A Constructive Algorithm for the Split Delivery Vehicle Routing Problem (SDVRP)

Francisco Jones, Rodrigo Astudillo, Benjamín Acosta, Alexis Olmedo, Alejandro Córdova, and Gustavo Gatica(✉)

Faculty of Engineering - CIS, Universidad Andres Bello, Santiago, Chile
ggatica@unab.cl

Abstract. Vehicle routing is a classic of operations research. The objective is to contribute with heuristic knowledge emphasizing the complexity of the divided delivery vehicle problem. A homogeneous vehicle fleet is considered, where the demand of all clients must be covered, being able to visit clients more than once. The preliminary results are encouraging because they allow the identification of several lines of research regarding implementing practical solutions for last-mile logistics problems. Furthermore, when comparing the response times of the heuristics (0.043 s.) with the LocalSolver application (1 s and 5 s), managing to solve large instances, constantly navigating in the space of feasible solutions.

Keywords: Vehicle Routing Problem · Split Delivery Vehicle Routing · Problem · Heuristic · Representation Problem

1 Introduction

Minimizing costs is a priority for companies, especially regarding operations associated with shipping goods to many clients. However, to achieve the goal of cost reduction, additional constraints must be considered [3]. Therefore, in recent years, there has been an increasing trend to generalize classical problems, such as the Traveling Salesman Problem (TSP), to incorporate more variables and parameters to address real-world scenarios [15].

This trend has led to vehicle routing issues that consider several factors. Some focus on time windows, such as the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) [15]. Others focus on vehicle capacity, such as the Capable Vehicle Routing Problem (CVRP) [5]. In addition, there is the split-delivery vehicle routing problem (SD-VRP), which aims to perform split deliveries as long as there is remaining demand to be met and vehicle capacity allows it [3].

The goal is to find the optimal route to visit each client with specific demand requirements to minimize the cost of travel along the route. Efficient utilization of space within a vehicle reduces volume and weight, leading to significant cost savings in the industry. Dror and Trudeau [9] showed that by implementing split deliveries and fully utilizing the capacity of a vehicle, the number of delivery vehicles needed can be reduced while still meeting total demand. Among the various variants of the original Vehicle Routing

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 245–259, 2023.
https://doi.org/10.1007/978-3-031-36805-9_17

Problem (VRP), the Split Delivery Vehicle Routing Problem (SD-VRP) is particularly challenging, with NP-Hard complexity [8]. This means that finding an optimal solution requires significant processing time. It should be noted that the SD-VRP is a deterministic model since its input parameters and initial conditions are known and do not involve randomness in data processing [3].

The SD-VRP originates as a generalization of the original VRP, allowing multiple visits to a client by splitting their demand, which is then delivered by different vehicles along separate routes. The goal is to meet client demand and maximize the capacity of each vehicle. Dror and Trudeau proposed this generalization, offering a new demand-driven perspective to demonstrate potential savings in travel distances and the number of cars needed to meet established demand [8].

Various approaches have been proposed in the academic literature to address the SDVRP. These approaches include genetic algorithms, tabu searches, ant colony algorithms, and combinatorial optimization methods. These approaches have proven effective in resolving SDVRP instances and have provided quality solutions in a reasonable amount of time.

To effectively solve SD-VRP from a software development perspective, the goal is to develop a heuristic solution capable of delivering solutions within a feasible zone for SD-VRP.

In conclusion, the split delivery vehicle routing problem presents complex challenges for minimizing costs and optimizing business delivery operations. However, by considering split deliveries and efficient use of vehicle capacities, significant cost savings can be achieved. This document aims to develop a heuristic solution that addresses SD-VRP from a software development perspective, providing feasible solutions to navigate the problem's constraints. This will be compared with a standard application known as a local solver to compare results, especially computational time.

1.1 SDVRP

As demonstrated by Dror and Trudeau [9], the application of split delivery, utilizing the total capacity of a vehicle, effectively reduces the number of delivery vehicles required to fulfill the entire demand. This approach ensures efficient coverage of clients while optimizing resource utilization.

It is important to note that the Split Delivery Vehicle Routing Problem (SD-VRP) is characterized as a deterministic model. This deterministic nature arises from the known input parameters and initial conditions, which do not introduce random values during the data processing stage. These characteristic guarantees consistency and predictability throughout the problem-solving process.

