Decision Science Letters 12 (2023) 441–456

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Design of a hybridization between Tabu search and PAES algorithms to solve a multi-depot, multi-product green vehicle routing problem

Juan Sebastián Azuero-Ortiz[a], María Alejandra Gaviria-Hernández[a], Vicky Magnolia Jiménez-Rodríguez[a], Edgar José Vale-Santiago[a] and Eliana María González-Neira[b*]

[a]*Research assistant, Departamento de Ingeniería Industrial, Facultad de Ingeniería, Pontificia Universidad Javeriana, Carrera 7 No. 40-62, Bogotá, D.C., Colombia*
[b]*Associate professor, Departamento de Ingeniería Industrial, Facultad de Ingeniería, Pontificia Universidad Javeriana, Carrera 7 No. 40-62, Bogotá, D.C., Colombia*

C H R O N I C L E

Article history:
Received: October 15, 2022
Received in revised format: November 2 2022
Accepted: November 25, 2022
Available online: November 25, 2022

Keywords:
Green VRP
Multi-depot
Multi-product
Tabu search
PAES

A B S T R A C T

Vehicle routing problem (VRP) is a classic problem studied in logistic. One of the most important variations within this problem is called Green Vehicle Routing Problem (GVRP), in which environmental aspects are considered when designing product delivery routes. This variant arises due to the high levels of pollution produced by transport vehicles, so it is a variation whose study represents a vital impact nowadays. This project will consider a GVRP and will be developed considering the characteristics of multi-depot (MDVRP) and multi-product (MPVRP) to minimize the costs of assignation of vehicles and CO2 emissions. To solve the problem, this project proposes a hybridization between the classic tabu search (TS) metaheuristic and the PAES algorithm (TS+PAES) to generate the Pareto frontier of both objectives. An integer mixed linear programming model is formulated and developed for each objective function separately to have an optimal point of comparison for the efficiency of the proposed algorithm. Also, the TS+PAES algorithm is compared to the nearest neighbor algorithm for large instances. Two computational experiments were carried out, one for small and the other one for large instances. The experiment for small instances showed that the GAP of each extreme of the frontier compared to the MILP model is on average 0.73%. For large instances, the metaheuristic improves in 0.1% the results presented by the MILP model showing that the metaheuristic provides closer near-optimal solutions in less computational time. Besides, the metaheuristic, in comparison with the nearest neighborhood heuristic, improves in 44.21% the results of emissions and in 3.88% the costs. All these results demonstrate the effectiveness of the metaheuristic.

1. Introduction

The *Vehicle Routing Problem* (VRP) consists of determining the optimal route for a set of vehicles, starting from a distribution center (or depot), to visit different nodes or customers that are geographically distributed and ending the route at a node (which can be the same distribution center or another point). These routes are made either for the delivery or collection of products or for the provision of services. Objectives such as minimization of distance, time, cost, among others, are usually considered (Wahyuningsih & Satyananda, 2020a). The *VRP* is a problem considered of NP-Hard complexity (Lenstra & Kan, 1981), where the number of solutions grows exponentially with the number of nodes, so the optimal solution through exact methods, in large instances, is not achieved in reasonable computational times. For this reason, different fast and efficient heuristic and metaheuristic methods have been implemented to solve the problem. According to Moghdani et al. (2021), the application of these techniques produces savings ranging from 5% to 20% in operational costs

* Corresponding author. Tel: +57 1 3208320
E-mail address: eliana.gonzalez@javeriana.edu.co (E. M. González-Neira)

doi: 10.5267/dsl.2022.11.004

at a global level, in addition to generating effectiveness in the execution times of the problems. These savings, generated by the application of computerized methods, are an extremely important factor for the global economic system, since the transportation process represents about 10% to 20% of the cost of the product. In this type of problem, the most usual characteristics are related to costs and times. In terms of costs, some of those taken into account are vehicle operating costs, costs of unsold products (losses) and costs related to the quantity of products transported. However, the arrival times to each of the customers, the times it takes for the products to arrive from the suppliers and the service times of vehicles are also taken into account (Toro et al., 2015; Rafati, 2022; Wofuru-Nyenke & Briggs, 2022). Another characteristic that is often studied, and in turn is related to real-life cases, is demand. This element affects both costs and time, since the amount of product to be transported occupies a certain amount of space in the vehicle, incurring costs related to the amount of product being transported. Likewise, time is also affected because the time it takes for the vehicle to load and unload the product depends on the amount of product (Zhang & Chen, 2014).

Among the different variants of VRP that have been studied, it is important to highlight three of them, which are currently receiving attention for their applicability in the industry and for making the problem closer to reality. These variants are: *Green Vehicle Routing Problem* (GVRP), *Multi-depot Vehicle Routing Problem* (MDVRP), and *Vehicle Routing Problem Multi-Product* (MPVRP). The GVRP has emerged as a response to the growing concern for the environmental factor. Due to the high rates of pollution caused by the transportation of products, the entire logistics and transportation industry has begun to address VRP problems through formulations that include the environmental factor. Currently, many of the efforts in the VRP are directed towards the development of cleaner solutions for the environment because distribution and transportation are not only important for economic development but also cause a dangerous effects on the environment because of their externalities such as pollution or noise (Sawik et al., 2017). In the MDVRP, several depots or distribution centers are studied. Since its first study in 1995 by Sumichras & Markham (1995), this topic has been constantly explored as it is more similar to experiences seen in the day-to-day operations of companies (Mehlawat et al., 2020). The MPVRP consists of transporting products with different characteristics in the same vehicle preventing them from being stored close to each other (Hameed et al., 2020). As an example, a supermarket that delivers groceries and toiletries must be clear about how to separate the products properly within the truck to bring the orders to their destination. Although this variation has been little studied, some researchers recommend including it to resemble the models in everyday situations (Hameed et al., 2020).

Considering the above elements, the purpose of this research is to address a multi-depot and multi-product bi-objective GVRP, finding the Pareto frontier between vehicle usage costs and CO2 emissions. As a solution method, it is proposed a hybridization of the Tabu Search (TS) metaheuristic (Glover, 1989) with the *Pareto Archived Evolution Strategy* (PAES) algorithm (Knowles & Corne, 2000), to obtain the Pareto frontier of the two selected objectives. TS is selected for its advantage over other metaheuristics of intelligently using the previous search history to influence future searches (Hajji et al., 2004). In addition, the PAES algorithm, which was originally developed as a local multi-objective search for the VRP and subsequently applied to different multi-objective combinatorial problems, has shown good results in these type of problems and other combinatorial problems such as scheduling.

The remainder of this paper is organized as follows. Section 2 presents literature review in GVRP, MDVPR and MPVRP. A mixed linear integer programming formulation is proposed in Section 3. Section 4 develops the TS+PAES algorithm. The computational experiments and results are discussed in section 5. Finally, section 6 exposes conclusions and further works.

