of the vehicles is only one and is considered a central warehouse.

V. Vehicles have capacity restrictions that are known in advance.

3. Classic heuristics to solve VRP

Heuristics are simple processes that perform a limited space search and generate acceptable solutions in moderate calculation times; an important characteristic of these methods is that they are designed to solve a specific optimization problem, and in general they cannot be used to solve other optimization problems. A more advanced class are the so-called metaheuristics, which are considered more advanced methods than heuristics, in the sense that they guide their construction and, therefore, are general purpose [5].

There are many advantages, and also disadvantages, when using heuristic algorithm methods to solve optimization problems, as described [6] within the reasons to use heuristic methods which are as follows:

I. The problem is that no exact method for its resolution is known.

II. Although there is an exact method to solve the problem, its execution is computationally very expensive.

III. A heuristic method is more flexible than an exact method, that is, difficult modeling conditions can be incorporated.

IV. The heuristic method is used as part of a global process that certifies an optimal solution. There are two possibilities:

   a. The heuristic method provides a good initial starting solution.

   b. The heuristic method participates in an intermediate step of the procedure, such as the selection rules of the variable to enter the base in the simplex method.

3.1 Savings based heuristics

There are several types of heuristic methods to solve the VRP, which are addressed extensively in Braekers et al. [3], trying to generate broad, nonexclusive categories, where the best-known heuristics are located to solve this problem; among them one of the most used and popular algorithms is the one of Clarke & Wright, and that has had contributions from different authors [7].

This algorithm is based on successively combining subtours until a Hamiltonian cycle is obtained, of which the subtours have a common node or vertex called base or initial.

The method can be described as follows:

- Having a solution of two different routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ can be joined by creating a new route $(0, \dots, i, j, \dots, 0)$.

- The distance savings obtained by the union is

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \tag{1}$$

In Eq. (1) s_{ij} is the savings on the total distance traveled if the two routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ are joined.

- An initial solution is started in this algorithm, and the unions that give greater savings are made as long as they do not violate the restrictions of the problem.

- When the maximum saving is negative, the combinations of the routes will increase the distance traveled, but the amount of routes in the solution will decrease; depending on the characteristics of each problem, this can generate circular or radial routes that can be avoided by placing a reference value λ, which penalizes the union routes with distant customers. Saving is proposed as

$$s_{ij} = c_{i0} + c_{0j} - \lambda c_{ij} \tag{2}$$

In Eq. (2) s_{ij} represents penalized savings with the weight λ in the total distance traveled if the two routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ are joined, which prevents when possible, merging routes with nodes far apart.

3.1.1 Application of savings algorithm

Step 1: With the coordinates of each client or city, prepare the distance matrix.
Step 2: Calculate the s_{ij} savings table for each pair of nodes.
Step 3: For each client or city i, build the route $(0, i, 0)$.
Step 4: Order savings from highest to lowest.
Step 5: Starting with the greatest savings, join the corresponding nodes, so that $s_{ij} = max\ s_{ij}$, where the maximum is taken between the savings that have not yet been considered; the route, $r_{i^* j^*}$, will be created, if i^* is the last customer of de r_{i^*}; and j^* is the first customer of r_{j^*}. Remove $s_{i^* j^*}$ from future considerations. Repeat step 5 until there are no more combinations of savings.

3.1.2 Example of application of the savings algorithm

A company wants to solve the problem of routing and design of the fleet of its product x to its 10 customers in the city and has a homogeneous fleet of trucks with capacity for 100 units of x product, with locations and demand shown in **Table 1**.

There are details in the Cartesian coordinates of the warehouse and each customer with the demand, while in **Figure 1**, the position of each customer and the warehouse is shown.

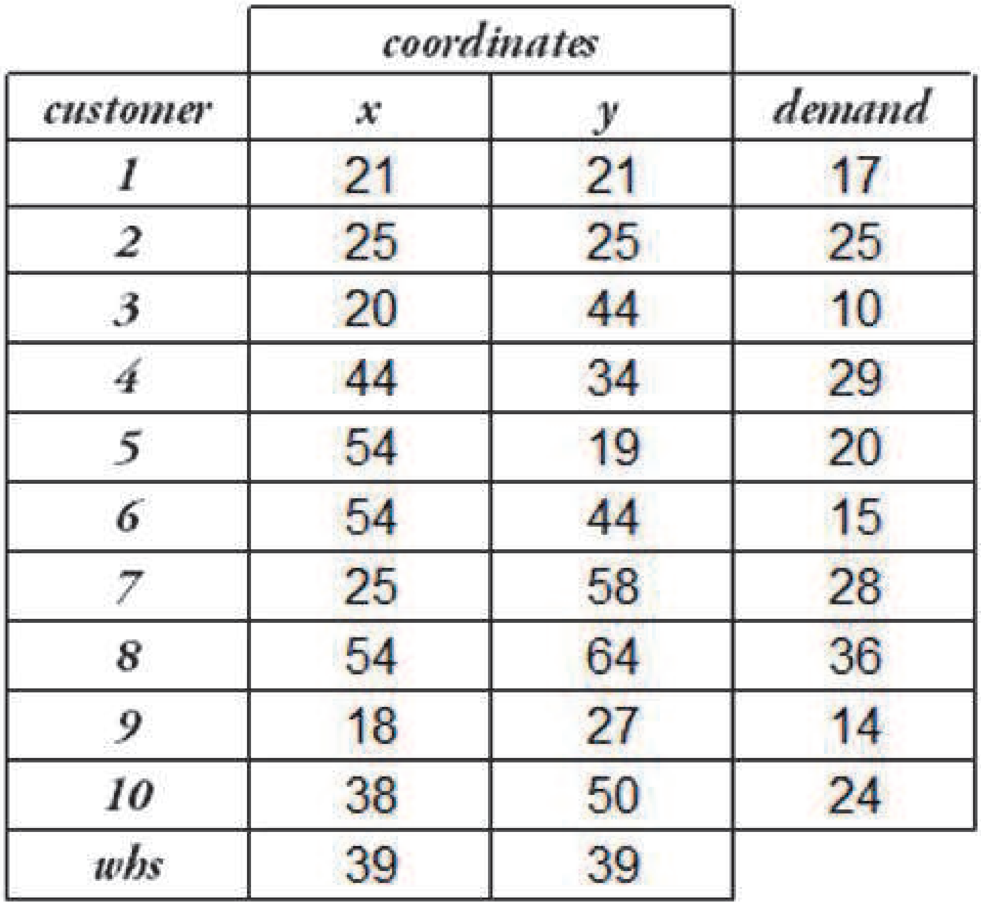
Step 1: The matrix of Euclidean distances between each pair of nodes is calculated: **Table 2** shows the distance matrix between all customers along with the warehouse. This matrix is symmetrical, that is, it has the same distance to go from client i to client j and vice versa. Point 0 has been considered for the warehouse (whs).
Step 2: Once the distance matrix is obtained, the savings are calculated. For the savings matrix, no row or column is placed for the warehouse.

For example, the savings between customer 1 and customer 2 is

$$s_{12} = c_{1bdg} + c_{bdg2} - c_{12} \tag{3}$$

$$s_{12} = 25.46 + 19.80 - 5.66 = 39 \tag{4}$$

In **Table 3** all the savings are shown; in the same way the matrix is symmetric.
Step 3: The route $(0, i, 0)$ is built for each client. In **Figure 2** each route is shown from the warehouse to each customer and back to the warehouse.
Step 4: Savings are organized from the highest to lowest.



	coordinates		
customer	x	y	demand
1	21	21	17
2	25	25	25
3	20	44	10
4	44	34	29
5	54	19	20
6	54	44	15
7	25	58	28
8	54	64	36
9	18	27	14
10	38	50	24
whs	39	39	

Table 1.
Cardinal coordinates of storage and customers with their demand.

In the list of savings to choose, only the savings that can be chosen are considered.

When the list is prepared with all the savings, those savings that one or both clients have already considered in a previous route are discarded.

