DOI: http://dx.doi.org/10.11144/Javeriana.iyu22-1.sact

Solving the Assignment of Customers to Trucks and Visit Days in a Periodic Routing Real-World Case[1]

Solución a la asignación de clientes a camiones y días de visita en un caso real de ruteo periódico[2]

Andrés Felipe Duque Correa[3]
María Gulnara Baldoquín de la Peña[4]

How to cite this article:

A. Duque-Correa, M. G. Baldoquín de la Peña "Solving the Assignment of Customers to Trucks and Visit Days in a Periodic Routing Real-World Case," Ing. Univ., vol. 22, no. 1, 2018. http://dx.doi.org/10.11144/Javeriana.iyu22-1.sact

[1] Submitted on: April 23th, 2017. Accepted on: November 11th, 2017. This article is derived from a Master's thesis in Applied Mathematics, Universidad EAFIT, Colombia.

[2] Fecha de recepción: 23 de abril de 2017. Fecha de aceptación: 11 de noviembre de 2017. Este artículo se deriva de una tesis de Maestría en Matemáticas Aplicadas de la Universidad EAFIT, Colombia.

[3] Ingeniero Financiero, Universidad de Medellín. MSc. en Matemáticas Aplicadas, Universidad EAFIT. Grupo de investigación en Análisis Funcional y Aplicaciones, Universidad EAFIT, Medellín, Colombia. E-mail: aduqueco@eafit.edu.co

[4] Matemática, Universidad Técnica de la Habana. Doctora en Matemáticas Aplicadas, Instituto Superior Politécnico Jose Echavarria. Grupo de investigación en Análisis Funcional y Aplicaciones, Universidad EAFIT, Medellín, Colombia. E-mail: mbaldoqu@eafit.edu.co

Abstract

Introduction: This work proposes a model and two heuristic algorithms to assign customers to trucks and visit days as a first phase in the solution of a real-world routing problem, which is closely related to the PVRP (Periodic Vehicle Routing Problem), but a strategic decision of the company imposes the additional constraint that every customer must always be visited by the same truck. *Methods:* The proposed model will group the customers that are visited the same day by the same truck as close as possible by means of centroid-based clustering. The first proposed heuristic has a constructive stage and three underlying improvement heuristics, while the second uses an exact linear programming algorithm. *Results:* The algorithms are evaluated by instances taken from the literature and generated, taking into account the characteristics presented in the real-world case.

Keywords

logistics; clustering; allocation; heuristics; periodic routing

Resumen

Introducción: Este trabajo propone un modelo y dos algoritmos heurísticos para asignar clientes a camiones y días de visita como una primera fase en la solución de un problema de ruteo, que está estrechamente relacionado con el PVRP (*Periodic Vehicle Routing Problem*), aunque una decisión estratégica de la compañía impone la restricción adicional, de que cada cliente debe ser siempre visitado por el mismo camión. *Métodos:* el modelo propuesto tiene como objetivo agrupar a los clientes que son visitados el mismo día por el mismo camión lo más cerca posible por medio de *clustering* basado en centroides. La primera heurística propuesta tiene una etapa constructiva y tres heurísticas subyacentes de mejora, mientras que la segunda usa un algoritmo de programación lineal exacto. *Resultados:* los algoritmos son evaluados por instancias tomadas de la literatura y generadas, teniendo en cuenta las características presentadas en el caso real abordado.

Palabras clave

logística; clustering; distribución; heurísticas; ruteo periódico.

1. Introduction

A common strategy for solving routing problems is the partitioning of the problem into different phases. For instance, in the VRP (vehicle routing problem) the "cluster first-route second" strategy is widely used, where each cluster represents the customers assigned to one truck. In [1], the clusters were created by solving a GAP (generalized assignment problem), while the sweep algorithms presented in [2] and [3] have a first phase of clustering and a second phase of solving a TSP (traveling salesman problem) for each cluster. In [4], two clustering phases are performed before the routing. For solving the MDVRP (multi-depot vehicle routing problem), in [5] and [6], a clustering phase is performed to assign customers to depots. For the SB-VRP (swap body vehicle routing problem), [7] creates independent clusters composed by customers and swap locations. For the PVRP (periodic vehicle routing problem), it is common to solve an allocation phase by assigning visit days to customers and then solve a VRP for each day. In [8], the customers are assigned to days by making compact clusters for each day using three statistical measures. In [9], the assignment of customers to days was made by adapting the GAP of [1] to the PVRP, while in [10] and [11], integer linear models were proposed to assign customers to visit days to minimize the maximum demand in each day, but there is no consideration of the geographical coordinates of the customers. In this paper, we bring together the truck and visit day assignment phases of our particular problem, which we will usually refer to as the clustering and allocation phases; consequently, each cluster is associated with one truck.