2. Literature review

Studies in VRP began with the work of Dantzig et al. (1954) who studied the *Travelling Salesman Problem (TSP),* where the object of this problem is to minimize the costs or distances traveled by a single agent-vehicle with infinite capacity. Years later, the study by Clarke & Wright (1964) considered for the first time the routing of more than one vehicle simultaneously, using a "*greedy"* approach known as the savings algorithm in which the authors minimized the total cost of the routes. Over the years, several variants of the VRP have been studied, which have served to expose exactly a particular real-world problem. The *Capacitated VRP (CVRP)* has as its main feature a limited and deterministic vehicle capacity (Olivera, 2004). The *VRP with time windows (VRPTW)* variant considers time windows for visiting or delivering products to customers, i.e., there is a time interval in which a customer can be visited (Olivera, 2004). The *Stochastic VRP (SVRP)* (Bianchic et al., 2004) presents one or more random variables such as distances, travel times, demands, among others. The *MDVRP* consists of several depots with a fleet of vehicles per depot, which must meet the demand of all customers *(*Shi et al., 2020). The *Split Delivery VRP (SDVRP)* problem (Lee et al., 2006) occurs when multiple vehicles can supply the same customer's demand if that reduces the total cost, this variant is useful when customer order sizes are as large or larger than the capacity of the vehicles. The *GVRP* has been developed over the past few years due to the need to decrease pollution and excessive energy usage produced by vehicles, as these factors present a threat to the integrity of environmental and ecological conditions (Asghari & Mirzapour Al-e-hashem, 2021). The *MDVRP* variation consists of multiple depots or starting points from where vehicles must depart to different customers or arrival points, returning to their starting point (Hanum et. Al., 2019; Samsuddin et al., 2020; EL Bouyahyiouy & Bellabdaoui, 2021). Finally, the *MPVRP* encompasses real-life situations where a single vehicle may contain different compartments of different volumes, so that multiple products can be transported in that vehicle without running the risk that the products being located in shared compartments may be damaged (Kabcome & Mouktonglang, 2015; Chowmali & Sukto, 2020; Chowmali & Sukto, 2021).

A review of the literature on VRP in the last 5 years was carried out in the SCOPUS and ISI Web of Science databases, finding a total of 99 articles. The participation of the different variants mentioned is shown in Figure 1. It should be clarified that the category "others" refers to the sum of those articles in which a particular topic of VRP has been studied that no other article has analyzed, with the same characteristics, during those 5 years reviewed in the literature consulted. Considering the importance of routing that does not pollute the environment, that involves multiple depots and multiple products, as it happens in several industries, we present a detailed review of GVRP, MDVRP, and MPVRP problems, which only represent 30.7% of the consulted investigations. In particular, GVRP accounts for 18.8% of participation, MDVRP represents 7.9% and MPVRP is analyzed in only 4% of the cases.

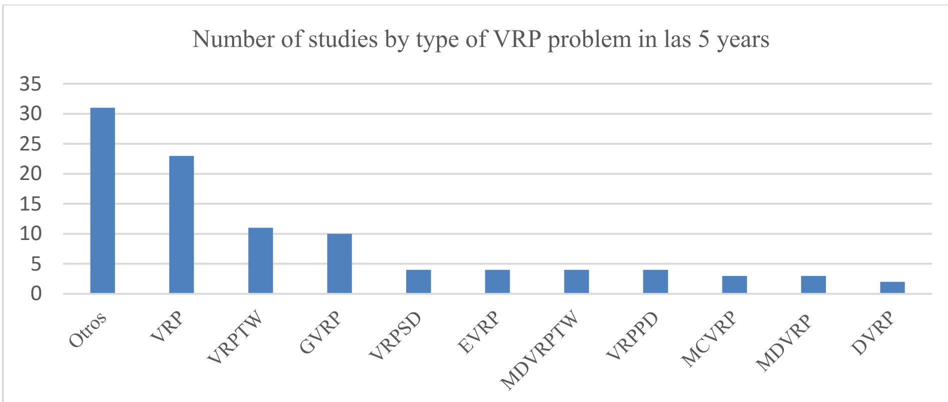

Fig. 1. Number of studies by type of VRP problem in las 5 years. Own Authorship

2.1 GVRP

Some studied variations of the GVRP are: *Electric vehicle routing problem* (EVRP), *Alternative fuel vehicle routing problem* (AFVRP) and *Hybrid vehicle routing problem* (HVRP). The main characteristic of EVRP is the use of electric vehicles and the application of strategies related to issues such as battery replacement, recharging cycle, battery life, partial recharging, among others. The AFVRP studies the utilization of alternative and environmentally friendly power sources such as biodiesel, electricity, ethanol, hydrogen, methanol, natural gas, among others. The main characteristic of HVRP is the ussage of vehicles with two or more propulsion sources (Ghorbani et al., 2020). In order to provide a solution to these GVRP variations, some studies have been developed in recent years, such as the one by Keskin et al. (2021), which focuses on EVRP by applying a simulation-based adaptive large neighborhood search (ALNS). Authors deal with the problem of the disruptions that can be generated in the operations due to vehicle recharging time. In the study of Löffler et al. (2020) the same problem was analyzed with the possibility of partial or full recharges by applying metaheuristics such as granular tabu search and large neighborhood search. Koç & Karaoglan (2016) focused on the AFVRP problem by applying limited driving range to vehicles because of limited alternative refueling infrastructure. Alizadeh Foroutan et al. (2020) conducted a study in which the green vehicle routing and scheduling problem with fleet heterogeneity, and reverse logistics in the form of returned goods collection was developed by applying metaheuristics such as genetic algorithms and simulated annealing.

2.2 MDVRP

This variation of the classical problem was first heard of by Sumichras and Markham (1995), who developed a multi-depot model in which the supplier had to transport to several company warehouses a certain amount of raw material (Azadeh & Farrokhi-Asl, 2019). Of the latter five years we can highlight the studies by Zheng (2019), Li et al. (2019), Karakatič (2021), Samsuddin et al. (2020), and Mehlawat et al. (2020). Zheng (2019) made a dynamic proposal for MDVRP with time windows based on *big data* analysis on traffic flow to have a real data approach to serve companies for their application. With the model based on data analysis they wanted to achieve minimization of total costs. Li et al. (2019) covered a multi-depot Green VRP seeking to minimize costs, time and emissions while maximizing profits. For this, they used an improved ant colony algorithm that implements a better pheromone system obtaining better results compared to a conventional ant colony algorithm. Karakatič (2021) analyzed the multi-depot problem taking into account time windows in electric vehicles having partial nonlinear recharging. To solve the problem the author proposed a two-layer genetic algorithm aiming to minimize driving times, the number of stops at recharging stations and their recharging time. Samsuddin et al. (2020) introduced a new hybrid algorithm called *Intelligent Water Drop* to solve the problem that considered as parameters the velocity of the water drops and the amount of soil they carried. The authors compared their results with an ant colony algorithm, showing better results in minimizing the distance traveled by the vehicles. Mehlawat et al. (2020) developed a multi-depot GVRP model considering generalized fuzzy travel times, split deliveries and heterogeneous vehicles, capacitated and powered by alternative fuels.

2.3 MPVRP

One of the characteristics that has a high presence in the works of MPVRP is the inclusion of costs. The fact that a vehicle has several compartments to transport heterogeneous goods reduces the costs related to the number of vehicles to be used and the operating costs of each of them (Asawarungsaengkul et al., 2013). Both the work of Asawarungsaengkul et al. (2013) and the study of Tavakkoli-Moghaddam et al. (2015) aimed to determine the optimal allocation of compartments in vehicles to design the delivery route, minimizing the total cost of the trip. For the first-mentioned work, the problem was solved using a local search, while the work of Tavakkoli-Moghaddam et al. (2015) is solved by employing the ε-constraints method. Likewise, the work of Parchami Afra & Behnamian (2021), whose objective function is to minimize the total operating costs, is solved using the Lagrangian relaxation (LR) algorithm. Another characteristic studied in the MPVRP is the volumes of products to be transported since these also incur operating costs and more time to load the vehicles. In the work of Zhang & Chen (2014) this cost was included within the objective function, solving the problem through a genetic algorithm.

2.4 Analysis of literature

After analyzing the elements presented in the three previous subsections, it can be highlighted that the present proposal presents an added value compared to the researches of Mehlawat et al. (2020) and Li et al. (2019) by including multiple products in the delivery routes. Besides, the article by Mehlawat et al. (2020) covers the GVRP problem by evaluating an alternative fuel for the assignment of vehicles to routes in a sustainable way, whereas our purpose is to reduce total emissions of the vehicles.