Step 5: To assemble the routes, the restrictions are considered; for this example the only restriction is the capacity of the truck that does not exceed 100 units of the product x.

For the first route, the highest savings are chosen and placed in the form $(0, i, 0)$; in this case 0 is the warehouse (whs), as the savings are chosen to add them to the existing route or create a new one; the demands of each client are added, and the route is closed when the sum of the demands is equal to or less than the capacity of the truck 100 units, but when adding one more client, the demand exceeds the capacity, and you can no longer choose that customer.

The composition of the routes is displayed step by step in **Figure 3**, and the complete route diagram is shown in **Figure 4**. Below is the composition of the routes with the demands.

Route 1: whs, c1, c9, whs	Route demand 1: 17 + 14 = 31
Route 1: whs, c2, c1, c9, whs	Route demand 1: 31 + 25 = 56
Route 2: whs, c3, c7, whs	Route demand 2: 10 + 28 = 38
Route 1: whs, c2, c1, c9, c3, c7, whs	Route demand 1: 56 + 38 = 94
Route 2: whs, c6, c8, whs	Route demand 2: 15 + 36 = 51
Route 2: whs, c6, c8, c10, whs	Route demand 2: 51 + 24 = 75

Route 2: whs, c5, c6, c8, c10, whs	Route demand 2: 75 + 20 = 95
Route 3: whs, c4, whs	Route demand 3: 29

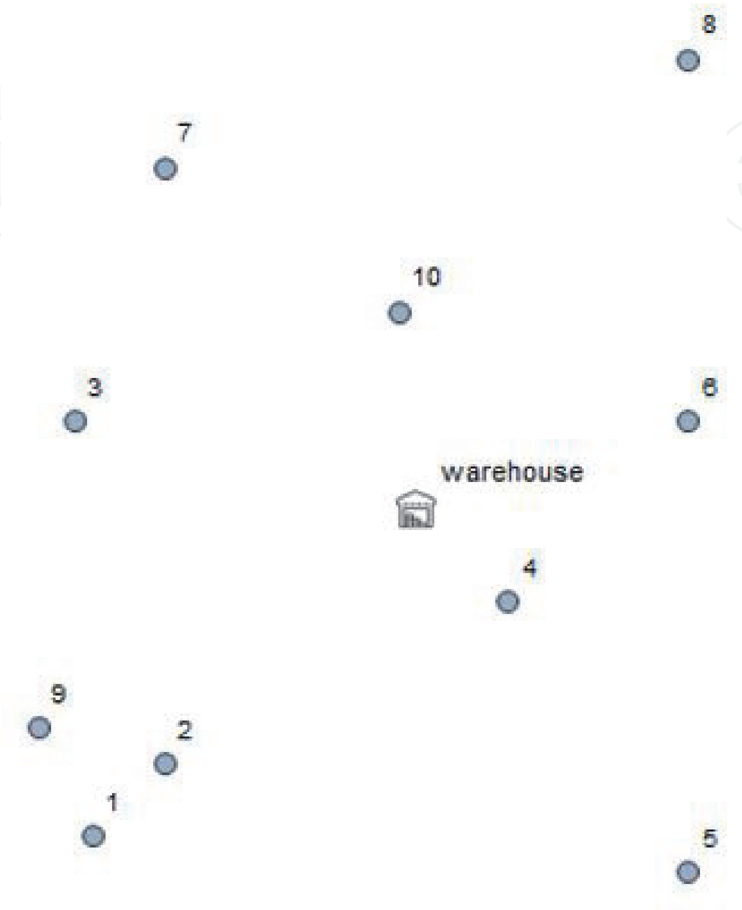


Figure 1.
Customer and warehouse positioning.

Cij	**whs**	**c1**	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**	**c9**	**c10**
whs	—	25.46	19.80	19.65	7.07	25.00	15.81	23.60	29.15	24.19	11.05
c1	25.46	—	5.66	23.02	26.42	33.06	40.22	37.22	54.20	6.71	33.62
c2	19.80	5.66	—	19.65	21.02	29.61	34.67	33.00	48.60	7.28	28.18
c3	19.65	23.02	19.65	—	26.00	42.20	34.00	14.87	39.45	17.12	18.97
c4	7.07	26.42	21.02	26.00	—	18.03	14.14	30.61	31.62	26.93	17.09
c5	25.00	33.06	29.61	42.20	18.03	—	25.00	48.60	45.00	36.88	34.89
c6	15.81	40.22	34.67	34.00	14.14	25.00	—	32.20	20.00	39.81	17.09
c7	23.60	37.22	33.00	14.87	30.61	48.60	32.20	—	29.61	31.78	15.26
c8	29.15	54.20	48.60	39.45	31.62	45.00	20.00	29.61	—	51.62	21.26
c9	24.19	6.71	7.28	17.12	26.93	36.88	39.81	31.78	51.62	—	30.48
c10	11.05	33.62	28.18	18.97	17.09	34.89	17.09	15.26	21.26	30.48	—

Table 2.
Distance matrix.

Sij	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
c1	—	39.60	22.08	6.11	17.40	1.04	11.84	0.41	42.93	2.89
c2		—	19.80	5.85	15.18	0.94	10.40	0.35	36.71	2.67
c3			—	0.72	2.44	1.46	28.38	9.36	26.72	11.72
c4				—	14.04	8.74	0.06	4.60	4.33	1.03
c5					—	15.81	0.00	9.15	12.31	1.16
c6						—	7.21	24.97	0.19	9.77
c7							—	23.14	16.01	19.38
c8								—	1.72	18.94
c9									—	4.75
c10										—

Table 3.
Savings matrix.

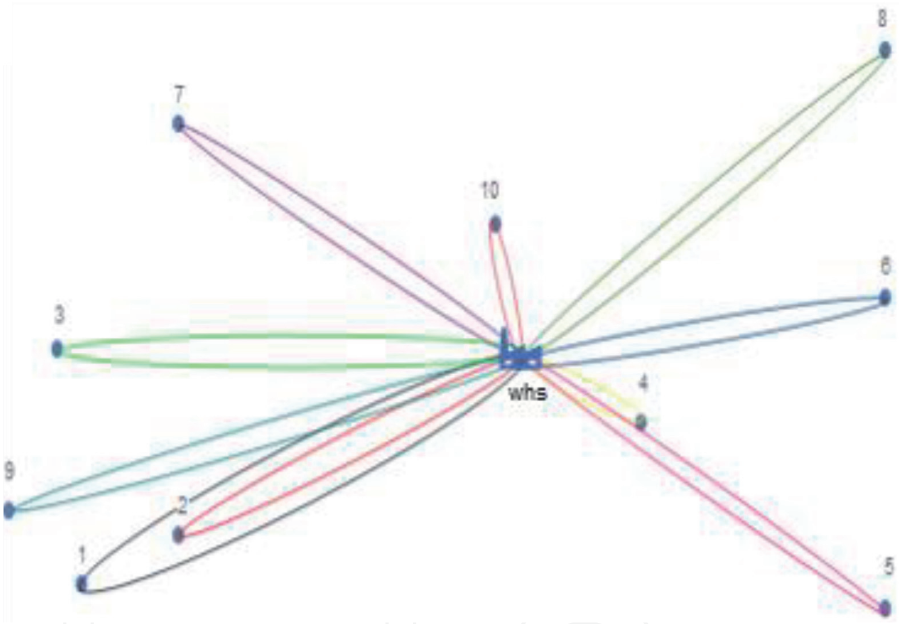


Figure 2.
Clarke & Wright heuristics step 1 route development $(0, i, 0)$.

Consequently, Clarke & Wright algorithm determines a solution for the routing problem in which the distance traveled is 204.20 units in length.