Dror and Trudeau [8] provide a fresh perspective from the demand side, illustrating the potential travel savings and reduced vehicle requirements associated with this delivery approach. In addition, their research highlights the benefits of considering split delivery strategies to optimize travel costs and enhance overall operational efficiency.

In formulating the mathematical model for the SD-VRP [3], the following variables should be taken into consideration:

- C_{ij} : is the cost of the edge (i, j).

- $X_{ij}^{v} \in \{0, 1\}$: is a binary variable that takes the value 1 if vehicle v travels directly from i to j, and 0 otherwise.
- $Y_{iv} \geq 0$: is the amount of demand i delivered by vehicle v.

Function Objective

$$Min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{v=1}^{m} C_{ij} X_{ij}^{v} \tag{1}$$

Constraints

$$\sum_{i=0}^{n} \sum_{v=1}^{m} X_{ij}^{v} \geq 1, j \;\; 0, \ldots, n \tag{2}$$

$$\sum_{i=0}^{n} X_{ip}^{v} - \sum_{j=0}^{n} X_{pj}^{v} = 0 \;\; p = 0, \ldots, n; v = 1, \ldots, m \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} X_{ij}^{v} \leq |S| - 1 \;\; v = 1, \ldots, m; S \subseteq V - \{0\} \tag{4}$$

$$y_{iv} \leq d_i \sum_{j=0}^{n} X_{ij}^{v} \;\; i = 1, \ldots, n; \;\; v = 1, \ldots, m \tag{5}$$

$$\sum_{v=1}^{m} y_{iv} = d_i \;\; i = 1, \ldots, n \tag{6}$$

$$\sum_{i=1}^{m} y_{iv} \leq d_i \;\; v = 1, \ldots, m \tag{7}$$

$$x_{ij}^{v} \in \{0, 1\} i = 0, \ldots, n; \;\; j = 0, \ldots, n; \;\; v1, \ldots, m \tag{8}$$

$$y_{iv} \geq 0 \;\; i = 1, \ldots, n; \;\; v = 1, \ldots, m \tag{9}$$

Constraint (3) ensures that each client is visited at least once, (4) seeks the conservation of flow, and the vehicle must follow the route without stopping at a single client. (5) eliminates the subtours, avoiding the creation of island routes; this means not considering the depot within the route but only generating routes with clients. (6) ensures that client i is served by truck v only if v passes through i. (7) seeks to satisfy the total demand of each client. Finally, (8) ensures that the quantity delivered by each truck does not exceed its capacity.

An SDVRP is defined [3] as a pairwise undirected Graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. Where a set of vertices $\mathbf{V} = \{\mathbf{0}, \mathbf{1}, \ldots, \boldsymbol{n}\}$, which contemplates from 0 to the nth value, where 0 represents the deposit, and the other vertices represent the clients, Each Client $\boldsymbol{i} \in \boldsymbol{V}$ - $\{\mathbf{0}\}$ is associated with a demand $\boldsymbol{d_i}$.

The set of Edges where an edge (,) $\in$ has a cost. $\boldsymbol{C_{ij}} > 0$**.** An unbounded fleet of vehicles with capacity $\boldsymbol{Q} > 0$ each and a lower bound of vehicles sufficient to satisfy $\boldsymbol{d_t}$ is given by $\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \frac{\boldsymbol{d_i}}{\boldsymbol{Q}}$.

2 Literature Review

The concept of routing problems can be traced back to the Travelling Salesman Problem (TSP) [7]. This problem emerged from the need to find the shortest route when visiting a specific number of cities, starting and ending at the same origin city, with known distances between cities. The TSP sparked significant interest in the field of mathematics and optimization. In 1950, the RAND Corporation organized a contest challenging participant to find an optimal tour for visiting the capitals of the United States. In 1954, three mathematical optimization specialists achieved the first exact solution [7] for this instance involving 49 cities, guaranteeing the optimal value.

Considering the problem proposed in the 50s, where many cities need to be visited, and the goal is to find the optimal travel value, finding a feasible solution by sequentially visiting cities may seem straightforward. However, the challenge lies in selecting the method or strategy for choosing the cities due to the numerous possible routes. This is why the TSP, its variant VRP, and its different generalizations are classified as NP-Hard. In [11], a TSP and a VRP exemplify two different methods of visiting the same number of clients. The TSP focuses on optimizing a single route without considering vehicles or constraints such as capacity. On the other hand, the Vehicle Routing Problem (VRP) feels like a limited and homogeneous fleet of vehicles with a maximum capacity to meet client demand, aiming to optimize the routes generated [11].