3. Mixed integer linear programming model for the deterministic multi-depot and multi-product GVRP

The deterministic mathematical model of the multi-depot and multi-product Green VRP problem is presented below.

Sets

N: Nodes $\{1, \ldots, |N|\}$

K: Vehicles $\{1, \ldots, |K|\}$

P: Products $\{1, \ldots, |P|\}$

$C \subset N$: Customers $\{1 \ldots |C|\}$

$D \subset N$: Depots $\{1 \ldots |D|\}$

Parameters

DD_{cp}: demand of customer $c \in C$ of product p $\in P$ $[units/day]$

O_{pd}: availability of product p $\in P$ on depot d $\in D$ $[units/day]$

DIC_{ij}: distance between node i $\in N$ and node $j \in N$ $[km]$

VV_k: volume capacity of vehicle $k \in K$ $[m^3]$

V_p: volume that occupies a unit of product $p \in P$ $[m^3/unit\ ]$

E_k: emissions per kilometer generated by vehicle $k \in K$ $[g\ CO_2/km]$

$DVMAX_k$: maximum distance that can be traveled by vehicle $k \in K$ $[km]$

TMAX: maximum time of travel of vehicle $k \in K$ per day $[h/day]$

M: big positive number

Vel: average velocity of travelling by vehicle $[km/h]$

CC_{ck}: fixed cost of charge and discharge of products of customer $i \in C$ on vehicle $k \in K$ $\left[\frac{\$}{customer \cdot vehicle}\right]$

Variables

$\alpha_{ijk:}$ binary variable that takes the value of 1 if vehicle $k \in K$ travels from node $i \in N$ to node $j \in N$, 0 otherwise.
$\beta_{dk:}$ binary variable that takes the value of 1 if vehicle $k \in K$ departs from deposit $d \in D$, 0 otherwise.
$\omega_{ik:}$ binary variable that takes the value of 1 if vehicle $k \in K$ is assigned to customer $c \in C$, 0 otherwise.

$\gamma_{dik:}$ binary variable that takes the value of 1 if vehicle $k \in K$ is assigned to customer $c \in C$ from depot $d \in D$, 0 otherwise.

Y_{ik}: total distance travelled of vehicle $k \in K$ when it arrives at node $i \in N$
X_k: total distance travelled by vehicle $k \in K$

Objective functions

$$min\ Z_1 = \sum_{\forall\, k \in K} X_k * E_k \tag{1}$$

$$min\ Z_2 = \sum_{\forall\, k \in K} \sum_{\forall i \in C} CC_{ik} * \omega_{ik} \tag{2}$$

subject to

$$\sum_{\forall\, i \in N} \sum_{\forall\, k \in K} \alpha_{ijk} = 1 \quad \forall\, i\, \in C \tag{3}$$

$$\sum_{\forall\, i \in N} \sum_{\forall\, k \in K} \alpha_{jik} = 1 \quad \forall\, i\, \in C \tag{4}$$

$$\sum_{\forall\, i \in N} \alpha_{ijk} = \sum_{\forall\, i \in N} \alpha_{jik} \quad \forall\, j\, \in C, \forall\, k\, \in K \tag{5}$$

$$\sum_{\forall\, i \in D} \sum_{\forall\, j \in D} \alpha_{ijk} = 0 \quad \forall\, k\, \in K \tag{6}$$

$$Y_{ik} + DIC_{ij} \leq Y_{jk} + M\left(1 - \alpha_{ijk}\right) \quad \forall\, i\, \in N, \forall\, j \in C, \forall\, k\, \in K \tag{7}$$

$$DVMAX_{\mathrm{k}} \geq Y_{ik} \qquad \forall i \in C\,; \forall\, k\, \in K \tag{8}$$

$$\sum_{\forall j \epsilon\, N} \alpha_{jik} = \omega_{ik} \qquad \forall\, i\, \in C, \forall\, k \tag{9}$$

$$\sum_{\forall j \epsilon\, N} \alpha_{ijk} = \omega_{ik} \quad \forall\, i\, \in C, \forall\, k\, \in K \tag{10}$$

$$\sum_{k \in K} \omega_{ik} = 1 \quad \forall\, i\, \in C \tag{11}$$

$$\sum_{\forall i\, \epsilon\, C} \sum_{\forall\, p\, \in P} \omega_{ik} * DD_{ip} * V_p \leq VV_k \qquad \forall\, k\, \in K \tag{12}$$

$$\sum_{i \in C} \alpha_{dik} = \beta_{dk} \quad \forall d \in D, \forall k \in K \tag{13}$$

$$\sum_{i \in C} \alpha_{idk} = \beta_{dk} \quad \forall d \in D, \forall k \in K \tag{14}$$

$$\sum_{d \in D} \beta_{dk} \leq 1 \quad \forall k \in K \tag{15}$$

$$Y_{ik} + DIC_{id} \leq X_k + M(1 - \beta_{dk}) + M(1 - \alpha_{idk}) \quad \forall i \in N, \forall k \in K, \forall d \in D \tag{16}$$

$$\sum_{\forall i\, \epsilon\, C} \sum_{k \in K} \gamma_{dik} * DD_{ip} \leq O_{dp} \qquad \forall p \in P, \forall d \in D \tag{17}$$

$$\sum_{d \in D} \gamma_{dik} = \omega_{ik} \quad \forall i \in C, \forall k \in K \tag{18}$$

$$\sum_{i \in C} \gamma_{dik} \leq \beta_{dk} * M \quad \forall d \in D, \forall k \in K \tag{19}$$

$$\sum_{i \in C} \gamma_{dik} \geq \beta_{dk} \quad \forall d \in D, \forall k \in K \tag{20}$$

$$\sum_{d \in D} \sum_{k \in K} \gamma_{dik} = 1 \qquad \forall i \in C \tag{21}$$

$$Y_{ik} \leq \mathrm{TMAX} * Vel \qquad \forall i \in N, \forall\, k\, \in K \tag{22}$$

$$\alpha_{iik} = 0 \qquad \forall i \in N, \forall\, k\, \in K \tag{23}$$

$$Y_{ik} \geq 0 \qquad \forall\, i \in N, \forall\, k\, \in K \tag{24}$$

$$\alpha_{ijk} \in \{1{,}0\}\ 0 \quad \forall\, i \in N, \forall\, j \in N, \forall\, k\, \in K, \forall\, p\, \in P \tag{25}$$

$$\beta_{ik} \in \{1{,}0\} \qquad \forall\, i \in D, \forall\, k\, \in K \tag{26}$$

$$\omega_{ik} \in \{1{,}0\} \qquad \forall\, i \in N, \forall\, k\, \in K \tag{27}$$

$$\gamma_{dik} \in \{1{,}0\} \qquad \forall\, d \in D, \forall i \in C, \forall\, k\, \in K \tag{28}$$

Eq. (1) and Eq. (2) refer to the objective functions, minimization of emissions and costs, respectively. Constraint sets (3) and (4) ensure that each customer has only one predecessor and one successor customer on the assigned vehicle route. Constraint set (5) ensures the balance of inputs and outputs of each vehicle at each node. Equation (6) ensures that there is no interference between depots. Eq. (7) allows calculating the distance traveled by a vehicle when it arrives at a given customer (node). Constraint set (8) ensures that each vehicle does not travel more than the maximum distance allowed for its route. Eq. (9), Eq. (10), and Eq. (11) ensure that a customer is visited by a single vehicle and cancel the possibility of a customer being on a route if the vehicle is not assigned to that customer. Constraint set (12) ensures that a vehicle does not carry more product than its allowed capacity. Constraint sets (13) and (14) ensure that vehicles return to depots and constraint set (15) assigns each vehicle to at most one depot, as not all available vehicles may be needed. Equation (16) calculates the total distance traveled by each vehicle to return to the depot. The sets of constraints (17) to (21) correspond to the fulfillment of offers and demands. The maximum route time per vehicle is constrained by Eq. (22) and Eq. (23) and Eq. (24) to Eq. (28) correspond to the domain of the decision variables.