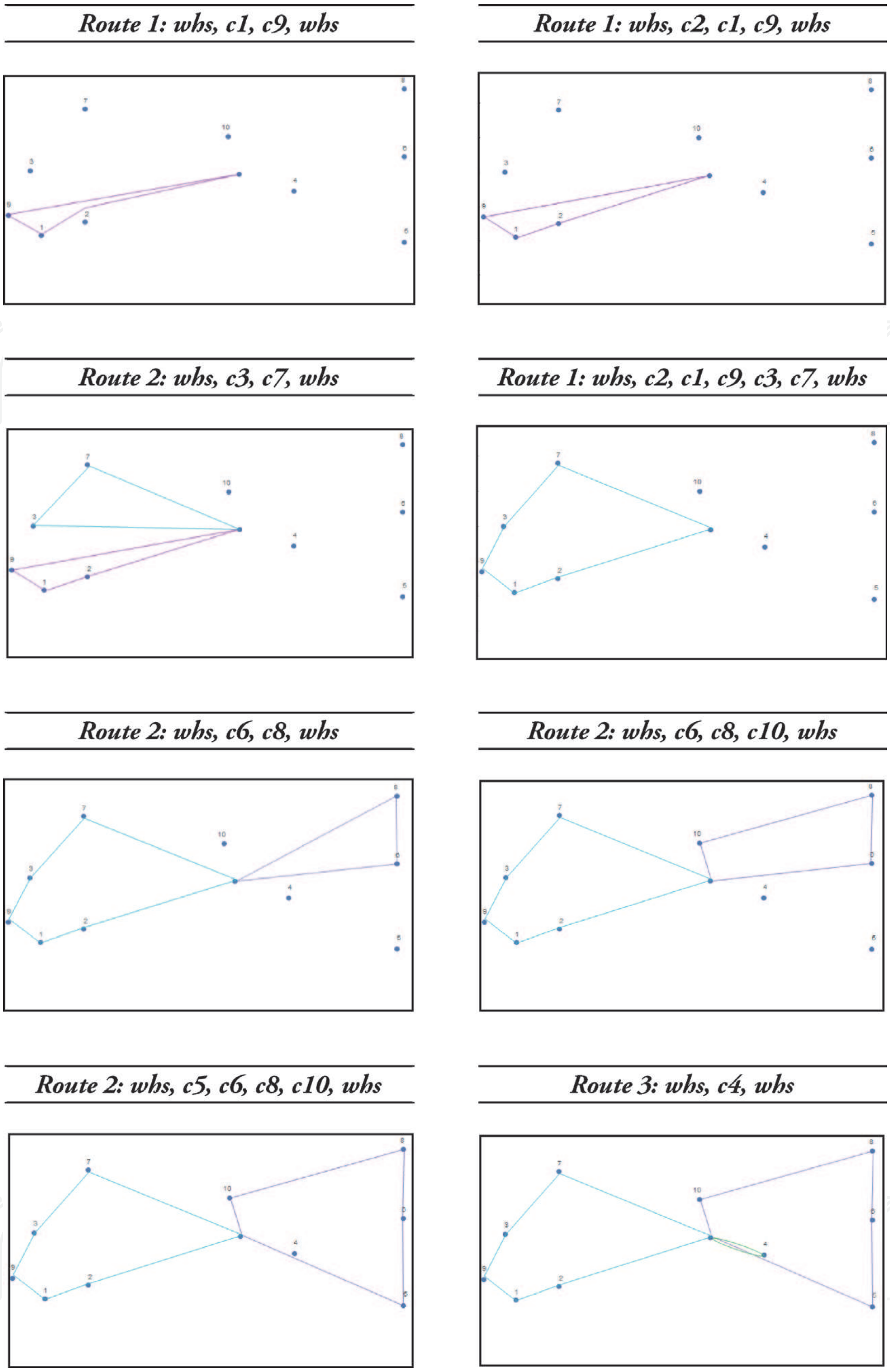
3.2 Heuristic method of assigning first, routing after

Sweep heuristics are the best-known method of assigning first, routing later.

This method is solved in two phases. First, groups of customers called clusters are created considering the capacity constraints of the vehicles, and second for each cluster, a route is generated that visits all customers.

In sweep heuristic, clusters are created by turning a half-straight in the central tank from the horizontal counterclockwise; after that the customers are incorporated into the mentioned group until the maximum capacity restriction of the vehicles is met.

This heuristic is used to find solutions to geographical problems, that is to say in which the nodes or vertices correspond to a point in the plane. It is assumed that the



Route 1: whs, c1, c9, whs

Route 1: whs, c2, c1, c9, whs

Route 2: whs, c3, c7, whs

Route 1: whs, c2, c1, c9, c3, c7, whs

Route 2: whs, c6, c8, whs

Route 2: whs, c6, c8, c10, whs

Route 2: whs, c5, c6, c8, c10, whs

Route 3: whs, c4, whs

Figure 3.
Step by step: routing of Clarke & Wright heuristics.

location of each client i can be represented through its polar coordinates (r_i, θ_i) having a single central deposit. It defines

$$\begin{gathered} \theta_i = \arctan\left[\frac{Y(i) - Y(whs)}{X(i) - X(whs)}\right] \\ where \quad -\pi < r_i < 0 \quad si \quad Y(i) - Y(whs) < 0 \\ y \;\; 0 < r_i < \pi \quad si \quad Y(i) - Y(1) \geq 0, (i = 1, 2, \dots, n) \\ r_i = polar \ radio \ coordinate \ of \ the \ i \ th \ position \ (i = 1, 2, \dots, n) \end{gathered} \tag{5}$$

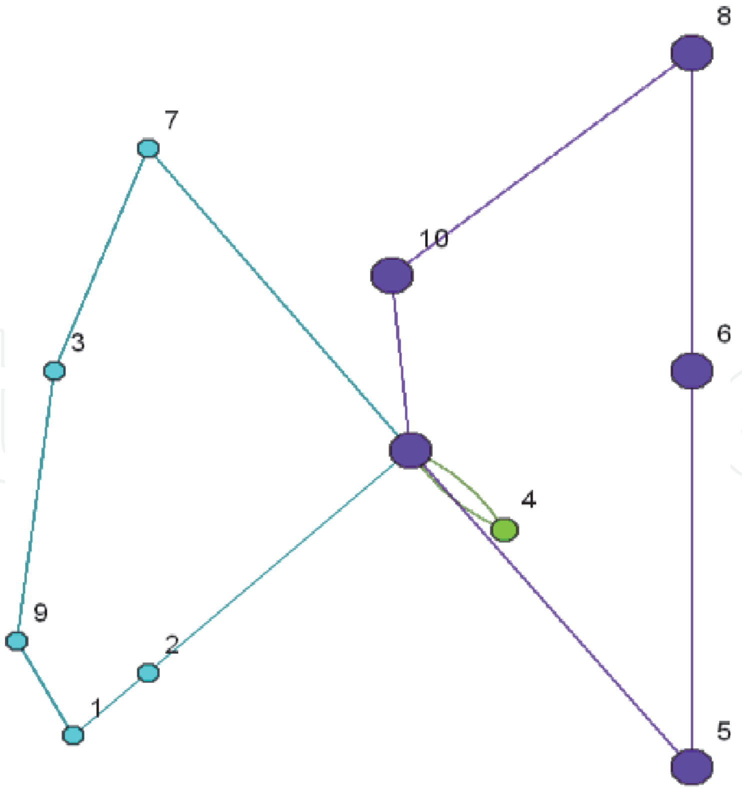


Figure 4.
Final routes by Clarke & Wright algorithm.

3.2.1 Steps for sweep heuristics

Step 1: Prepare the table of the location of the nodes in polar coordinates.
Step 2: Customers or cities are sorted in ascending order by θ; if two clients or cities have the same θ value, the one with the lowest r value is chosen first. Then a customer or city w is selected to start and make $k = 1\, y\ C_k = \{w\}$.
Step 3: If all clients or cities are in a cluster, go to step 4. Otherwise, a client or city is selected; w_i and w_i are added to C_k if you do not exceed the capacity restrictions; if you exceed them, create a new cluster for which, $k = k + 1$ and $C_k = \{w_i\}$. Repeat step 2 until there are no clients or cities without a cluster.
Step 4: For each cluster C_k for $t = 1, \dots, k$, solve traveling salesman problem (TSP) with its clients and a solution that can be a local optimum is obtained until not checking otherwise.
Step 5: Return to step 2 to reorder customers where the first becomes the last, the second the first, and so on until the original sorting. For each change, steps 3 and 4 are performed again, and the best of the solutions obtained is taken.