The Split Delivery Vehicle Routing Problem (SDVRP) introduces specific conditions that must be met for a solution to be considered optimal or feasible, depending on the problem-solving perspective.

Based on the premise of the SDVRP, the following conditions must be satisfied:

Only one depot is considered: The demand to be distributed and the vehicles allocated for this task originate from a single depot.

Each client must be visited at least once: For a client to be considered service, the delivery vehicle must pass by the client's location at least once. This problem allows for multiple visits if the client's complete demand still needs to be met.

Homogeneous load is assumed: The demands in terms of volume and value are equal for all clients, disregarding individual variations. Therefore, they occupy the same space within the vehicles.

Each vehicle must adhere to its capacity without exceeding it Fig. 1 illustrates how the SDVRP can be solved through a split distribution of the demand among the vehicles if there is pending demand that can be fully satisfied and the delivery vehicles do not exceed their maximum capacity while traveling these routes [8].

Various techniques and mathematical methods have recently been proposed as new approaches to solving routing problems. Split partitioning has received attention and has been tackled through different solutions. For instance, Belenguer implemented an algorithm based on cutting planes [6]. Arquetti, on the other hand, has proposed several resolution methods, including the three-phase tabu search and a branch-and-price-and-cut focused algorithm [2, 4]. Jin has put forward an optimal solution algorithm [10]. Alemán utilized the local search method to create an adaptive memory algorithm [1]. Wilcker and Cavalier recently developed a constructive procedure specifically for the SD-VRP [16].

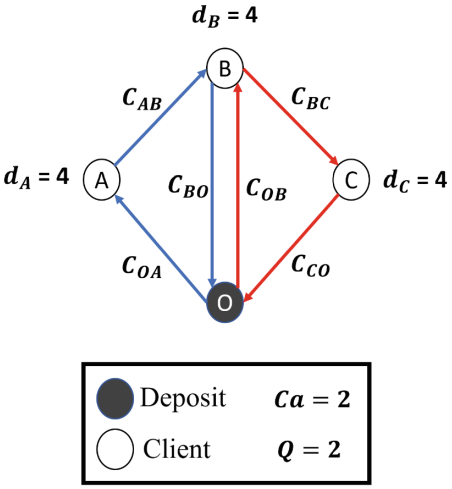

Fig. 1. Own illustration of an example for SD-VRP

3 Methodology

The methodology employed in this study comprised four main tasks, each aimed at minimizing the complexity of the process. These tasks are described below:

1. **Analysis and interpretation:** To comprehend the problem at hand, it was necessary to analyze both its definition and the corresponding mathematical model. This involved abstracting the figurative elements and devising a solution that operates within a feasible framework.
2. **Solution development:** The initial step involved developing a Python script based on the fundamental principles of abstract data structures. Three criteria were implemented, and an optimization algorithm was applied to enhance the obtained results.
3. **Validation**: After completing the development of the Python script, various tests were conducted using different instances, thereby expanding the dataset pertaining to the clients. During the validation stage, these solution paths were examined and evaluated against the problem's constraints and requirements to determine their feasibility as solutions for the SD-VRP.
4. **Comparative analysis:** Once feasible solutions were identified, a similar algorithmic proposal was studied, focusing on creating an approach rooted in mathematical programming modeling. This approach allowed for multiple iterations as the processing time increased. A quantitative comparative analysis was conducted, considering variables that accurately represented the obtained results.

3.1 Variable Storage Design

To ensure the proposed representation, careful consideration was given to the internal structure of arrays. Therefore, in this initial stage, the internal structure for storing

information was defined in a structured manner, facilitating its management within the arrays.

3.2 Storage of Variables I (Client/depot)

By establishing the repository (Fig. 2) and compiling a list of clients, a simple linked list containing all the clients (Fig. 3) present in the given instance was generated based on the data read. Additionally, the repository could be stored in a separate node, isolated from the client list.

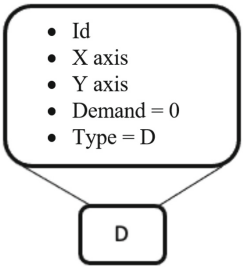

Fig. 2. Illustration of a node (reservoir).