4. Hybridization between tabu search and PAES (TS+PAES) algorithms to solve the multi-depot multi-product Green VRP

To solve the multi-depot and multi-product GVRP, an algorithm that hybridizes TS and PAES algorithms is proposed (TS+PAES). The TS allows improving local search by keeping a memory of the solutions performed, and optimizing the search. The PAES algorithm deals with the construction of the Pareto frontier between the two proposed objective functions. This proposal starts by performing an initial routing based on the nearest neighbor algorithm (NNA), which allows assigning the available vehicles to the departure depot and the next nodes to be visited, considering the availability of products, capacity of vehicles, and demand. Algorithm 1 shows the pseudo-code of the methodology used for the initial routing taking into account the NNA.

Algorithm 1.
Initial Routing Pseudo Code. Own Authorship

```
For i in the range of number of deposits
  Add deposit offer to auxiliary variables
    While there are nodes to be visited and the product offer is available in the warehouse
      For j in the range of number of vehicles
        If evaluates whether the vehicle is still available for assignment
          Add to the initial node the i deposit
          Add available capacity of the vehicle to the auxiliary variable
          Add the first node of the vehicle route (Start node = depot)
          Add counter that records the number of nodes visited by the vehicle.
            While the vehicle has available capacity and supply in the warehouse
              Use nearest neighbor function
                If the nearest neighbor is different from the current node
                  Add new node to visit
                  Add assigned occupied value to the new node
                  Subtract new node demand from available vehicle demand
                  Add node to vehicle routing matrix
                  Add in matrix quantity of product 1 delivered to the node
                  Add in matrix quantity of product 2 delivered to the node
                  Subtract demand of node to supply of product depot 1
                  Subtract demand from node to supply from product depot 2
                Else nearest neighbor is equal to current node
                  Available capacity is zero
                  The vehicle returns to the depot
                  Add occupied value to the vehicle
                    If the position of the vehicle still has no nodes to attend to
                      Add available value to the vehicle
       Add 0 in deposit offer
Even FO1 value with function 1
Even FO2 value with function 2
Return value FO1 and FO2
```

To visualize the generated routing solution a two-dimension matrix is proposed. Each colum shows the assignment made to each vehicle. Each row presents the order in which the visit will be made. Table 1 shows an example of a solution for the initial routing of an instance of the problem in which there are two depots (nodes 1 and 2), two types of products, four customers (nodes 3 to 6), and three vehicles.

Table 1
Example of route matrix. Own Authorship

	VEHICLES		
	1	**2**	**3**
ROUTE	1	1	1
	4	6	5
	3	1	1
	1		

Since sets C and D are subsets of set N, the enumeration is shared. That is, in the example, we have a set of nodes with $N = \{1, \dots, 6\}$, where the first two nodes correspond to the depots and the remaining ones (3 to 6) correspond to the customers. Considering the above, for the case of the first vehicle, the assigned route is: 1-4-3-1, which means that it leaves the first depot, visits customers three and four in that order, and returns to the depot from which it left. Likewise, since the first assignment node for the three vehicles is number one, it means that the supply of products from this depot is sufficient to cover the demand of all customers and it is not necessary to activate the second depot. Likewise, with this solution, a value for both objective functions is obtained. Once the initial routing is developed, leaving, as a result, the matrix of generated routes, the TS is implemented. In each iteration of the TS, a change or update is made in the structure of the solution that will reestablish the initial solution of the following iteration, generating variations in the objectives to be evaluated in order to find improvements in them. To carry out this change in the solution structure, it is necessary to explore the neighborhood or set of possible candidate route configurations. When a neighborhood solution is evaluated, it must also be evaluated for feasibility and the change made must be recorded in the *Tabu List.* The number of iterations that are forbidden to make the change found is limited by an input parameter called Tabu list size. Each time an iteration occurs the values in the *Tabu List*, found in previous iterations, are updated, acting as a counter that goes down each time an iteration occurs (see Algorithms 2 and 3).

Algorithm 2
Pseudo code of TS. Own Authorship

1. ***Generate*** *initial solution*
2. ***Do While*** *the number of iterations is not met*
3. *Calling a function that identifies the* ***best non-tabu neighborhood shift***
4. *Make the change in the initial solution*
5. *Update Tabu list counter*
6. *Restrict in tabu list the change between clients that has just been made*
7. *Evaluate running time so that it does not exceed 2 hours per instance.*
8. ***Loop***

Algorithm 3
Pseudo-code function better change of non-tabu neighborhood shift. Own Authorship

1. ***Receives initial solution***
2. ***For*** *v in the range of the solution columns*
3. ***For*** *k in the range of the solution columns*
4. ***For*** *i in the range of solution rows*
5. ***For*** *j in the range of solution rows*
6. *Identify the change between customers: Routes(j,k) = Customer 1 Routes(i,v) = Customer 2*
7. ***If*** *the change is not registered in the Tabu List*
8. *Switching between clients on the route: Routes(j,k) = Client 2 Routes(i,v) = Customer 1*
9. ***If*** *The route with the change meets feasibility conditions*
10. *Evaluate target functions*
11. *Call PAES*
12. ***If*** *the solution is not mastered*
13. *Assign positions of changes in the route*
14. *Stop neighborhood search and return positions*
15. ***Else***
16. *Reverse the change made to the route: Routes(j,k) = Customer 1 Routes(i,v) = Customer 2*
17. ***If*** *the function did not find a dominated solution*
18. ***Return*** *positions of change that generated better FO in exploration*

In order to find neighborhood solutions, two types of changes in the structure of the neighborhood proposed by (Wahyuningsih & Satyananda, 2020b) were implemented. The first one corresponds to Exchange, presented in Fig. 2, which is a permutation of the order of visiting two clients of the same vehicle (intra-route). The second one is the Swap(1,1), shown in Fig. 3, which consists of a swap between one customer from the route of one vehicle and one customer-j from the route of a second vehicle.

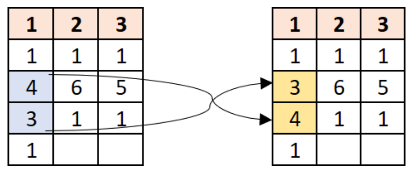

Fig. 2. Exchange. Own Authorship

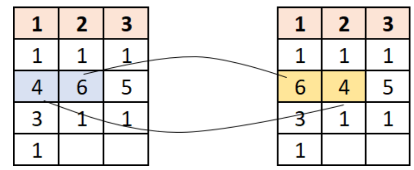

Fig. 3. Swap(1,1). Own Authorship

At the time of implementing the local search, each solution found is evaluated with the PAES algorithm in order to testing dominance between solutions. Each solution found can be classified into two groups: *Dominated* or *Not Dominated*. In the case of the present work a solution S1 is said to be dominated by a solution S2 when *Emissions (S1) > Emissions (S2) and Cost (S1) ≥ Cost (S2)* or also when *Emissions (S1) ≥ Emissions (S2) and Cost (S1) > Cost (S2)*, referring to the values of both objective functions. If a solution $S1$ is not dominated by $S2$ and $S2$ is not dominated by $S1$ it is said that there is no dominance between $S1$ and $S2$. Algorithm 4 shows the PAES pseudocode for a better understanding.