Problem description

The problem was introduced in [12] and [13] and has not been found in the literature reviewed, although we can find a variety of related problems such as the PVRP and the PTSP (periodic traveling salesman problem). The reader is referred to [14] for a complete review of periodic routing. The problem characteristics are as follows: A company has a set of customers that must be visited one or several times every month, according to a specific frequency determined by their sales volume. There are 4 types of visit frequencies: weekly (the same day each week, for example, every Thursday); biweekly (twice a week, which must be Monday and Thursday or Tuesday and Friday); bimonthly (twice a month, in the first and third weeks or the second and fourth weeks, but always on the same day of the week); and monthly. The company has nine trucks to visit the customers; each one travels from a central depot to which it must return after its journey. An important strategic decision of the company imposes the restriction that each customer must always be visited by the same truck. A month is considered to be 4 weeks with 20 days of delivery, from Monday to Friday. The capacity of the trucks is higher than the demand of the maximum number of customers who can be visited in a day. Although the trucks have a limited capacity, the current restriction is the truck's total available travel time, given by the maximum working time of sellers each day. Since the creation of routes is not part of the scope of this work, a feasible solution of the clustering and allocation phases will be the assignment of each customer to a certain truck and an allowable combination of visit days associated with the frequency type. In many practical problems similar to this, it is important to not only minimize the total distance traveled by all the trucks in the planning horizon but also minimize the maximum number of daily visits made by each truck.

Note that this problem is a PVRP, with the additional constraint that every customer must always be visited by the same truck. It also differs in some other aspects from the PVRP and PTSP benchmark instances proposed by [15], [16], [11], and [17] because, in those instances, the allowable combinations of visit days is usually smaller than 6, but here, we face customers with a much greater set of possible combinations. This is the main reason for not making a random initial assignment of allowable combinations of visit days, as it is done in [17] and [18]. These two methods are very powerful for the literature instances since, in the local search procedures, customers change the combinations of visit days among a small set of possibilities, and the random initial assignment of visit days is not an issue. If you have to address larger sets, the initial assignment of visit days to customers requires more attention.

Another reason to present a novel way to perform the initial assignment of vehicles and visiting days was mentioned in [19]. Sometimes, there is a need for an aesthetic quality in the set of routes and to partition the space in certain regions for each vehicle, which can be useful in cases where it is desirable for the drivers to become familiar with certain regions.

For our problem, efforts in solving the clustering and allocation phases separately have been done. In [13], various methods were used for solving the clustering phase, following a philosophy similar to the one proposed in [20] and [21]. They adapted a k-means method using centroids to provide a clear physical interpretation of a real-world problem (the grouping of geographical areas) and provided the geometric center of each cluster. In [12], an allocation phase was solved inside each cluster, where customers were assigned to visiting days, restricting the maximum number of customers to visit daily. In this paper, we propose a model and two heuristic solution approaches to solve these two phases simultaneously. The resulting model is an extension of the one presented in [12], whose objective function adopts the idea of the well-known MSSC (minimum sum of squares clustering) problem; see [22]. We do not consider the square distances but the distances themselves.

The main contributions of this paper can be summarized as follows: (i) We propose a novel heuristic to solve a real-world clustering and allocation problem as a first stage of a periodic routing problem.

(ii) We show that if the constraint that every customer must always be visited by the same truck in a PVRP is added, the problem can be transformed back to a classical PVRP with one truck, a wider planning horizon, and a greater set of allowable combinations of visit days. (iii) We propose two heuristics for solving overlapping centroid-based clustering. Because the centroid is a notion of a Euclidean distance space, the scope of this work is limited in this way.