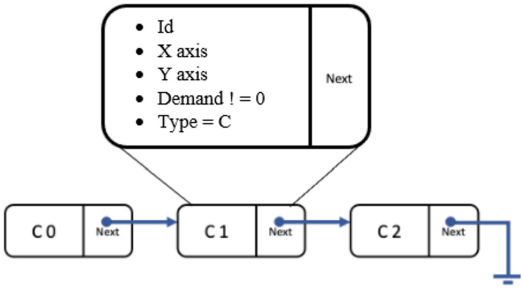

Fig. 3. Simple linked list (clients)

3.3 Variable Storage II (Vehicle)

In the second design phase, our objective is to create a storage system that effectively manages the information used in the support structures (Fig. 4). This design aims to capture and store crucial details related to vehicles, including the total number of clients within a route, total demand of the course, the average direction of the road and total distance to be traveled within the same way.

3.4 Solution Design

The solution design considers the coordinates of both the warehouse and the clients to establish a visitation order. The warehouse's coordinates serve as the reference point

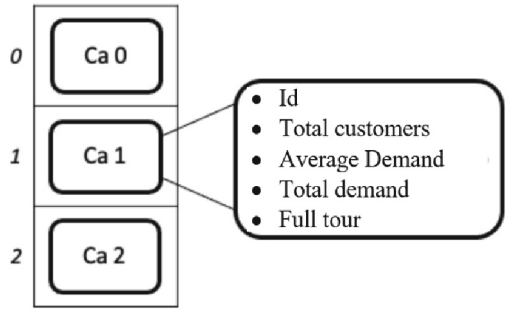

Fig. 4. Variable storage II (vehicle), prepared by the company.

(OX, OY) within a 2D plane, and the design divides the plane into four quadrants as follows:

- Quadrant 1: The X-coordinate of the client is greater than that of the warehouse, and the Y-coordinate of the client is greater than that of the warehouse.
- Quadrant 2: The X-coordinate of the client is greater than that of the warehouse, but the Y-coordinate of the client is less than that of the warehouse.
- Quadrant 3: The X-coordinate of the client is less than that of the warehouse, and the Y-coordinate of the client is also less than that of the warehouse.
- Quadrant 4: The X-coordinate of the client is less than that of the warehouse, but the Y-coordinate of the client is greater than that of the warehouse.

In addition to organizing the clients by quadrants, ordering criteria are applied to determine the visitation sequence within each quadrant. The ordering criteria are as follows:

- Criterion A: Clients within a quadrant are sorted based on their proximity to the axis intersecting the quadrant in a clockwise direction. For example, in quadrant 1, the X-axis is crossed, so the clients are ordered based on their proximity to this axis, considering the value of the Y-axis. The same logic applies to the other quadrants, with the ordering based on the proximity to the corresponding axis.
- Criterion B: This criterion follows a counterclockwise path. In quadrant 1, for instance, the counterclockwise path intersects the Y-axis, so the clients are ordered based on their proximity to this axis, considering the value of the X-axis. The same principle applies to the other quadrants, with the ordering determined by the proximity to the corresponding axis.
- Criterion C: Unlike the previous criteria, this criterion calculates the distance between the warehouse and each client while considering the defined quadrants. The length is calculated using Euclidean space, and the clients are ordered from the most minor to the most considerable distance.

Simple linked lists are created for each criterion for each quadrant, storing the clients in the respective order. To facilitate the subsequent process of assigning clients to routes associated with specific vehicles, these lists from all four quadrants are combined into a single index per criterion.

The process of assembling the routes follows these steps:

- Iterate through the list of vehicles while traversing the unified list based on the chosen criterion.
- For each vehicle, check its availability and add each client's demands to the vehicle as long as the availability exceeds the demand. The information in the node storing the vehicle's data is updated accordingly.
- As client demands are associated with a particular vehicle, the clients are inserted into a circular linked list, representing the structure that will store the solution routes (Fig. 5). It is important to note that both the first and last nodes in this list correspond to the warehouse.
- In cases where a client's demand exceeds the vehicle's availability, the market is divided into two parts. The first part equals the remaining availability to be added to the vehicle. At the same time, the total demand of the client is updated to reflect the remaining value, representing the remaining unsatisfied demand that will be addressed by the next vehicle. Once the maximum capacity of a vehicle is reached, this process is repeated with the next vehicle in the list.
- This process is repeated for each vehicle in the list, updating the data for each vehicle (number of clients to deliver, average demand, total demand, and total distance to each client) and the list containing the solution routes.