Algorithm 4
PAES pseudocode (1+1). Based on (Knowles & Corne, 2000)

1. ***Generate*** *initial solution c and add it to the file*
2. ***Mutate*** *c to produce m and evaluate it.*
3. ***If*** *c dominates m, discard m*
4. ***Else if*** *m dominates c*
5. ***Replace*** *c with m and add m to the file*
6. ***Else if*** *m is dominated by any other member of the file, discard m*
7. ***Else*** *perform the (c, m, file) test to determine what the new solution will be and if m is added to the file*
8. ***Until*** *the last criterion has been reached, return to line 2.*

Taking into account the PAES operation methodology, there is a matrix (Pareto archive) in which the values of the objective functions of the non-dominated solutions will be found. Within this archive the current solution is located next to the group of possible solutions. Table 2 shows an example of a Pareto archive obtained with 10 solutions. It can be seen that, according to the criterion explained above, no solution is dominated by another one inside the archive.

Table 2
Example of a Pareto archive with contains non-dominated solutions. Own Authorship

		Position *i* of a solution into the archive	Objective function 1 Emissions [g CO2]	Objective function 2 Costs [$]
Solution	**S5**	1	166841	79882
	S3	2	171952	75638
	S4	3	173896	67403
	S8	4	175266	61363
	S9	5	182415	56737
	S10	6	224541	53626
	S6	7	239002	53324
	S2	8	248508	53028
	S1	9	261335	53006
	S7	10	264475	52995

When a new solution found within the local search is to be evaluated, the dominance test must be performed with the current solution. In case of non-dominance between the mutated solution and the current solution, the first one must be compared with the other solutions of the archive because it must be ruled out if it is a dominated solution. In case it is not dominated, it is evaluated using the *Crowding Region* methodology. This method allows the identification of those regions with a higher number of solutions within a Cartesian plane.

Each solution is assigned a value called *Crowding Distance which* means the distance in which each solution is located with respect to the two closest ones. In order to exemplify the calculation of the *Crowding distance* we will use the PAES file of Table 2, calculating it for the solution identified with the number 6. Initially, the solutions of the file must be organized in ascending order, by one of the objective functions, and the formula expressed in Eq. (29) must be applied to calculate the distance between solution i and its closest neighbors for each objective function. Thence Eq. (30) must be used to calculate the overall crowding distance of solution i.

$$d_i(Emissions) = \frac{|Emissions_{i+1} - Emissions_{i-1}|}{Emissions_{max} - Emissions_{min}} \tag{29}$$

$$d_i(Costs) = \frac{|Costs_{i+1} - Costs_{i-1}|}{Costs_{max} - Costs_{min}} \tag{30}$$

$$cd_i = d_i(Emissions) + d_i(Costs) \tag{31}$$

where $d_i(Emissions)$ is equivalent to the *crowding distance* of solution i for emission objective, $Emissions_{i+1}$ is the value of emissions of the solution located immediately after solution i in the Pareto archive, $Emissions_{i-1}$is the value of emissions of the solution located immediately before solution i in the Pareto archive, $Emissions_{max}$ to the maximum value of emissions within the Pareto archive, $Emissions_{min}$ refers to the minimum value of the emissions within the archive, the same definitions are applied to the costs objective and, cd_i is the *overall crowding distance* of solution i.

An infinite *crowding distance* value is assigned to the solutions located at the extremes of the Pareto archive, because the higher this value is, the less populated the region is. It forces the solutions at the extremes to remain on the border. Eq. (29) to (31) are used for the solutions located in the middle of the extreme solutions. Next, in Eq. (32) to Eq. (34) is exemplified the calculation of overall crowding distance for the solution S6 located in the position $i = 7$ (cd_7) of Pareto archive of Table 3.

$$d_6(Emissions) = \frac{|248508 - 224541|}{264475 - 166841} = 0{,}2455 \tag{32}$$

$$d_i(Costs) = \frac{|53028 - 53626|}{79882 - 52995} = 0{,}0222 \tag{33}$$

$$cd_i = d_i(Emissions) + d_i(Costs) = 0{,}2455 + 0{,}2222 = 0{,}2677 \tag{34}$$

Once the *overall crowding distance* value of all the solutions in the archive is obtained, the corresponding changes are made, seeking that only solutions that are in less populated regions remain in the archive. In case that the value of *overall crowding distance* of the mutated solution is higher than the overall crowding distance of current solution, the mutated solution enters to the archive as the new solution. In case it is lower, compared to all the solutions in the file, the mutated solution is discarded.

5. Computational experiments

5.1 Instances to be tested

Considering that the multi-depot multi-product GVRP has not been solved before, to the best of our knowledge, the instances to solve were generated based on some characteristics of instances presented in research of (Christofides & Eilon, 1969), (Cordeau & Maischberger, 2012), (Subramanian et al., 2010) and (Chao I-Ming, Bruce L. Golden, 2013). In this vain, we defined the number of customers, their location coordinates and demands; the number of depots and their coordinates; and the number of avaliable vehicles and their type.

Since there are few studies carried out on multi-product VRP, instances that allow defining characteristics for variables of this typology were not found. However, the study conducted by (Ramos et al., 2011), in which a multi-product and multi-deposit VRP was developed, it is mentioned the difficulty of implementing large instances with algorithms that can reach the optimum. Therefore, it was decided to handle only two products in order to test the multi-product characteristic. For the demands of the second product per customer, a randomization is developed between the minimum value and the maximum value of the demands of the first product, which were taken from the previously mentioned instances.

Considering that vehicle emissions are required for the problem being developed, the guide for the calculation of greenhouse gas (GHG) emissions published by (Oficina Catalana del Cambio Climático, 2014) was investigated. In this one, an analysis of freight transport is made where it is stated that the amount of CO2 emissions produced by a vehicle depends on its characteristics and speed. However, they mention a range of emissions between 169 and 265 gCO2/Km. This is why the decision was made to generate random values in this range for the instances created. Likewise, in order to manage versatility in the model analysis, heterogeneous vehicles from mini box trucks to six-axle articulated vehicles are used. Table 3 shows the weight and volume that can be transported by each type of fleet in Colombia, so the allocation of volume and type of fleet in the instances is generated randomly, considering this classification. The time limit for the duration of the vehicle on a route is associated with the eight-hour work schedule.

Table 3

Fleet classification based on (Martinez, 2014)

Type of vehicle	Weight capacity [ton]	Volume capacity [m^3]
Mini box truck	2	12
Box truck	4.5	18
Two axle truck	8	32
Three axle truck	17	36
Three axle articulated vehicle	15	65
Five axle articulated vehicle	30	65
Six axle articulated vehicle	35	65

Finally, to obtain the value of the loading and unloading costs, the Association of Manufacturers and Distributors (JPIsla Asesores y Consultoría Logística, 2002) was consulted. They mention that for a 44-ton capacity truck loading 33 palletized units, the total loading and unloading time, including the processing of documentation, is standardized at 90 minutes. With this value, it is possible to obtain the loading and unloading time per unit for each type of vehicle. Once this value is calculated, and bearing in mind that for the year 2021 in Colombia the ordinary hourly salary is \$3786, the total cost of loading and unloading for each combination of customer and vehicle is computed by multiplying the ordinary salary per hour with the time for loading and unloading the total demand of each client. Considering the aforementioned elements, a total of 10 small instances and 32 large instances were created, the main characteristics of which are listed in Table 4.