The remainder of this paper is organized as follows. In Section 2, we present the proposed mathematical formulation, and in Section 3, we describe two solution approaches. In Section 4, we show the computational results, and in Section 5, we show the analysis of the results. Finally, in Section 6, our main conclusions and future works are described.

2. Mathematical formulation

We use the letters L, J and K to denote the sets of customers, days, and clusters; $F = \{M, BM, W, BW\}$ is the set of visit frequencies, where M is monthly, BM bimonthly, W weekly, and BW biweekly. Each customer $l \in L$ has a defined visit frequency $f \in F$ and a set C_f of allowable combinations of visit days associated with the frequency f. L_f is the set of customers with frequency f, and $\vec{V}_l$ is the coordinate vector of customer l. The problem consists of assigning each customer to one cluster $k \in K$ and one allowable combination of visit days $c \in C_f$ in such a way that the customers visited by the same truck in the same day are as close as possible; the closeness criterion is explained later.

The characteristics of the problem given in Section 1 are as follows: $K = \{1,2, \ldots, 9\}$ (nine trucks),$J = \{1,2, \ldots, 20\}$ (20 days), and C_f is given by equations (1)-(3). For instance, $C_{BM} = \{\{1,11\}, \{2, 12\}, \ldots, \{10,20\}\}$ because, as we mentioned in Section 1, the customers with bimonthly frequency must be visited on the first and third Mondays (days 1 and 11) or on the first and third Tuesdays (days 2 and 12), etc.

$$C_f = \bigcup_{\forall p} C_f^p \tag{1}$$

Where

$$C_f^p \begin{cases} \bigcup_{i=0}^{3} \{5i+p, 3+5i+p\}\ p \in J_{BW}, & \text{,if} f = BW \\ \bigcup_{i=0}^{3} \{5i+p\}\ p \in J_W & \text{,if} f = W \\ \bigcup_{i=0}^{1} \{10i+p\}, p \in J_{BM} & \text{,if} f = BM \\ \{p\}\ p \in J_M \end{cases} \tag{2}$$

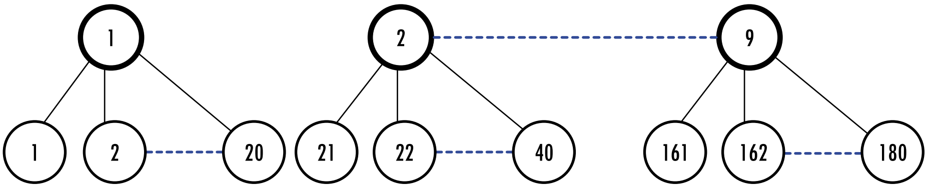
$$J_f = \begin{cases} \{1,2\} & \text{,if} f = BW \\ \{1,2,3,4,5\} & \text{,if} f = W \\ \{1,2,...,10\} & \text{,if} f = BM \\ \{1,2,...,20\} & \text{,if} f = M \end{cases} \tag{3}$$

Our general idea for bringing together the clustering and allocation phases lies in the fact that we can represent the customers of each cluster (truck) that are visited in the same day by a sub-cluster. The sub-clusters are denoted by an index n, which depends on the values of $j \in J$ and $\in K$, $n(j, k) = j + 20(k + 1)$, that is, one sub-cluster n corresponds to each day and each cluster. In Figure 1, the top circles represent the clusters and the low circles the sub-clusters. Note that each cluster has 20 associated sub-clusters that represent the 20 days of the planning horizon.