3.2.2 Example of sweep heuristics

We will take the example of the savings algorithm.

Step 1: Formula (5) is used to obtain the polar coordinate table, where θ_i is expressed in radians and r_i is the directed distance. **Table 4** shows the polar coordinates for each client i. The change of polar coordinates is displayed (**Figure 5**).
Step 2: It is sorted by θ_i from least to greatest, as seen in **Table 5**.
Step 3: To elaborate the routes, it is done in two phases, the first one where the clients are grouped by the sweep method and the second one where a TSP is resolved (step 4).

For the sweep method, the angles from the smallest to the largest are chosen, and it moves counterclockwise.

As can be seen in **Table 5**, customers are already sorted in ascending order by their angular polar coordinate, and customers are chosen until they fail to comply with the capacity restriction of the truck that is 100 units of product x. Considering this the routes are as follows:

Route 1: whs, c9, c2, c1, c5, whs	Route demand 1: 14+25+17+20 = 76
Route 2: whs, c4, c6, c8, whs	Route demand 2: 29+15+36 = 80
Route 3: whs, c10, c7, c3, whs	Route demand 3: 24+28+10 = 62

In **Figure 6**, the sweeps are visualized starting with client 9 that has the greatest angle, thus grouping them in zones in this case by colors and within each one for their best distance. The sweep groups customers do not violate the restriction of the truck.

In **Figure 7**, the solution is shown with three routes before the TSP is applied.

Step 4: In the second phase to each route already generated in the first, it is resolved by TSP, for this case with the nearest node.

	r_i	θ_i
c1	25.46	−2.36
c2	19.80	−2.36
c3	19.65	2.88
c4	7.07	−0.79
c5	25.00	−0.93
c6	15.81	0.32
c7	23.60	2.21
c8	29.15	1.03
c9	24.19	−2.62
c10	11.05	1.66

Table 4.
Polar coordinates for each i.

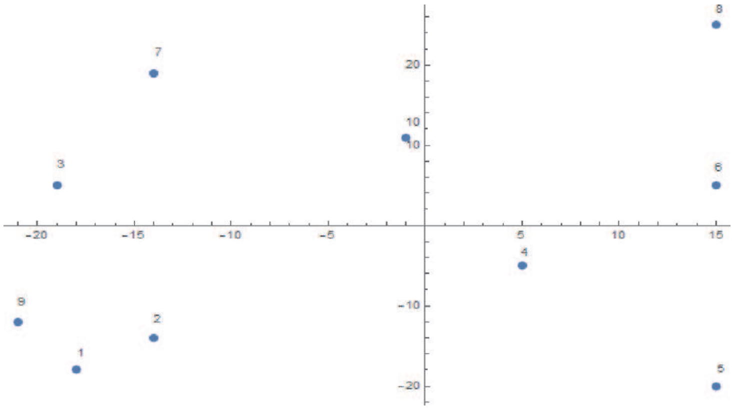


Figure 5.
Node location.

	r_i	θ_i
c9	24.19	−2.62
c2	19.80	−2.36
c1	25.46	−2.36
c5	25.00	−0.93
c4	7.07	−0.79
c6	15.81	0.32
c8	29.15	1.03
c10	11.05	1.66
c7	23.60	2.21
c3	19.65	2.88

Table 5.
Ascending ordering of customers.

The routes are as follows:

Route 1: whs, c9, c1, c2, c5, whs	Route demand 1: 14+17+25+20 = 76
Route 2: whs, c4, c6, c8, whs	Route demand 2: 29+15+36 = 80
Route 3: whs, c10, c7, c3, whs	Route demand 3: 24+28+10 = 62

Thus, the sweep algorithm has a local solution for the routing problem in which the distance traveled is 222.36 units in length.

In **Figure 8**, the route diagram is displayed.

Step 5: Repeat step 2 where the customers already ordered from **Table 5**, continue to rotate the position until the first returns to be first, and for each rotation, steps 3 and 4 are performed, and after all iterations, the best is selected.

Below is the iteration that had the best result. **Table 6** shows the fifth iteration of nine where you start with client six (**Figure 9**).

After performing step 3 in **Figure 10**, the grouping of customers is appreciated to not violate the restriction of the truck's capacity.

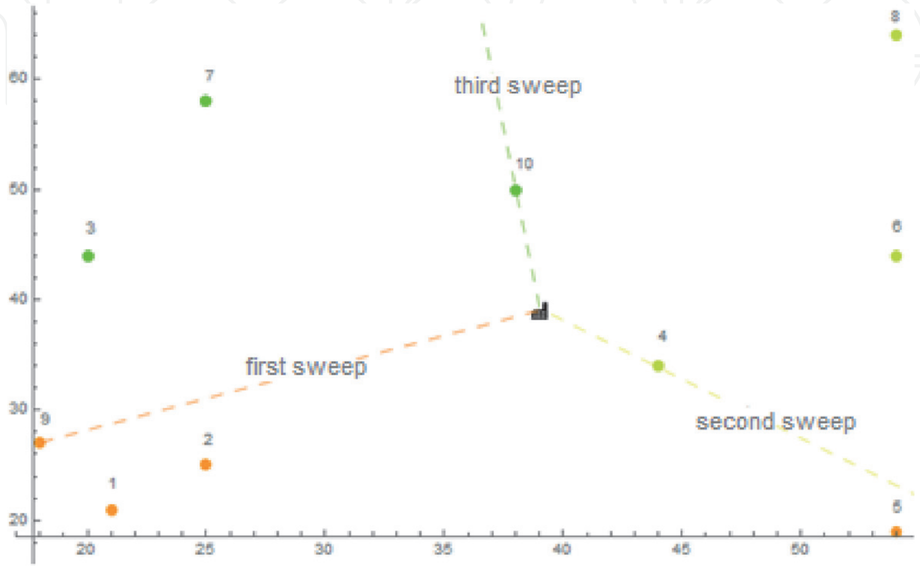


Figure 6.
First phase of sweep heuristics, grouping.

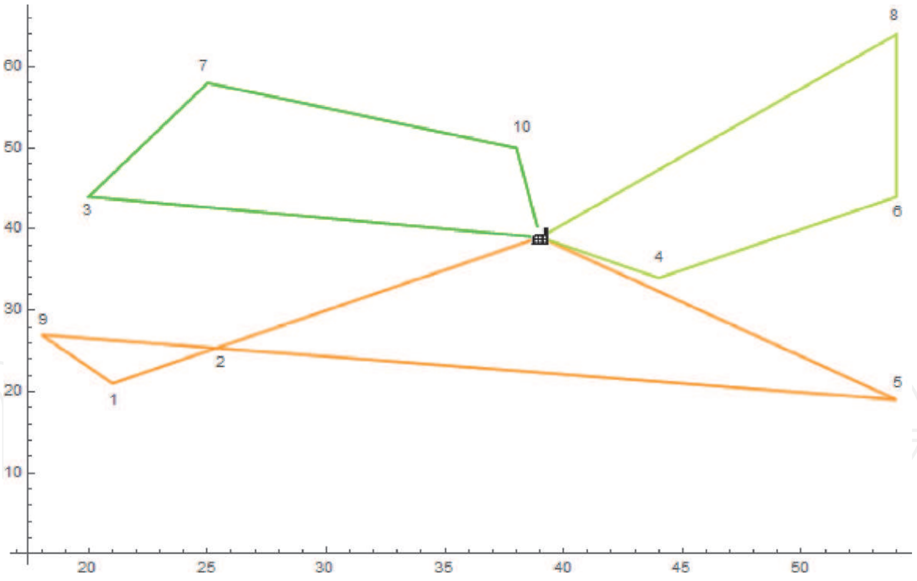


Figure 7.
First phase of sweep heuristics, solution with three routes.