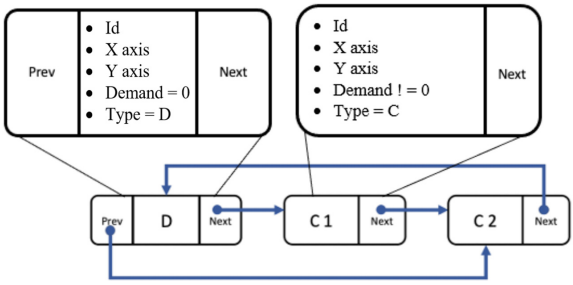

Fig. 5. Circular linked list (solution routes).

3.5 Rectification of Routes

To enhance the results and prevent arcs from intersecting within the same routes, an additional measure was implemented alongside the proposed structure. A heuristic known as 2-OPT was employed for this purpose. By utilizing an arc matrix, this heuristic examines the angles and rectifies them by changing the visiting order of clients within each individual route. It is important to note that these rectifications are applied to each course separately to avoid merging clients from different routes. This approach reduces the processing time of the algorithm and minimizes the number of clients to be processed.

The 2-OPT algorithm, as described and implemented in the document (Slootbeek, Average-Case Analysis of the), employs nested loops within another loop, resulting in an efficiency of order $\theta(n2)$ [14].

The rectification of initial solutions generated by the proposed structures significantly improves the routes by preventing trucks from traversing intersections in a similar manner.

3.6 Comparative Analysis

To conduct a comprehensive comparative analysis, it is crucial to have a clear understanding of the algorithm against which the heuristic solution is being compared.

In the research process, a comparative analysis is performed by exploring other existing algorithms that can address routing problems with similar approaches. Various programming language APIs, such as Java, C++, and Python, are considered, which provide solutions for routing problems like CVRP (Capacitated Vehicle Routing Problem), CVRPTW (Vehicle Routing Problem with Time Windows), TSP (Travelling Salesman Problem), SDVRP (Split Delivery Vehicle Routing Problem), and others [12].

The selected algorithm for this analysis can deliver feasible solutions that closely approximate an exact solution. As stated in its documentation, the algorithm is based on mathematical programming models, utilizing decision variables, constraints, and objectives [13]. Specifically, for the SDVRP, the algorithm defines an objective function expression, establishes constraints associated with the process, and utilizes floating-point decision variables within a range of possibilities. Furthermore, the algorithm allows for runtime configuration and performs multiple iterations to optimize the path by progressively reducing the improvement range to zero.

The chosen algorithm is tested with execution times of 1 and 5 s. These times are selected because the developed heuristic can generate three feasible solutions in a fraction of a second (less than 1 s) for the same instance. It is important to note that the type of solution chosen is a viable solution and not an exact solution that guarantees finding the minimum path.

4 Results

The developed model was applied to a randomly generated sample for the 20-client experiment (Fig. 6).

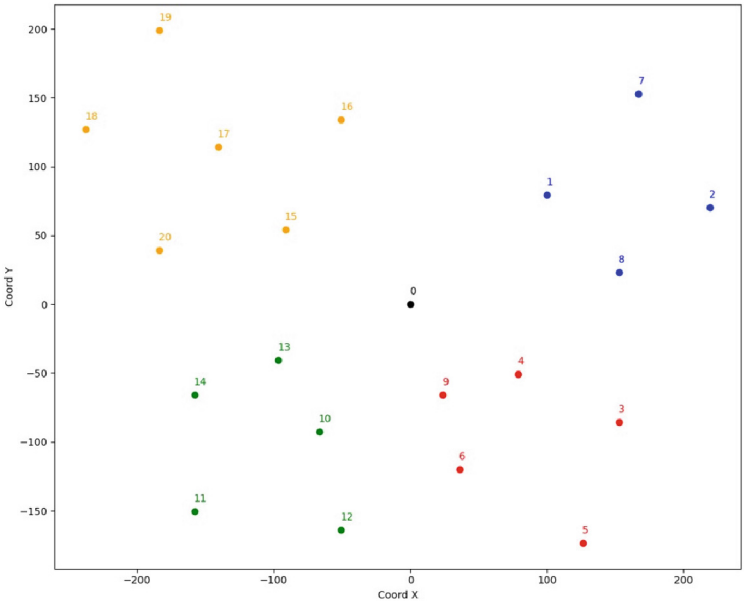

Fig. 6. Sample distribution in the plane

The models are developed based on various criteria, yielding the following results (Figs. 7, 8, 9 and Tables 1, 2, 3):