The previous transformation forces us to change the set of allowable combination visit days (C_f) to a set of allowable sub-cluster combinations $C_f^{'}$ associated with each frequency. A compact expression for $C_f^{'}$ is given by equations (4)-(6).

$$C_f^{'} = \bigcup_{\forall p \forall k} C_f^{'pk} \tag{4}$$

Figure 1. Relationship of clusters and sub-clusters



Source: Author's own elaboration

Where

$$C_f^{'pk} = \begin{cases} \bigcup_{i=0}^{3} \{5i+p+20(k-1), 3+5i+p+20(k-1)\} k \in K, p \in J_{BW} & \text{,if } f = BW \\ \bigcup_{i=0}^{3} \{5i+p+20(k-1)\} k \in K, p \in J_{W} & \text{,if } f = W \\ \bigcup_{i=0}^{1} \{10i+p+20(k-1)\} k \in K, p \in J_{M} & \text{,if } f = M \\ \{p+20(k-1)\} k \in K, p \in J_{M} & \text{,if } f = M \end{cases} \tag{5}$$

$$J_f = \begin{cases} \{1,2\} & \text{,if } f = BW \\ \{1,2,3,4,5\} & \text{,if } f = W \\ \{1,2,...,10\} & \text{,if } f = Q \\ \{1,2,...,20\} & \text{,if } f = M \end{cases} \tag{6}$$

Now we obtain a classical PVRP with one truck, 120 days, and a new set of allowable combinations of visit days $C_f^{'}$ for each visit frequency *f*. In creating the sub-clusters, we group the customers by a closeness criterion. A common one is to minimize the sum of the distances of all customers to all of the sub-cluster centroids $\left(\vec{C}_n\right)$ that they belong to. In addition to the variables $\left(\vec{C}_n\right)$, we define the binary variables y_{lc} equal to 1 if the allowable sub-cluster combination $c \in C_f^{'}$ is assigned to customer *l* and zero otherwise. We use binary constants a_{nc} equal to 1 if sub-cluster *n* belongs to $c \in C_f^{'}$ and zero otherwise; the maximum number of customers allowed in each sub-cluster is denoted by *w.* Because *w* is minimized, it must be the nearest integer greater than or equal to the average number of customers by sub-cluster.

The model is as follows:

Objective function:

$$\text{Minimize} \sum_{n=1}^{180} \sum_{\forall f \in F} \sum_{\forall c \in C_f^{'}} a_{nc} y_{lc} \left\| \vec{V}_l - \vec{C}_n \right\| \qquad (7)$$

Subject to:

$$\sum_{\forall c \in C_f^{'}} y_{lc} = 1 \quad \forall f \in F, l \in L_f, n \qquad (8)$$

$$\vec{C}_n = \frac{\sum_{\forall f \in F} \sum_{\forall l \in L_f} \sum_{\forall c \in C_f^{'}} a_{nc} y_{lc} \vec{V}_l}{\sum_{\forall f \in F} \sum_{\forall l \in L_f} \sum_{\forall c \in C_f^{'}} a_{nc} y_{lc}} \forall n \qquad (9)$$

$$\sum_{\forall f \in F} \sum_{\forall l \in L_f} \sum_{\forall c \in C_f^{'}} a_{nc} y_{lc} \le w \quad \forall n \qquad (10)$$

$$y_{lc} \in \{0,\ 1\} \quad f \in F, \forall l \in L_f, c \in C_f^{'} \qquad (11)$$

Constraints (8) guarantee that each customer is assigned to one allowable combination of sub-clusters. Constraints (9) are the equations of the sub-cluster centroids, and finally, the role of constraints (10) is to impose a maximum number of customers that belong to each sub-cluster.

The resulting model is non-linear and combinatory, with continuous $\left(\vec{C}_n\right)$ and binary variables (y_{ln}). In the next section, we propose two heuristic approaches for obtaining the solution.

3. Solution approach

Before presenting the heuristics, additional notation is introduced in Table 1. We drop the variables y_{lc}; instead, we use x_{ln} equal to 1 if customer l belongs to sub-cluster n and zero otherwise. An important feature of model (7)-(11) is that it becomes an integer linear program (ILP) if the variables $\left(\vec{C}_n\right)$ are fixed as parameters. This modified model is denoted by M_{fix}.