Table 4

Main characteristics of instances. Own Authorship

Instance number	Size	Number of customers	Number of depots	Number of vehicles	Number of products
1	Small	4	2	3	2
2	Small	4	2	3	2
3	Small	4	2	3	2
4	Small	4	2	3	2
5	Small	4	2	3	2
6	Small	4	2	3	2
7	Small	4	2	3	2
8	Small	4	2	3	2
9	Small	4	2	3	2
10	Small	4	2	3	2
11	Large	50	4	4	2
12	Large	50	4	2	2
13	Large	75	5	3	2
14	Large	100	2	8	2
15	Large	100	2	5	2
16	Large	100	3	6	2
17	Large	100	4	4	2
18	Large	249	2	14	2
19	Large	249	3	12	2
20	Large	249	4	8	2
21	Large	249	5	6	2
22	Large	80	2	5	2
23	Large	80	2	5	2
24	Large	80	2	5	2
25	Large	160	4	5	2
26	Large	160	4	5	2
27	Large	160	4	5	2
28	Large	240	6	5	2
29	Large	240	6	5	2
30	Large	240	6	5	2
31	Large	360	9	5	2
32	Large	360	9	5	2
33	Large	360	4	9	2
34	Large	96	4	4	2
35	Large	144	4	4	2
36	Large	192	4	4	2
37	Large	240	4	4	2
38	Large	288	4	4	2
39	Large	72	4	6	2
40	Large	144	4	6	2
41	Large	216	4	6	2
42	Large	288	4	6	2

6. Parameterization of the TS+PAES

In order to be able to run the proposed TS+PAES algorithm initially, the value of the following parameters was selected through computational experiments:

a. Tabu List Size (TLS): Number of spaces in a list where those solutions or attributes already tested are registered. This is used so that the algorithm does not re-evaluate them as possible solutions within the local search and has the possibility of evaluating them as possible solutions.

b. Number of Iterations without improvement (NIWI): It is the maximum number of iterations that the algorithm performs without finding a solution that presents an improvement in the objective functions. This is one of the stop criteria taken into account and it is fulfilled in case the algorithm finishes and the maximum running time has not yet expired.

Regarding the size of Pareto archive, 15 was the upper limit for the quantity solutions to be saved in Pareto archive. The selection of the values for both TLS and NIWI was developed by running some experiments with 10 instances selected at random from all the instances created. The GAP in Eq. (30) was measured for both objective functions in order to have an indicator that allows to choose the values for TLS and NIWI.

$$GAP = \frac{ResultObjectiveFunction_{metaheuristic} - ResultObjectiveFunction_{MILP}}{ResultObjectiveFunction_{MILP}} * 100 \tag{30}$$

Fig. 4 presents the results of GAP for different NIWI values and Fig. 5 for different TLS values. The NIWI value where both objective functions converge is 110 iterations, with an approximate GAP value of 0.1%. In the case of TLS, the value for which both objective functions converge is 25, which reaches a GAP of approximately 0.30%. In the case of these tests, negative GAP values are evident, for which it can be predicted that at the time of running the code a value closer to the optimum than the mathematical model can be found. This occurs especially with large instances, since the mathematical model is not able to find the optimum solution in reasonable computational times due to the complexity of the problem. Taking into account the previously mentioned elements, the variables are parameterized as follows: the tabu list size in 25, and the number of iterations without improvement in 110.

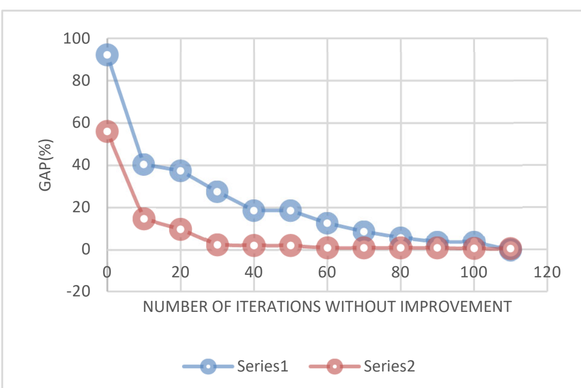

Fig. 4. Graph of Parameter Iterations. Own Authorship

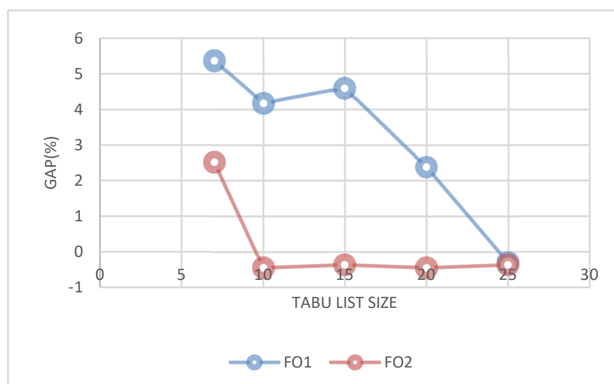

Fig. 5. Tabu List Parameter Graph. Own Authorship

7. Evaluation of the TS+PAES in comparison to the MILP model.

7.1 Evaluation of Small Instances

This section presents the results obtained for the small instances by solving them through the MILP model in the software GLPK and with the TS+PAES proposed in order to compare the results of the objective functions and the effectiveness of the proposed algorithm. For this performance analysis, the GAP indicator explained in Eq. (30) is used. The GAP allows knowing how good the results of the TS+PAES are in comparison with the MILP model. For this purpose, the solution of the Pareto frontier with the best objective function value for each of the functions is taken and compared against the value of the respective objective function given by the MILP. Table 5 shows the results of the TS+PAES, the mathematical model, and the GAP value obtained, as well as the run time obtained for the TS+PAES. The mathematical model was run for a limit time of 7200 seconds.

Table 5
Results obtained for the small instances. Own Authorship

Instance Number	MATHEMATICAL MODEL		TS+PAES		GAP		TS+PAES run time (s)
	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)	
1	23150	955	23366	968	0.93%	1.36%	28.23
2	25357	4373	25771	4373	1.63%	0.00%	33.07
3	10652	5311	10717	5339	0.61%	0.53%	29.05
4	28132	4818	28132	4863	0.00%	0.93%	30.11
5	17302	2632	17376	2654	0.43%	0.84%	30.01
6	22742	7424	22967	7494	0.99%	0.94%	29.06
7	30011	6734	30145	6801	0.45%	0.99%	32.56
8	57226	8821	57283	8899	0.10%	0.88%	31.48
9	118756	11162	119908	11259	0.97%	0.87%	27.09
10	105227	14118	106083	14179	0.81%	0.43%	29.25

The results obtained show that TS+PAES presents an excellent performance, finding optimal solutions or very close to the optimal values, obtaining an average GAP of 0.69% for emissions and 0.78% for costs.

7.2 Evaluation of Large Instances

In order to have another point of comparison between the TS+PAES and the mathematical model, also large instances were evaluated to determine the performance of the proposed algorithm. The runs of the mathematical model were performed also by using the GLPK software, being clear that the size of instances could run outside the two-hour limit and find a non-optimal result or no result at all in any of the instances. To help mitigate this inconvenience, the Neos Solver platform was utilized, which employs its servers to execute the modeling problem and provides a result for each instance after a maximum running time of 2 hours. Table 6 shows the results obtained from the objective functions for this group of instances and the calculation of the GAP, using Eq. (30).