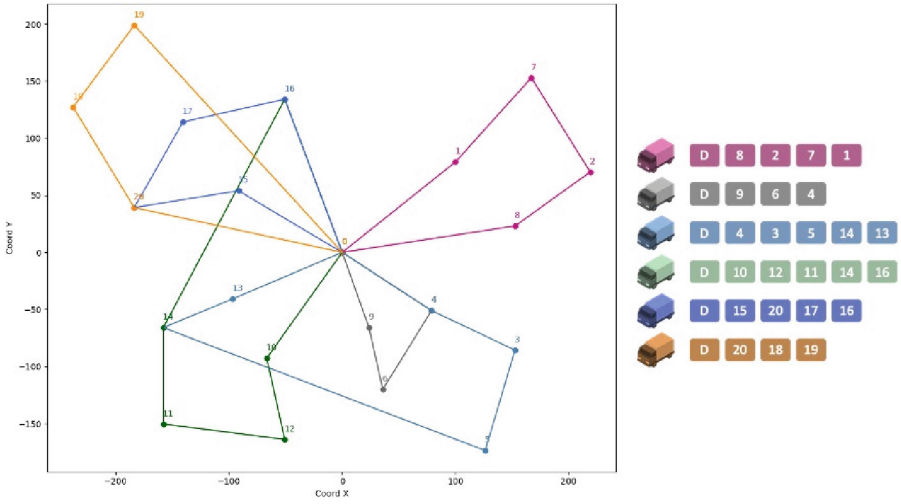

Fig. 7. Routes Criterion A.

Table 1. Information on each vehicle Criterion A

INFORMATION TRUCKS				
ID_TRUCKS	NUM_CUST	AVG_DEMAND	CAPACITY_ACHIEVED	TOTAL_TRAVEL
0	4	10	40	560.7515172
1	3	13.33	40	300.7292421
2	5	8	40	742.2734389
3	5	8	40	750.2377217
4	4	10	40	521.5598685
5	3	3	9	651.535338

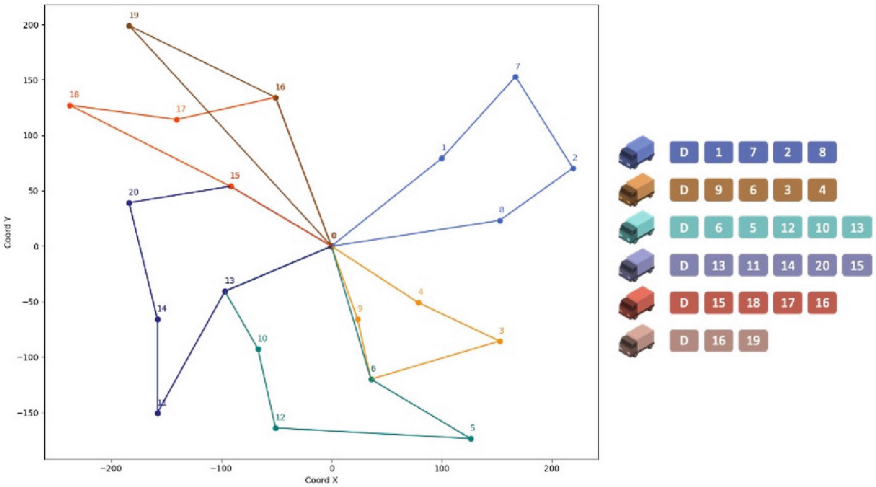

Fig. 8. Routes Criterion B

Table 2. Information on each vehicle Criterion B

INFORMATION TRUCKS				
ID_TRUCKS	NUM_CUST	AVG_DEMAND	CAPACITY_ACHIEVED	TOTAL_TRAVEL
0	4	10	40	560.7515172
1	4	10	40	422.6431497
2	5	8	40	645.484488
3	5	8	40	622.6179426
4	4	10	40	603.153697
5	2	4.5	9	562.0926964

In order to evaluate the program's efficiency, multiple instances are created with varying numbers of clients while maintaining the same format. These instances are designed not only to assess efficiency but also to test the program's performance against increasingly larger examples.