Table 1. Notation used in the algorithms

x_{ln}	Equal to 1 if customer l belongs to sub-cluster n, $x_{ln} = a_{nc}y_{lc}$
X	Current solution matrix where $X_{l,n} = x_{ln}$
X*	Best solution found
	Centroid of sub-cluster n in the best solution found
of	Value of the objective function of the current solution
*of**	Best value found for the objective function
M_{fix}	Model (7)-(11), fixing $\vec{C}_n$ $\forall n$ as parameters

Source: Author's own elaboration

3.1. Heuristic 1

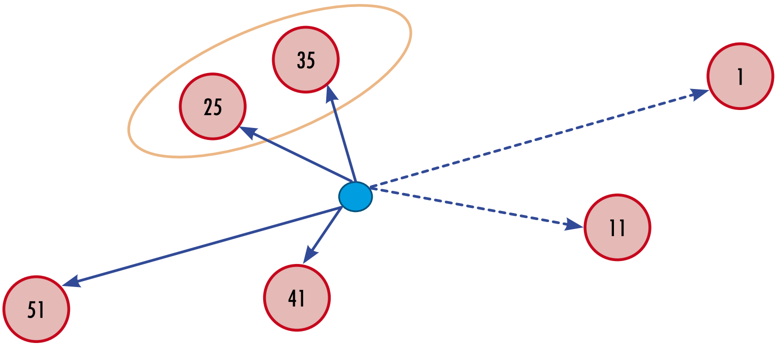
Heuristic 1 (Algorithm 1), includes four sub-heuristics; the first one (Algorithm 2) builds an initial solution, while the others (Algorithms 3-5) try to improve the solutions obtained.

```
Algorithm 1 Heuristic 1
Apply Algorithm 2
while Stopping criteria is not satisfied do
    Randomly select one algorithm among Algorithms 3-5 and apply it to the current best solution
end while
```

3.1.1. Constructive heuristic (Algorithm 2)

The core of the constructive heuristic (Algorithm 2) is to assign each customer to its closest $c \in C'_f$ that is not saturated, where closest means that the average distance to all sub-clusters $n \in c$ is minimum, and saturated means that at least one $n \in c$ already has w customers. A graphical example of this procedure is given in Figure 2, where the red circles represent the sub-cluster centroids, and the blue circle is a customer $l \in L_{BM}$. Suppose that pairs $\{1,11\}$, $\{25,35\}$ and $\{41,51\}$ are the only $c \in C'_{BM}$ that are not saturated. Therefore, the customer is assigned to c $= \{25,35\}$ because, on average, the customer is closer to $n = 25$ and $n = 35$ than to the other two pairs. Regarding the algorithm's structure, the inner while loop performs the previous procedure, calculates the new centroids, and repeats these two steps interactively. The outer while loop explores other regions in the solution space by randomly restarting all centroids. Note that in the first iteration of the outer while loop, M_{fix} is solved. The purpose of this is to ensure that the algorithm starts from a feasible solution.

Figure 2. Customer assignment to $c \in C'_{BM}$



Source: Author's own elaboration

Algorithm 2. Constructive Heuristic

```
Ensure: $X^*$, $of^*$
$of^* \leftarrow \infty$
$i \leftarrow 1$
$X \leftarrow 0$
wile $i \leq \eta$ do
    Randomly generate 180 seeds as explained in Section 4.4.2
    if $i = 1$ then
        Solve $M_{fix}$ as a ILP, and compute the new $\vec{C}_n$
    end if
    repeat
      for all $f \in F = BW, W, BM, M$ do
          for all $l \in L_f$ do
              Remove customer l from its current sub-clusters combination, and assign the customer
    to its closest set of allowable sub-clusters combination $c \in C'_f$ that is not saturated
          end for
        end for
```

Algorithm 2. Constructive Heuristic

```
        if $of < of^*$ then
            $of^* = of$
            $X^* = X$
        end if
        Compute $\vec{C}_n$ (∀n)
    until Local minimum is reach
    i ← i + 1
end while
```

3.1.2. Improvement heuristics

Algorithms 3-4 are shaking procedures, with the common idea of making changes in the coordinates of one or several centroids, followed by performing the same routine as the inner while loop of Algorithm 2, henceforth referred to as Routine 1. Algorithm 3 selects one sub-cluster randomly (*n*’) and sets up the vector **H** with its closest sub-clusters, where the closeness criterion is the distance between the sub-cluster centroids. Then, one sub-cluster *n*’’∈ **H** is taken, and the coordinates of the centroids of *n*’ and *n*’’ are swapped, followed by Routine 1. Algorithm 4 selects one sub-cluster randomly (*n*0) and one of its customers and changes the centroid’s coordinates of n_0 and its ϕ closest sub-clusters for the selected customer coordinates, followed again by Routine 1. Finally Algorithm 5 is a local search that swaps the assigned sub-cluster combination between two customers that have the same visit frequency.