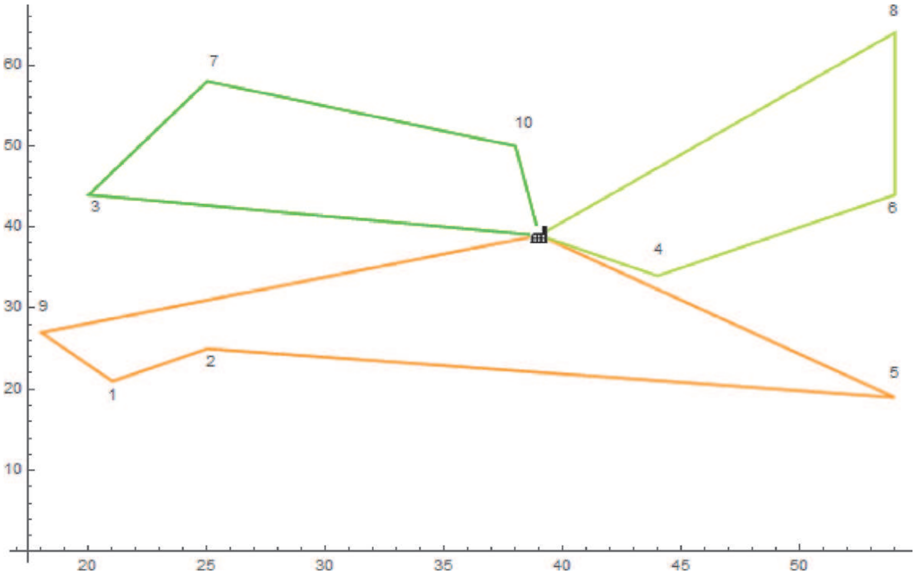


Figure 8.
Routes by sweep algorithm.

	r_i	θ_i
c6	15.81	0.32
c8	29.15	1.03
c10	11.05	1.66
c7	23.60	2.21
c3	19.65	2.88
c9	24.19	−2.62
c1	25.46	−2.36
c2	19.80	−2.36
c5	25.00	−0.93
c4	7.07	−0.79

Table 6.
Customer ordering—fifth iteration.

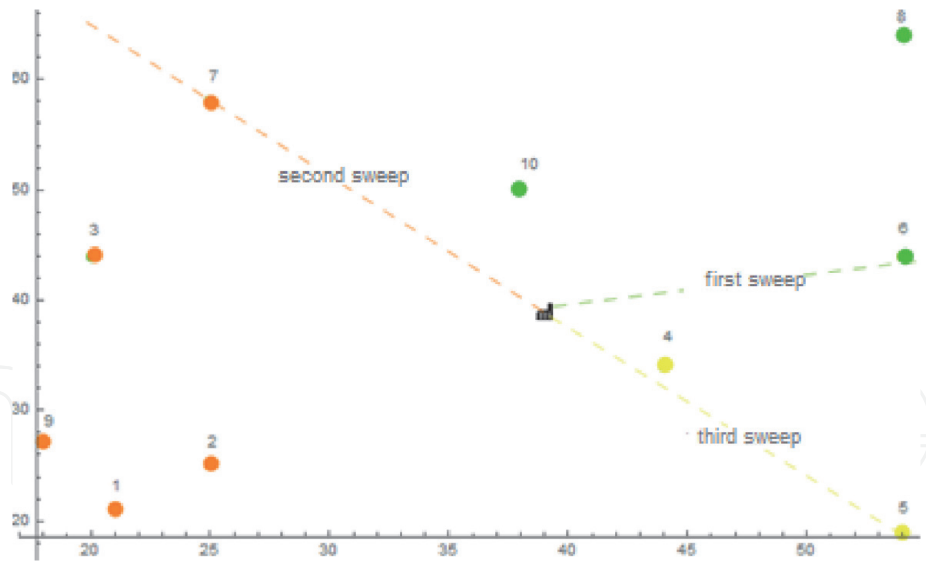


Figure 9.
First phase of sweep heuristics, grouping—fifth iteration.

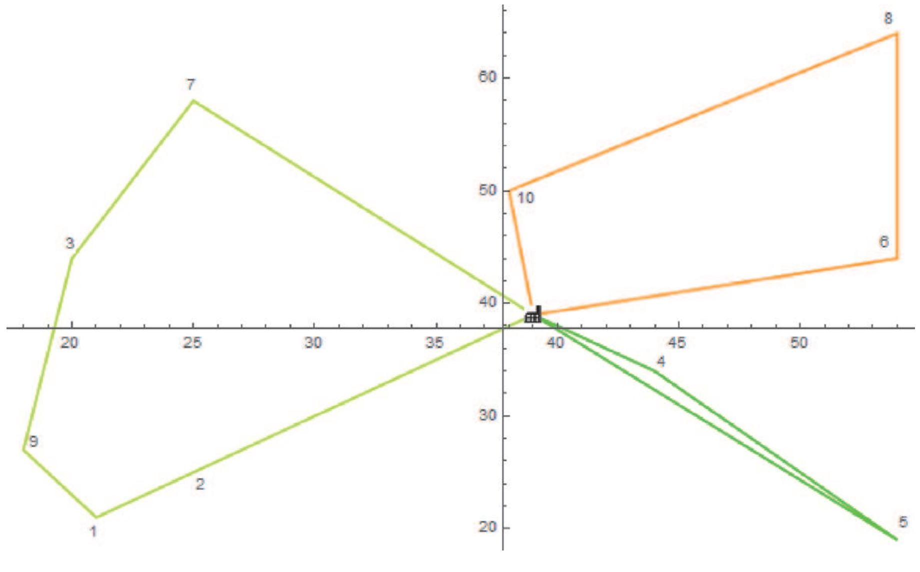


Figure 10.
Routes by sweep algorithm—fifth iteration.

In step 4, each grouping is resolved with a TSP, and the following routes are obtained:

Route 1: whs, c6, c8, c10, whs	Route demand 1: 15+36+24 = 75
Route 2: whs, c7, c3, c9, c1, c2, whs	Route demand 2: 28+10+14+25+17 = 94
Route 3: whs, c5, c4, whs	Route demand 3: 20+29 = 49

In **Figure 10**, the diagram of the routes of the fifth iteration is displayed, which obtained the best response.

The sweep algorithm determines a solution for the routing problem in which the distance traveled is 205.96 units in length, that is, a solution of lower quality than that obtained by the Clarke & Wright algorithm with 204.20 units in length.

4. Metaheuristics

The term metaheuristics first appeared in the seminal article about taboo search (Glover, 1987). The term metaheuristics is obtained by putting the suffix “meta” before the word heuristic, which means “beyond” or “at a higher level.”

Metaheuristics are generic procedures that, through approximate algorithms, guide a subordinate heuristic by combining the exploration of the solution space for optimization problems, obtaining better results than classical heuristics in a longer period; however, this time is less than if the exact methods are used.

Metaheuristics that have been considered for this comparative study are shown below, which correspond to constructive and local search procedures [6].

4.1 GRASP

GRASP methods had their origins at the end of the 1980s in order to find a solution to problems of joint coverings, and in 1995 by Feo and Resende, this metaheuristic is of general purpose [8].

The word GRASP comes from the acronym of greedy randomized adaptive search procedures that would be something like search procedures based on voracious adaptive random functions.

GRASP has a multistart process in which each step has a construction and an improvement phase. In the construction phase, the constructive heuristic process obtains a good initial solution, which is improved in the second phase by a local search algorithm. The best of all solutions examined is saved as the final result.

There are many implementations of GRASP metaheuristics, including variants and hybridizations with other procedures such as variable neighborhood search or path relinking, with which this metaheuristic has proven to work very well in practice as demonstrated in Marti and Sandoya [9]. A simple scheme to represent the operation of this algorithm is as follows:

While (stop condition)

Construction phase:

- Choose a list of candidate elements.
- Have a restricted list with the best candidates.
- Select an item randomly from the restricted list.

Improvement phase:

- Perform a local search process based on the solution built until it can no longer be improved.