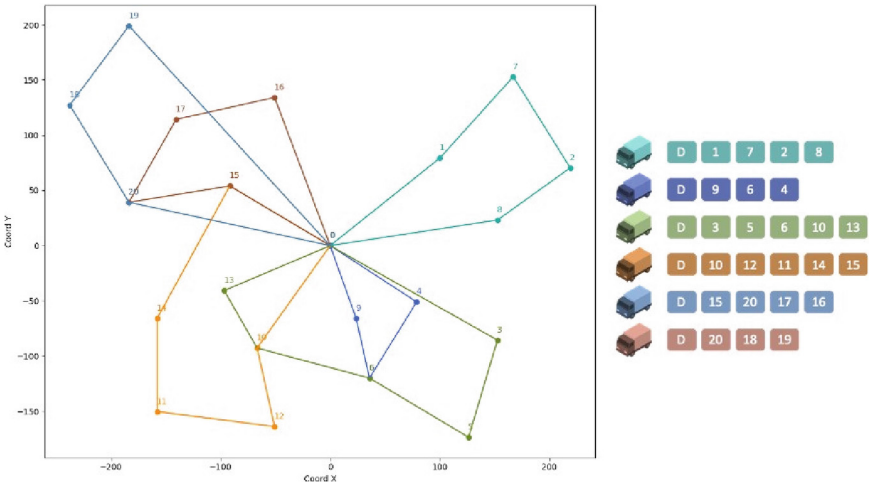

Fig. 9. Routes Criterion C

Table 3. Information for each vehicle in the 2-Opt route (Criterion C).

INFORMATION TRUCKS				
ID_TRUCKS	NUM_CUST	AVG_DEMAND	CAPACITY_ACHIEVED	TOTAL_TRAVEL
0	4	10	40	560.7515172
1	3	13.33	40	300.7292421
2	5	8	40	643.0658405
3	5	8	40	622.9965454
4	4	10	40	521.5598685
5	3	3	9	651.535338

Table 4 presents the instances that were tested, where the number of clients was incrementally increased to assess the efficiency of the heuristic in tackling more substantial examples. The third column indicates the number of trucks proposed by each model, while the fourth column demonstrates a significant decrease in this value. This reduction is achieved through the exploration of a lower limit for the number of vehicles required to fulfill the total demand.

Table 5 displays the total travel costs for each tested instance, both before and after implementing the 2-Opt algorithm.

4.1 Results of the Computational Experiment

The evaluation of the results for this computational experiment involves a comprehensive analysis of various factors, including execution time, number of iterations, covered demand, and total runtime. Additionally, the solution paths generated by the algorithm are taken into consideration. However, it is important to note that while the algorithm

Table 4. Comparison table

INSTANCE	N° OF CLIENTS	N° OF TRUCKS PROPOSED	MAXIMUM CAPACITY	N° TRUCS USED	DEMAND COVERED
1	10	5	25	4	81
2	20	13	30	12	352
3	30	20	35	13	432
4	40	26	40	17	677
5	50	35	52	27	1383
6	65	48	61	27	1600
7	80	60	75	33	2411
8	100	75	99	41	4046
9	120	98	130	51	6551

Table 5. Comparison 2-opt.

INSTANCE	TOTAL TRAVEL			2-Opt TOTAL TRAVEL		
	Criterion A	Criterion B	Criterion C	Criterion A	Criterion B	Criterion C
1	3861.95	3416.72	4024.10	3394.48	3189.46	3441.29
2	8334.78	8689.83	8577.33	8245.78	8321.43	8320.10
3	7677.18	7838.23	7896.29	7251.46	7019.82	7284.82
4	11391.02	11689.08	11720.76	10736.46	10539.30	10953.92
5	16537.99	15641.03	16859.40	15540.50	14561.34	15744.29
6	16591.50	16539.01	15666.16	15277.48	15649.08	14692.18
7	19486.90	19523.02	19614.87	18132.37	18263.63	17751.27
8	24990.24	22943.65	24546.60	22845.47	21013.63	21687.43
9	31271.47	29127.80	30462.75	28351.67	26859.26	26990.03

successfully minimizes the total path cost for this instance, to classify the resulting solution as exact, it is necessary to consider the concept of divided partitioning, which is not directly derived from the solution paths.