Table 6
Results obtained for the large instances using NEOS. Own Authorship

Instance number	MILP MODEL		TS+PAES		GAP		Running time metaheuristic (s)
	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)	
1	143487	98919	146636	96266	-2.19%	2.68%	218.08
2	121055	44019	117847	44156	2.65%	-0.31%	215.09
3	163932	83899	161549	85974	1.45%	-2.47%	235.22
4	213386	179099	210468	180272	1.37%	-0.65%	908.51
5	211943	117555	216891	116759	-2.33%	0.68%	749.69
6	204327	109659	195441	110577	4.35%	-0.84%	470.03
7	229549	88722	227337	88941	0.96%	-0.25%	422.34
8	912543	863577	951047	850820	-4.22%	1.48%	2822.83
9	1185792	1109456	1183510	1095919	0.19%	1.22%	7224.68
10	979763	969983	951923	961903	2.84%	0.83%	7359.31
11	779245	1159340	799030	1155451	-2.54%	0.34%	7212.42
12	282687	16486	290726	16520	-2.84%	-0.21%	177.78
13	297898	20754	303746	20784	-1.96%	-0.14%	277.13
14	333576	38765	335300	38841	-0.52%	-0.20%	309.84
15	807613	41408	832035	41486	-3.02%	-0.19%	2573.38
16	699232	70362	708994	70749	-1.40%	-0.55%	850.71
17	613638	70653	624265	71099	-1.73%	-0.63%	1971.95
18	1135362	127693	1110955	124590	2.15%	2.43%	7273.62
19	965797	89672	947786	88708	1.86%	1.08%	7217.91
20	997627	38232	994898	37395	0.27%	2.19%	5344.07
21	1229834	228345	1222400	224113	0.60%	1.85%	5464.36
22	1339328	206718	1315065	205767	1.81%	0.46%	2960.27
23	1458121	1769901	1445396	1742715	0.87%	1.54%	2321.86
24	372198379	62493	382061344	63074	-2.65%	-0.93%	1160.50
25	571213491	78913	581686592	79998	-1.83%	-1.37%	1675.76
26	446957841	183493	454565952	188593	-1.70%	-2.78%	3115.60
27	426870120	169279	419438528	168631	1.74%	0.38%	2222.31
28	739563139	243381	732678080	235145	0.93%	3.38%	194.45
29	258763568	83814	261129504	83982	-0.91%	-0.20%	2362.11
30	553309826	129999	554776640	132893	-0.27%	-2.23%	2359.25
31	567945786	240987	558617344	241765	1.64%	-0.32%	2354.25
32	617894235	259864	608606656	251818	1.50%	3.10%	1660.47

The GAP calculation shows that the results for the large instances are closer to the results obtained with the mathematical model, allowing to obtain a GAP of 0.09% for emissions and -0.29% for costs. Likewise, it is observed that the GAP value in 50% of the results is negative, which means that the proposed metaheuristic finds solutions closer to optimum in lesser computational times than the mathematical model.

8. Evaluation of metaheuristics with respect to nearest neighbor heuristic on large instances.

One of the objectives of the present work is to compare the results of the TS+PAES against the solution given by the NNA. For this purpose, the percentage improvement (PM) indicator presented in Eq. (31) is calculated for each objective function separately. Likewise, Table 7 shows the results obtained from the objective functions for this group of instances and the PM.

$$PM = \frac{ResultObjectiveFunction_{metaheuristic} - ResultObjectiveFunction_{NNA}}{ResultObjectiveFunction_{NNA}} * 100 \quad (31)$$

Table 7
Results obtained for the large instances of the NNA and the metaheuristic. Own Authorship

Instancia	NNA heuristic		TS+PAES		PM	
No	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)	FO1 (Emissions)	FO2 (Costs)
1	334952	99855	146636	96266	-56.22%	-3.59%
2	272899	48811	117847	44156	-56.82%	-9.54%
3	455141	90383	161549	85974	-64.51%	-4.88%
4	542615	180543	210468	180272	-61.21%	-0.15%
5	525077	127608	216891	116759	-58.69%	-8.50%
6	531758	113585	195441	110577	-63.25%	-2.65%
7	388374	88941	227337	88941	-41.46%	0.00%
8	5418415	850820	951047	850820	-82.45%	0.00%
9	2014114	1111446	1183510	1095919	-41.24%	-1.40%
10	1870744	969721	951923	961903	-49.12%	-0.81%
11	884626	1158820	799030	1155451	-9.68%	-0.29%
12	922908	18244	290726	16520	-68.50%	-9.45%
13	398945	22684	303746	20784	-23.86%	-8.38%
14	887992	41057	335300	38841	-62.24%	-5.40%
15	1936997	64548	832035	41486	-57.05%	-35.73%
16	1870563	70749	708994	70749	-62.10%	0.00%
17	1743361	73825	624265	71099	-64.19%	-3.69%
18	3207058	124590	1110955	124590	-65.36%	0.00%
19	1976968	100145	947786	88708	-52.06%	-11.42%
20	2960056	37395	994898	37395	-66.39%	0.00%
21	1589448	234351	1222400	224113	-23.09%	-4.37%
22	1375809	209567	1315065	205767	-4.42%	-1.81%
23	4015028	1742715	1445396	1742715	-64.00%	0.00%
24	499368425	63311	382061344	63074	-23.49%	-0.37%
25	724181089	80931	581686592	79998	-19.68%	-1.15%
26	2412124699	197650	454565952	188593	-81.15%	-4.58%
27	2566159384	168631	419438528	168631	-83.66%	0.00%
28	799406397	239908	732678080	235145	-8.35%	-1.99%
29	261129596	84124	261129504	83982	0.00%	-0.17%
30	557492037	135416	554776640	132893	-0.49%	-1.86%
31	558617356	242973	558617344	241765	0.00%	-0.50%
32	608961185	255930	608606656	251818	-0.06%	-1.61%

The PM gives negative results which means that there is an improvement (decrease) in the objective functions with respect to the NNA results. The average improvement for emissions is -44.21% and for costs is -3.88% showing an improvement in the objective functions in 89.1% of the solutions. This shows that the metaheuristic is efficient and does produce an improvement over the heuristic initially employed.

9. Limitations, conclusions and recommendations

In this paper, a multi-depot and multi-product bi-objective GVRP was solved by finding the Pareto frontier for the minimization of vehicle usage costs and CO_2 emissions. As a solution method, the Tabu Search metaheuristic hybridized with the *Pareto Archived Evolution Strategy* (PAES) algorithm was developed to obtain the Pareto frontier of the two selected objectives.

The metaheuristic was evaluated in small and large instances. First, a comparison of the best results of each objective function given by the Pareto frontier versus the result given by the implementation of the mathematical model was performed. An average GAP of 0.73% was obtained in small instances and -0.1% in large instances, showing that the metaheuristic is efficient in finding good solutions. Additionally, comparisons were made of the results of the metaheuristic against those given by the initial solution (nearest neighbor). The percentage improvement in this case for emissions was 44.21% and for costs 3.88%.

Among the limitations observed in the development of the present work, we can highlight the running time that was used as a stop criterion, since the maximum execution time was 2 hours. However, if more time was given, the results could be improved. Therefore, it is suggested to make the code more efficient, so that in less time it can obtain better results.

For future studies it is recommended to hybridize the PAES algorithm with other metaheuristics that can be found in the literature such as the memetic algorithm, GRASP, among others, and compare the effectiveness of both methods used. Finally, it is also proposed to solve the stochastic version of the problem by including stochastic demands, emissions and times of travelling.

References

Alizadeh Foroutan, R., Rezaeian, J., & Mahdavi, I. (2020). Green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods. *Applied Soft Computing Journal*, *94*, 106462. https://doi.org/10.1016/j.asoc.2020.106462.

Asawarungsaengkul, K., Rattanamanee, T., & Wuttipornpun, T. (2013). A multi-size compartment vehicle routing problem for multi-product distribution: Models and solution procedures. *International Journal of Artificial Intelligence*, *11*(13 A), 237-256.

Asghari, M., & Mirzapour Al-e-hashem, S. M. J. (2021). Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, *231*(August 2020), 107899. https://doi.org/10.1016/j.ijpe.2020.107899. https://doi.org/10.1016/j.ijpe.2020.107899

Azadeh, A., & Farrokhi-Asl, H. (2019). The close-open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles. *Transportation Letters*, *11*(2), 78-92. https://doi.org/10.1080/19427867.2016.1274468.

Bianchic, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., & Schiavinotto, T. (2004). Metaheuristics for the vehicle routing problem with stochastic demands. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3242*(1), 450-460. https://doi.org/10.1007/978-3-540-30217-9_46

Bouyahyiouy, K., & Bellabdaoui, A. (2021). A mixed-integer linear programming model for the selective full-truckload multi-depot vehicle routing problem with time windows. *Decision Science Letters, 10*(4), 471-486.