Update:

- If the solution obtained improves to the best stored, update it.

In the construction phase, a possible solution is built iteratively, considering an element in each step. In each iteration the choice of the next element to be added to the partial solution is determined by a greedy function, which examines the benefit of adding each of the elements according to the objective function and choosing the best one.

This metaheuristic works with a restricted list of the best candidates, which makes the best candidate randomly selected for each iteration of the construction phase.

In the improvement phase, the results that are obtained from the construction phase are not usually local optimal; therefore, a local search procedure is applied as post-processing to perfect the solution obtained.

Performing several iterations is a way of sampling the solution space.

4.2 Simulated annealing

The simulated annealing metaheuristics was introduced in the 1950s by Metropolis Hastings to be used in the field of statistical thermodynamics simulating cooling processes of a material.

In 1983 the method was refocused to solve combinatorial optimization problems of great complexity by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi, and independently in 1985 by Vlado Cerny. For its implementation ease, this metaheuristic had a great boom in the 1980s.

Simulated annealing has its procedure based on local search by environments that is characterized by an acceptance criterion of neighboring solutions that are adapted throughout its execution.

A temperature variable is used, T, that determines the extent to which neighboring solutions, worse than the current n, can be accepted. This variable is about starting it with a high value, which is called the initial temperature, T_0, which generates a high probability of accepting a nonimprovement movement. In each iteration the temperature decreases through a temperature cooling mechanism, α, having a smaller probability until approaching the optimal solution and reaching a final temperature, T_f. Costs also decrease as the temperature decreases, making it increasingly difficult to accept bad movements in search of the solution.

In each iteration a specific number of neighbors is generated, which can be fixed for the entire execution or depend on each iteration.

Each time a neighbor is generated, the acceptance criterion is applied to see if it replaces the current solution:

- If the neighboring solution is better than the current one, it is automatically accepted, as it would be done in the classic local search.
- If the neighboring solution is worse than the current one, there is still a chance that the neighbor will replace the current solution. This allows the algorithm to exit from local optimum, in which the classic local search would be trapped.

This model is given by the following structure:

```
Take an initial solution $x$
Take an initial temperature $T$
While (not frozen)
Perform $L$ times
 Take $x$' from $N(x)$
 $d = f(x') - f(x)$
 If $(d < 0)$ do$x = x'$
 If $(d > 0)$ do$x = x'$ with $p = e^{-d/T}$
 Take action of the cooling mechanism $(T = rT)$
```

The following parameters are determined:

I. Initial temperature: it is established by doing a series of tests to reach a certain fraction of accepted movements.

II. Cooling speed r.

III. Length L that must be proportional to the expected size of $N(x)$.

IV. When the cooling sequence ends, it is frozen $cont = cont + 1$ when a temperature is completed and the percentage of movements accepted is less than $MinPercent.cont = 0$ when the best stored solution is improved.

5. Implementation of heuristics and metaheuristics for the resolution of CVRP

5.1 Test instances

The cases to be evaluated are divided into three groups classified by the type of client with 10 examples each. Next, some tables will be presented, which have the name of instance, the truck's capacity in column *cap*, the number of customers in column n, the number of vehicles to be used in column k, and the optimal solution in column *opt*.

I. Clustered clients, as shown in **Table 7**, belong to the Augerat B set in 1995 [10] and specify that the coordinates are points between [0,100] × [0,100] in the grid that are chosen to form neighborhood groups (NC) closest, where $k \leq NC - 1$. The demands have a uniform distribution $U\ (1, 30)$; however, $n/10$ was multiplied by 3.

II. Random clients, as shown in **Table 8**, belong to the Augerat set A in 1995 and specify that the coordinates are points between [0,100] × [0,100] placed randomly. The demands have a uniform distribution $U\ (1, 30)$; however, $n/10$ was multiplied by 3.

III. Clustered and random clients, as shown in **Table 9**, belong to the Augerat set X in 1995 and specify that the coordinates are points between [0,1000] x [0,1000] that are grouped and placed randomly, where k is the minimum feasible number of vehicles.

	Instances	Cap	*n*	*k*	Opt
Clustered	B-n31-k5	100	30	5	672
	B-n34-k5	100	33	5	788
	B-n35-k5	100	34	5	955
	B-n38-k6	100	37	6	805
	B-n39-k5	100	38	5	549
	B-n41-k6	100	40	6	829
	B-n43-k6	100	42	6	742
	B-n44-k7	100	43	7	909
	B-n45-k5	100	44	5	751
	B-n45-k6	100	44	6	678

Table 7.
Instances of set B.

	Instances	Cap	*n*	*k*	Opt
Random	A-n32-k5	100	31	5	784
	A-n33-k6	100	32	6	742
	A-n34-k5	100	33	5	778
	A-n36-k5	100	35	5	799
	A-n37-k5	100	36	5	669
	A-n37-k6	100	36	6	949
	A-n38-k5	100	37	5	730
	A-n39-k5	100	38	5	822
	A-n39-k6	100	38	6	831
	A-n44-k6	100	43	6	937

Table 8.
Instances of set A.

	Instances	Cap	*n*	*k*	Opt
Clustered and random	X-n101-k25	206	100	25	27,591
	X-n106-k14	600	105	14	26,362
	X-n110-k13	66	109	13	14,971
	X-n115-k10	169	114	10	12,747
	X-n120-k6	21	119	6	13,332
	X-n125-k30	188	124	30	55,539
	X-n129-k18	39	128	18	28,940
	X-n134-k13	643	133	13	10,916
	X-n139-k10	106	138	10	13,590
	X-n143-k7	1190	142	7	15,700

Table 9.
Instances of set X.

6. Results

The results for the 30 selected test cases are shown below, applying the heuristics and metaheuristics studied in Chapter 3 and 4 to know which one has a response that is closer or equal to the optimum by group of client positioning.

To define which has a better quality solution, the gap analysis or difference analysis is used, which consists in calculating the difference between the optimal solution and the solution obtained, divided for the solution obtained and expressed as a percentage.

The solution of the real case is also presented through the heuristics and metaheuristics that offer the best solution given the characteristic of the clients' positions.

6.1 Test cases resolved by heuristics

Clarke & Wright heuristics have better quality solutions, solving problems where customers with a small *n* are grouped.

Table 10 shows that for the group of clients with gathered positions, the gap is 3.63%; for the positions of random clients, the gap is 5.18%; and in less effective way for customers with grouped and random positions, it has a gap of 6.55%.

	Data					Clarke & Wright		
	Instances	**Cap**	***n***	***k***	**Opt**	***k***	**Result**	**Gap (%)**
Clustered	B-n31-k5	100	30	5	672	5	681.20	1.37
	B-n34-k5	100	33	5	788	5	794.30	0.80
	B-n35-k5	100	34	5	955	5	978.30	2.44
	B-n38-k6	100	37	6	805	6	832.10	3.37
	B-n39-k5	100	38	5	549	5	566.70	3.22
	B-n41-k6	100	40	6	829	7	898.10	8.34
	B-n43-k6	100	42	6	742	6	782.00	5.39
	B-n44-k7	100	43	7	909	7	937.70	3.16
	B-n45-k5	100	44	5	751	5	757.20	0.83
	B-n45-k6	100	44	6	678	7	727.80	7.35
	Average							3.63
Random	A-n32-k5	100	31	5	784	5	843.70	7.61
	A-n33-k6	100	32	6	742	7	776.30	4.62
	A-n34-k5	100	33	5	778	6	810.40	4.16
	A-n36-k5	100	35	5	799	5	828.50	3.69
	A-n37-k5	100	36	5	669	5	707.80	5.80
	A-n37-k6	100	36	6	949	6	976.60	2.91
	A-n38-k5	100	37	5	730	6	768.10	5.22
	A-n39-k5	100	38	5	822	5	902.00	9.73
	A-n39-k6	100	38	6	831	6	863.10	3.86
	A-n44-k6	100	43	6	937	7	976.00	4.16
	Average							5.18
Clustered and random	X-n101-k25	206	100	25	27,591	28	28940.00	4.89
	X-n106-k14	600	105	14	26,362	14	27280.00	3.48
	X-n110-k13	66	109	13	14,971	13	15870.00	6.00
	X-n115-k10	169	114	10	12,747	11	13490.00	5.83
	X-n120-k6	21	119	6	13,332	6	14540.00	9.06
	X-n125-k30	188	124	30	55,539	33	58830.00	5.93
	X-n129-k18	39	128	18	28,940	18	30300.00	4.70
	X-n134-k13	643	133	13	10,916	14	11520.00	5.53
	X-n139-k10	106	138	10	13,590	11	14530.00	6.92
	X-n143-k7	1190	142	7	15,700	7	17770.00	13.18
	Average							6.55

Table 10.
Clarke & Wright heuristics results.