Algorithm 3.

```
Require: $X^*$,$of^*$, ρ
while Stopping criteria is not satisfied do
    Randomly selects one sub-cluster ($n'$)
    H ← ρ closest sub-clusters to $n'$
    for all $n'' \in$ H do
        Swap $\left(\vec{C}_{n'}, \vec{C}_{n''}\right)$
        Routine 1
        if $of < of^*$ then
            $of^* = of$
            $X^* = X$
        end if
    end for
end while
```

Algorithm 4.

```
Require: $X^*$, $of^*$, $\phi$
while Stopping criteria is not satisfied do
    Randomly select one sub-cluster ($n_0$), one of its customers ($l$), and its $\phi$ closest sub-clusters ($n_1$, $n_2$,..., $n_\phi$)
    $\vec{C}_{n_r} \leftarrow \vec{V}_l \quad r = 0,1,\ldots,\phi$
    Routine 1
    if $of < of^*$ then
        $of^* = of$
        $X^* = X$
    end if
end while
```

3.2. Heuristic 2

The second solution method (Algorithm 6) is an adaptation of the heuristic presented in [12], with m any more binary variables, as it considers all the customers of the company to be visited. The algorithm starts by randomly selecting 180 initial seeds as centroids C_n and solving the resulting M_{fix} using adequate software, in our case, Gurobi with AMPL. With the solution, the new centroids are calculated, and a new M_{fix} is solved. These two steps are repeated interactively until a local minimum is reached (we consider that the local minimum is reached if the best solution has not improved more than 0.05%). When the local minimum is reached, new random initial seeds are generated in order to explore other regions in the solution space.

Algorithm 5.

```
Require: $X^*$, $of^*$, $\gamma$
for all $l$ do
    P ← $\gamma$ closets customers to $l$ with the same frequency that do not belong to the same $c \in C'_f$
    for all $l' \in$ P do
        Swap($x_{ln}$, $x_{l'n}$) $\forall n$
        Compute $\vec{C}_n$ for all sub-clusters that have been modified and calculated $of$
        if $of < of^*$ then
            $of^* = of$
            $X^* = X$
        end if
    end for
end for
```

Algorithm 6.

```
Ensure: $X^*$, $of^*$
1: $of^* \leftarrow$ inf
2: $i \leftarrow 1$
3: $X \leftarrow 0$
4: wile i $\leq \eta'$ do
5:     Ramdomly generated 180 seeds as the initial centroids $\vec{C}_n$ as in Section 4.4.2
6:     repeat
7:         Solve $M_{fix}$
8:         Compute $\vec{C}_n$ $\forall n$
9:     until Local minimum is reach
10:    if $of < of^*$ then
11:        $of^* = of$
12:        $X^* = X$
13:    end if
14: end while
```

3.3. Hybrid Heuristic

The best solution obtained by Heuristic 1 is sensitive to further improvement, as M_{fix} can be solved with the centroids of the best solution found. We call the Hybrid Heuristic the one that takes the solution obtained in Heuristic 1 and applies the lines 6-9 of Heuristic 2.

4. Computational experiments

4.1. Test instances

A set of instances from [13] are considered, taking into account the characteristics of the company. The instances are differentiated by two main criteria: first, the geographical distribution of the customers in $\mathbb{R}^2$, where class r is those where the customers are uniformly distributed, class c those with dense clusters of customers, and rc a combination of r and c; second, the different frequencies of distribution relative to the total number of customers, which are denoted by 1, 2 and 3.

We use the notation r*, c* and rc* when we refer to instances of type r, c and rc with any of the different frequency types of distribution (1, 2 or 3). Analogously, *1, *2 and *3 are associated with instances with specific frequencies of distribution (1, 2 or 3), regardless of the configuration of customers with regard to their geographical distribution (r, c or rc).