Table 6. Localsolver vs. proposed heuristics comparison.

	Local solver 1	Local solver 5	Proposed model
Time (sec)	1	5	0.043
Number of iterations	1675	80000	N/A
Covered demand (UN)	22500	22500	22500
Total travel	404	375	495.81

5 Conclusion

The proposed heuristic solution effectively addresses the constraints imposed by the mathematical model and satisfies the necessary conditions for feasible solutions. In particular, the algorithm demonstrates the ability to resolve large instances in less than one second of execution time (0.043 s) (Table 6).

Regarding the solutions generated based on different criteria, it is impossible to identify a consistent pattern that consistently delivers the optimal path. Multiple instances were generated with the same number of clients but different coordinate values and demands, and no single criteria consistently produced the lowest route. Therefore, finding a near-optimal feasible solution will depend on the specific instance values and not just the number of clients.

In terms of comparative analysis, the solutions provided by the LocalSolver algorithm are promising. However, it fails to fully implement the concept of split delivery in its solutions, which is a key aspect of the SD-VRP relaxation of the original VRP. The algorithm only allows one visit per client, without making the type of installment deliveries required.

Considering the concepts discussed and comparing them with the heuristics proposed earlier in this conclusion, it can be concluded that heuristics offer a reasonable solution for this type of routing problem. It effectively meets all the requirements and conditions described during problem definition.

References

1. Aleman, R.E., Zhang, X., Hill, R.R.: An adaptive memory algorithm for the split-delivery vehicle routing problem. J. Heurist. **16**, 441–473 (2010)
2. Archetti, C., Speranza, M.G.: A tabu search algorithm for the split delivery vehicle routing problem (2011)
3. Archetti, C., Speranza, M.G.: Vehicle routing problems with split deliveries. Int. Trans. Oper. Res. **19**(1–2), 3–22 (2012)
4. Archetti, C., Bianchessi, N., Speranza, M.G.: A column generation approach for the split delivery vehicle routing problem. Networks **58**, 241–254 (2011)
5. Archetti, C., Feillet, D., Speranza, M.G.: The complexity of the VRP and SDVRP (2011)
6. Belenguer, J.M., Martinez, M.C., Mota, E.: A lower bound for the split delivery vehicle routing problem. Oper. Res. **48**, 801–810 (2000)

7. Chvátal, V., Cook, W., Dantzig, G.B., Fulkerson, D.R., Johnson, S.M.: Solution of a large-scale traveling-salesman problem. In: Jünger, M., Liebling, Thomas M., Naddef, Denis, Nemhauser, George L., Pulleyblank, William R., Reinelt, Gerhard, Rinaldi, Giovanni, Wolsey, Laurence A. (eds.) 50 Years of Integer Programming 1958-2008, pp. 7–28. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-68279-0_1
8. Dror, M., Trudeau, P.: Savings by split delivery routing. Transp. Sci. **23**(2), 141–145 (1989). https://doi.org/10.1287/trsc.23.2.141
9. Dror, M., Trudeau, Pierre: Split delivery routing: split delivery. Naval Res. Logist. (NRL) **37**(3), 383–402 (1990). https://doi.org/10.1002/nav.3800370304
10. Jin, M., Liu, K., Bowden, R.O.: A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. Int. J. Prod. Econ. (2007)
11. Lin, T.-H., Sheng-Hung, C., Lin, I.-C.: A decision system for routing problems and rescheduling issues using unmanned aerial vehicles (2022)
12. LocalSolver. Retrieved from Example Tour - Split Delivery Vehicle Routing Problem: https://www.localsolver.com/docs/last/exampletour/split-delivery-vehicle-routing-problem-sdvrp.html
13. LocalSolver. Mathematical modeling features (2023). https://www.localsolver.com/docs/last/modelingfeatures/mathematicalmodelingfeatures.html
14. Slootbeek, J.J.A.: Average-Case Analysis of the 2-opt Heuristic for the TSP (2017)
15. Velazquez, E.: Application of the Vehicle Routing Problem (VRP) to a distribution problem. University of the Americas Puebla (2005)
16. Wilck IV, J., Cavalier, T.: Una construcción heurística para el vehículo de reparto dividido problema de enrutamiento. Am. J. Oper. Res. (2012)