It also compares the number of vehicles (k) that were obtained when solving each case against the optimal solution, and it was obtained that 12 cases had a vehicle more than the optimum B-n41-k6, B-n45-k6, A-n33-k6, A-n34-k5, A-n38-k5, A-n44-k6, X-n106-k14, X-n115-k10, X-n134-k13, and X-n139-k10, and two cases had three more than the optimal vehicles X-n101-k25 and X-n125-k30.

Sweep heuristics are more effective in solving problems where customers with a small n are grouped together. However, the difference in the average gap between random and grouped customers is short.

Table 11 shows that for the group of customers with grouped positions, the gap is 8.68%; for random customer positions; the gap is 8.85%; and in a less effective way for customers with grouped and random positions, it has a gap of 17.00%.

It also compares the number of vehicle numbers (k) that were obtained when solving each case against the optimal solution, and it was obtained that seven cases had a vehicle more than the optimal B-n45-k6, A-n38-k5, A-n39-k5, X-n115-k10, X-n129-k18, X-n134-k13, and X-n139-k10; one case had five more vehicles than the optimum X-n101-k25, and one case had six more vehicles than the optimal X-n125-k30.

In each group of clients, the sweep heuristic obtained better answers than the Clarke & Wright heuristics with 30% in the group of clients with a grouped position, 30% in the group of clients with a random position, and 20% in the group of clients with grouped and random position. In other words, Clarke & Wright heuristics are superior with 70% in the first two groups of clients and with 80% in the last group of clients.

A comparison among the values of the Distance traveled in the solution found by the heuristic, the optimal solution and the GAP for each one of the considered test instances is shown in **Table 12**.

On the other hand, **Table 13** shows a summary of the minimum, maximum and average gap for each of the three classes of problems considered: Clustered, Random and Clustered, and Random.

6.2 Test cases resolved through metaheuristics

The GRASP metaheuristics are based on a previous solution for which Clarke & Wright heuristic responses were selected since their responses are of better quality than the sweep heuristics.

The following parameters were considered for its implementation:

- α = 0.5
- Number of iterations = 10,000

GRASP's metaheuristics are more effective in solving problems where customers with a small n are grouped together.

Customers	Minimum gap (%)	Maximum gap (%)	Average gap (%)
Clustered	0.80	8.34	3.63
Random	2.91	9.73	5.18
Clustered and random	3.48	13.18	6.55

Table 11.
Clarke & Wright heuristic gaps comparison.

	Data					Sweep		
	Instances	**Cap**	***n***	***k***	**Opt**	***k***	**Result**	**Gap (%)**
Clustered	B-n31-k5	100	30	5	672	5	696.69	3.67
	B-n34-k5	100	33	5	788	5	889.51	12.88
	B-n35-k5	100	34	5	955	5	966.93	1.25
	B-n38-k6	100	37	6	805	6	838.99	4.22
	B-n39-k5	100	38	5	549	5	613.45	11.74
	B-n41-k6	100	40	6	829	6	884.53	6.70
	B-n43-k6	100	42	6	742	6	750.92	1.20
	B-n44-k7	100	43	7	909	7	1137.46	25.13
	B-n45-k5	100	44	5	751	5	836.08	11.33
	B-n45-k6	100	44	6	678	7	736.62	8.65
	Average							8.68
Random	A-n32-k5	100	31	5	784	5	885.04	12.89
	A-n33-k6	100	32	6	742	6	751.65	1.30
	A-n34-k5	100	33	5	778	5	786.44	1.08
	A-n36-k5	100	35	5	799	5	862.71	7.97
	A-n37-k5	100	36	5	669	5	736.35	10.07
	A-n37-k6	100	36	6	949	7	1087.46	14.59
	A-n38-k5	100	37	5	730	6	818.46	12.12
	A-n39-k5	100	38	5	822	5	882.53	7.36
	A-n39-k6	100	38	6	831	6	900.14	8.32
	A-n44-k6	100	43	6	937	6	1056.84	12.79
	Average							8.85
Clustered and random	X-n101-k25	206	100	25	27,591	30	34368.50	24.56
	X-n106-k14	600	105	14	26,362	14	30035.90	13.94
	X-n110-k13	66	109	13	14,971	13	15769.30	5.33
	X-n115-k10	169	114	10	12,747	11	14894.20	16.84
	X-n120-k6	21	119	6	13,332	6	14495.40	8.73
	X-n125-k30	188	124	30	55,539	36	69342.40	24.85
	X-n129-k18	39	128	18	28,940	19	36941.80	27.65
	X-n134-k13	643	133	13	10,916	14	13835.90	26.75
	X-n139-k10	106	138	10	13,590	11	14850.90	9.28
	X-n143-k7	1190	142	7	15,700	7	17593.50	12.06
	Average							17.00

Table 12.
Sweep heuristic results.

Table 14 shows that for the clients with grouped positions; the gap is 3.09%; for the positions of random clients, the gap is 4.38%; and less effectively for customers with grouped and random positions, it has a gap of 5.97%.

Customers	Minimum gap (%)	Maximum gap (%)	Average gap (%)
Clustered	1.20	25.13	8.68
Random	1.08	14.59	8.85
Clustered and random	5.33	27.65	17.00

Table 13.
Sweep heuristic gap comparison.

	Data					Grasp		
	Instances	cap	*n*	*k*	Opt	*k*	Result	Gap (%)
Clustered	B-n31-k5	100	30	5	672	5	679.05	1.05
	B-n34-k5	100	33	5	788	5	788.00	0.00
	B-n35-k5	100	34	5	955	5	968.85	1.45
	B-n38-k6	100	37	6	805	6	830.45	3.16
	B-n39-k5	100	38	5	549	5	564.85	2.89
	B-n41-k6	100	40	6	829	7	897.24	8.23
	B-n43-k6	100	42	6	742	6	777.98	4.85
	B-n44-k7	100	43	7	909	7	932.36	2.57
	B-n45-k5	100	44	5	751	5	755.23	0.56
	B-n45-k6	100	44	6	678	7	719.80	6.17
	Average							3.09
Random	A-n32-k5	100	31	5	784	5	830.67	5.95
	A-n33-k6	100	32	6	742	7	776.02	4.58
	A-n34-k5	100	33	5	778	6	809.38	4.03
	A-n36-k5	100	35	5	799	5	823.20	3.03
	A-n37-k5	100	36	5	669	5	695.42	3.95
	A-n37-k6	100	36	6	949	6	976.61	2.91
	A-n38-k5	100	37	5	730	6	765.87	4.91
	A-n39-k5	100	38	5	822	5	901.99	9.73
	A-n39-k6	100	38	6	831	6	856.93	3.12
	A-n44-k6	100	43	6	937	7	951.73	1.57
	Average							4.38
Clustered and random	X-n101-k25	206	100	25	27,591	28	28891.90	4.71
	X-n106-k14	600	105	14	26,362	14	27199.80	3.18
	X-n110-k13	66	109	13	14,971	13	15847.90	5.86
	X-n115-k10	169	114	10	12,747	11	13436.60	5.41
	X-n120-k6	21	119	6	13,332	6	14192.90	6.46
	X-n125-k30	188	124	30	55,539	33	58809.10	5.89
	X-n129-k18	39	128	18	28,940	18	30298.40	4.69
	X-n134-k13	643	133	13	10,916	14	11492.20	5.28
	X-n139-k10	106	138	10	13,590	11	14521.10	6.85
	X-n143-k7	1190	142	7	15,700	7	17491.80	11.41
	Average							5.97

Table 14.
GRASP metaheuristic results.