Chowmali, W., & Sukto, S. (2020). A novel two-phase approach for solving the multi-compartment vehicle routing problem with a heterogeneous fleet of vehicles: a case study on fuel delivery.*Decision Science Letters, 9*(1), 77-90.

Chowmali, W & Sukto, S. (2021). A hybrid FJA-ALNS algorithm for solving the multi-compartment vehicle routing problem with a heterogeneous fleet of vehicles for the fuel delivery problem. *Decision Science Letters, 10*(4), 497-510.

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, *20*(3), 309-318.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *12*(4), 568-581. https://doi.org/10.1287/opre.12.4.568

Cordeau, J. F., & Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, *39*(9), 2033-2050.

Dantzig, G. B., Fulkerson, R., & Johnson, S. (1954). Solution of a Large-Scale TSP. In *Journal of the Operations Research Society of America* (pp. 393-403).

Ghorbani, E., Alinaghian, M., Gharehpetian, G. B., Mohammadi, S., & Perboli, G. (2020). A survey on environmentally friendly vehicle routing problem and a proposal of its classification. *Sustainability (Switzerland)*, *12*(21), 1-72. https://doi.org/10.3390/su12219079. https://doi.org/10.3390/su12219079

Glover, F. (1989). Tabu Search-Part I. *ORSA Journal on Computing*, *1*(3), 190-206. https://doi.org/10.1287/ijoc.1.3.190

Hajji, O., Brisset, S., & Brochet, P. (2004). A new tabu search method for optimization with continuous parameters. *IEEE Transactions on Magnetics*, *40*(2), 1184-1187. https://doi.org/10.1109/TMAG.2004.824909.

Hameed, A. S., Aboobaider, B. M., Mutar, M. L., & Choon, N. H. (2020). A new hybrid approach based on discrete differential evolution algorithm to enhancement solutions of quadratic assignment problem. *International Journal of Industrial Engineering Computations*, *11*(1), 51-72. https://doi.org/10.5267/j.ijiec.2019.6.005

Hanum, F., Hadi, M., Aman, A & Bakhtiar, T. (2019). Vehicle routing problems in rice-for-the-poor distribution.*Decision Science Letters , 8*(3), 323-338.

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

Kabcome, P., & Mouktonglang, T. (2015). Vehicle routing problem for multiple product types, compartments, and trips with soft time windows. *International Journal of Mathematics and Mathematical Sciences*, *2015.* https://doi.org/10.1155/2015/126754.

Karakatič, S. (2021). Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. *Expert Systems with Applications*, *164*, 114039.

Keskin, M., Çatay, B., & Laporte, G. (2021). A simulation-based heuristic for the electric vehicle routing problem with

time windows and stochastic waiting times at recharging stations. *Computers and Operations Research*, *125*, 105060. https://doi.org/10.1016/j.cor.2020.105060. https://doi.org/10.1016/j.cor.2020.105060
Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, *8*(2), 149–172. https://doi.org/10.1162/106365600568167
Koç, Ç., & Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing Journal*, *39*, 154-164.
Lee, C. G., Epelman, M. A., White, C. C., & Bozer, Y. A. (2006). A shortest path approach to the multiple- vehicle routing problem with split pick-ups. *Transportation Research Part B: Methodological*, *40*(4), 265-284. https://doi.org/10.1016/j.trb.2004.11.004. https://doi.org/10.1016/j.trb.2004.11.004
Li, Y., Soleimani, H., & Zohal, M. (2019). An improved ant colony optimization algorithm for the multi- depot green vehicle routing problem with multiple objectives. *Journal of Cleaner Production*, *227*, 1161-1172. https://doi.org/10.1016/j.jclepro.2019.03.185
Löffler, M., Desaulniers, G., Irnich, S., & Schneider, M. (2020). Routing electric vehicles with a single recharge per route. *Networks*, *76*(2), 187-205. https://doi.org/10.1002/net.21964.
Mehlawat, M. K., Gupta, P., Khaitan, A., & Pedrycz, W. (2020). A Hybrid Intelligent Approach to Integrated Fuzzy Multiple Depot Capacitated Green Vehicle Routing Problem with Split Delivery and Vehicle Selection. *IEEE Transactions on Fuzzy Systems*, *28*(6), 1155-1166.
Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2021). The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, *279*, 123691.
Olivera, A. (2004). *Heuristics for Vehicle Routing Problems*.
Parchami Afra, A., & Behnamian, J. (2021). Lagrangian heuristic algorithm for green multi-product production routing problem with reverse logistics and remanufacturing. *Journal of Manufacturing Systems*, *58*(PA), 33-43. https://doi.org/10.1016/j.jmsy.2020.11.013.
Rafati, E. (2022). The bullwhip effect in supply chains: Review of recent development. *Journal of Future Sustainability*, *2*(3), 81-84.
Ramos, T. R. R. P., Gomes, M. I., & Barbosa-Póvoa, A. P. (2011). Solving a multi-product, multi-depot vehicle routing problem by a hybrid method. *Livro de Actas Do 15º Congresso Da APDIO IO2011*, 1- 13.
Samsuddin, S., Shahizan Othman, M., & Mi Yusuf, L. (2020). Utilizing Ant Colony Optimization and Intelligent Water Drop for Solving Multi Depot Vehicle Routing Problem. *IOP Conference Series: Materials Science and Engineering*, *864*(1). https://doi.org/10.1088/1757-899X/864/1/012095. https://doi.org/10.1088/1757-899X/864/1/012095.
Sawik, B., Faulin, J., & Pérez-Bernabeu, E. (2017). A Multicriteria Analysis for the Green VRP: A Case Discussion for the Distribution Problem of a Spanish Retailer. *Transportation Research Procedia*, *22*, 305-313. https://doi.org/10.1016/j.trpro.2017.03.037. https://doi.org/10.1016/j.trpro.2017.03.037
Shi, Y., Lv, L., Hu, F., & Han, Q. (2020). A heuristic solution method for multi-depot vehicle routing-based waste collection problems. *Applied Sciences (Switzerland)*, *10*(7). https://doi.org/10.3390/app10072403. https://doi.org/10.3390/app10072403.
Subramanian, A., Drummond, L. M. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Computers and Operations Research*, *37*(11), 1899-1911. https://doi.org/10.1016/j.cor.2009.10.011.
Sumichras, R. T., & Markham, I. S. (1995). A heuristic and lower bound for a multi-depot routing problem. *Computers & Operations Research*, *22*(10), 1047-1056.
Tavakkoli-Moghaddam, R., Raziei, Z., & Tabrizian, S. (2015). Solving a bi-objective multi-product vehicle routing problem with heterogeneous fleets under an uncertainty condition. *International Journal of Transportation Engineering*, *3*(3), 207-225.
Toro O, E. M., Escobar Z, A. H., & Granada E, M. (2016). Literature review on the vehicle routing problem in the green transportation context. *Luna Azul*, (42), 362-387.
Wahyuningsih, S., & Satyananda, D. (Improvement of solution using local search method by perturbation on VRPTW variants. *Journal of Physics: Conference Series*, *1581*(1), 0-7. https://doi.org/10.1088/1742- 6596/1581/1/012004
Wofuru-Nyenke, O., & Briggs, T. (2022). Predicting demand in a bottled water supply chain using classical time series forecasting models. *Journal of Future Sustainability*, *2*(2), 65-80.
Zhang, Y., & Chen, X. D. (2014). An optimization model for the vehicle routing problem in multiproduct frozen food delivery. *Journal of Applied Research and Technology*, *12*(2), 239-250.
Zheng, S. (2019). Solving Vehicle Routing Problem: A Big Data Analytic Approach. *IEEE Access*, *7*, 169565- 169570. https://doi.org/10.1109/ACCESS.2019.2955250