On average metaheuristic GRASP based is better than Clarke & Wright heuristics by 0.53%, 0.77%, and 0.38% in the solutions of the positions of the grouped, random, and grouped and random clients, the group of clients with random positions being those that obtained a greater improvement in the quality of the solutions.

	Data					**Simulated annealing**		
	Instances	**Cap**	***n***	***k***	**Opt**	***k***	**Result**	**Gap (%)**
Clustered	B-n31-k5	100	30	5	672	5	681.20	1.37
	B-n34-k5	100	33	5	788	5	793.20	0.66
	B-n35-k5	100	34	5	955	5	959.50	0.47
	B-n38-k6	100	37	6	805	6	819.50	1.80
	B-n39-k5	100	38	5	549	5	565.00	2.91
	B-n41-k6	100	40	6	829	7	897.00	8.20
	B-n43-k6	100	42	6	742	6	778.60	4.93
	B-n44-k7	100	43	7	909	7	937.30	3.11
	B-n45-k5	100	44	5	751	5	756.20	0.69
	B-n45-k6	100	44	6	678	7	726.16	7.10
	Average							3.13
Random	A-n32-k5	100	31	5	784	5	830.70	5.96
	A-n33-k6	100	32	6	742	7	776.30	4.62
	A-n34-k5	100	33	5	778	6	810.40	4.16
	A-n36-k5	100	35	5	799	5	828.50	3.69
	A-n37-k5	100	36	5	669	5	695.00	3.89
	A-n37-k6	100	36	6	949	6	976.60	2.91
	A-n38-k5	100	37	5	730	6	762.00	4.38
	A-n39-k5	100	38	5	822	5	888.60	8.10
	A-n39-k6	100	38	6	831	6	856.90	3.12
	A-n44-k6	100	43	6	937	7	967.60	3.27
	Average							4.41
Clustered and random	X-n101-k25	206	100	25	27,591	28	28850.00	4.56
	X-n106-k14	600	105	14	26,362	14	27240.00	3.33
	X-n110-k13	66	109	13	14,971	13	15790.00	5.47
	X-n115-k10	169	114	10	12,747	11	13480.00	5.75
	X-n120-k6	21	119	6	13,332	6	14420.00	8.16
	X-n125-k30	188	124	30	55,539	33	58790.00	5.85
	X-n129-k18	39	128	18	28,940	18	30300.00	4.70
	X-n134-k13	643	133	13	10,916	14	11500.00	5.35
	X-n139-k10	106	138	10	13,590	11	14530.00	6.92
	X-n143-k7	1190	142	7	15,700	7	17670.00	12.55
	Average							6.26

Table 15.
Results of simulated annealing metaheuristics.

The simulated annealing metaheuristics start from a previous solution for which Clarke & Wright heuristic responses were selected since their responses are of better quality than the sweep heuristics.

The following parameters were considered for its implementation:

- Current temperature = 250
- Final temperature = 10
- Cooling coefficient = 0.8
- Number of iterations = 10,000

The simulated annealing metaheuristic is more effective in solving problems where customers with a small *n* are grouped together.

Table 15 and **17** shows that for clients with grouped positions, the gap is 3.13%; for the positions of random clients, the gap is 4.41%; and less effectively for clients with grouped and random positions, it has a gap of 6.26%.

On average simulated annealing heuristics based is better than Clarke & Wright heuristics by 0.52%, 0.71%, and 0.21% in the solutions of grouped, random, and grouped and random clients' positions, the group of clients with random positions being those that obtained a greater improvement in the solutions quality.

Within each group of clients, the simulated annealing metaheuristics obtained better answers than the GRASP metaheuristics with 30% in the group of clients with a grouped position, 50% in the group of clients with a random position, and 40% in the group of clients with grouped and random position. That is, the GRASP metaheuristic is superior with 70% in the first group and with 60% in the third group of clients and is equal with 50% in the second group of clients.

A comparison between the minimum and maximum gap within each group is established in **Table 16** test results with clustered clients have better results. Therefore, obtaining a minimum gap of 0%, that is, in the case of B-n34-k5, the GRASP metaheuristic obtained the optimal solution (**Table 17**).

Customers	Minimum gap (%)	Maximum gap (%)	Average gap (%)
Clustered	0.00	8.23	3.09
Random	1.57	9.73	4.38
Clustered and random	3.18	11.41	5.97

Table 16.
GRASP metaheuristic gap comparison.

Customers	Minimum gap (%)	Maximum gap (%)	Average gap (%)
Clustered	0.47	8.20	3.13
Random	2.91	8.10	4.41
Clustered and random	3.33	12.55	6.26

Table 17.
Simulated annealing metaheuristics gap comparison.

7. Conclusions

The results obtained by solving the test cases by heuristics and metaheuristics show both generate better quality solutions when solving cases where customers are grouped together and had their worst result in the group of clients with a grouped and random position since they had large n with reference to the other groups of clients. Between the heuristics, Clarke & Wright heuristics generated better quality results than sweep heuristics, having a big difference in the maximum gaps of each one for each group of clients. However, the numbers of vehicles obtained in the solutions of both heuristics were compared with the optimal solution, and the sweep heuristics had more solutions in which it reached the optimum. This is a very important contribution, since Clarke & Wright heuristics get solutions with shorter distances than the sweep heuristic, but this gets greater distances with fewer vehicle units. For some companies it will be more important to reduce the units to buy than the distance traveled.

Analyzing the metaheuristics, the GRASP metaheuristics generated better quality results than simulated annealing metaheuristics, with minimal differences in average gap for each group of clients, and both metaheuristics obtained greater improvements in relation to the initial solutions of Clarke & Wright heuristics in the test cases of randomized clients. Also GRASP algorithm with B-n34-k5 case of grouped customers reached the optimal solution, being the only test instance of the thirty that were done.

It is recommended that for future studies, each group of clients by positioning has a number of clients (n) with greater variability to be able to deduce exactly if heuristics and metaheuristics have better or worse solutions when n are larger or smaller.

Author details

Fernando Francisco Sandoya Sánchez, Carmen Andrea Letamendi Lazo*
and Fanny Yamel Sanabria Quiñónez
Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil, Ecuador

*Address all correspondence to: cletamendi@gmail.com

References

[1] Ballou, RH. Business Logistics/ Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain. Upper Saddle River, New Jersey: Pearson/Prentice Hall; 2004 (Print)

[2] Toth P, Vigo D, editors. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics; Jan 2002. Available from: http://dx.doi.org/10.1137/1.9780898718515

[3] Braekers K, Ramaekers K, Van Nieuwenhuysec I. The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering. 2016;**99**:300-313

[4] Safra S, Schwartz O. On the complexity of approximating TSP with neighborhoods and related problems. Computational Complexity. 2005;**14**: 281-307

[5] Abdel-Basset M, Abdel-Fatah L, Kumar Sangaiah A. Metaheuristic algorithms: A comprehensive review. Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications. 2018:185-231

[6] Sandoya F, Martinez-Gavara A, Aceves R, Duarte A, Martı R. Diversity and equity models. In: Martí R, Panos P, Resende M, editors. Handbook of Heuristics. Cham: Springer; 2015. pp. 1-20

[7] Altınel I, Öncan T. A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. Journal of the Operational Research Society. 2005; **56**(8):954-961

[8] Resende M, Ribeiro C. Optimization by GRASP. Greedy Randomized Adaptive Search Procedures. New York, NY: Springer; 2016

[9] Marti R, Sandoya F. GRASP and path relinking for the equitable dispersion problem. Computers & Operations Research. 2013;**40**(12):3091-3099

[10] Augerat. Capacitated VRP Instances. 2013. Available from: http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/ [Accessed: 12 September 2019]