Instances of type *1 have the lowest percentage of customers with monthly frequencies compared to the other two types of distributions, with an intermediate percentage of customers with biweekly and weekly frequency compared to the other two types of distributions. Instances of type *2 do not have customers with biweekly frequencies and have the highest percentage of customers with monthly and weekly frequencies and the lowest percentage of customers with bimonthly frequencies compared to the other two types of distributions; the most significant differences are in monthly distributions. Instances of type * 3 have the highest percentage of customers with biweekly frequencies compared to the other two types of distributions. These characteristics are shown in Table 2. The last row of Table 2 shows one type of instance (rc3a) very similar to instance rc3. It is constructed from rc3 by changing the visit frequency of 3 customers from monthly to bi-weekly. The number of total visits to be made increases, but the gap of the average of visits to the nearest smaller integer increases.

Table 2. Instance characteristics

Instance	Days	Truchs	Visits	Average	w	Customers by frequencies (%)			
						BW	W	BM	M
r1	20	9	3103	17.23	18	1.7	51.18	43	3.5
c1	20	9	3113	17.29	18	1.5	52.7	42.7	3.1
rc1	20	9	3195	17.75	18	1.8	56	38.9	3.3
r2	20	9	2933	16.29	17	0	61.9	7.6	30.5
c2	20	9	2775	15.41	16	0	56.8	7.1	36.1
rc2	20	9	2898	16.1	17	0	60.6	8	31.4
r3	20	9	3204	17.8	18	6.2	45.5	40.5	7.8
c3	20	9	3222	17.9	18	6.7	45.2	39.7	8.4
rc3	20	9	3223	17.9	18	5.9	47.3	39.1	7.7
rc3a	20	9	3244	18.02	19	6.2	47.3	39.1	7.4

Source: Author's own elaboration

4.2. Parameter settings, stopping criteria and run times

4.2.1. Heuristic 1

The algorithm's parameters must be adjusted in consideration of the size of the instance. In our tested instances, we found efficient results for $\eta = 100$, $\rho = 8$, $\phi = 6$, $\gamma = 5$.

Regarding the stopping criteria for Algorithm 1, we found that repeating line 3 eight times was sufficient; above this value, we did not obtain a significant improvement in the solution in contrast to the time spent, while Algorithms 3-4 stop if the objective function has not improved at least 0.05% within 50 seconds. With this configuration, we ran Algorithm 2 10 times for all instances. The algorithm was coded in MATLAB with an average run time of 15 minutes.

4.2.2. Heuristic 2

To make the times comparable between both heuristics, we report the results when setting $\eta' = 80$, although it does not provide considerably better results that those obtained with a lower η'. This is shown in Table 4.

4.3. Numerical Results

Table 3 reports the results obtained for each of the instances. In column 2, the best value of the Hybrid Heuristic, in column 3, the best value of Heuristic 1, and in columns 4 and 5, the average and standard deviation with respect to the solutions of Heuristic 1 are shown. Columns 6 to 8 show the results of Heuristic 2. The last column reflects the gap in the average values obtained by both Heuristics 1 and 2.

Table 3. Results of heuristics 1 and 2

Heuristic 1					Heuristic 2			GAP (%)
Instance	Best Hybrid	Best	Mean	SD	Best	Mean	SD	
r1	83281	83504	84172	546	92844	99359	2606	18.0
c1	60081	61210	63960	1966	69259	793304	3310	24.0
rc1	80709	84017	85440	1288	84541	94573	4500	10.7
r2	77017	77112	78292	825	79709	83620	1715	6.8
c2	51392	556896	58255	1089	51532	58412	2246	0.3
rc2	63338	64861	66340	1240	66532	70769	1926	6.7
r3	101918	104830	107047	1650	106762	114182	3708	6.7
c3	95789	100390	104903	3290	90373	102667	4517	2.2
rc3	88222	91980	98084	4237	96065	107007	4523	9.1
rc3a	85072	85769	86980	1231	93073	103025	3996	0.0

Source: Author’s own elaboration

Table 4. Results with different η' values in Heuristic 2

	Best		Mean		GAP	
Instance	40 iter	80 iter	40 iter	80 iter	Min	Mean
r1	92844	92844	99466	99359	0	0.11
c1	69259	69259	79587	79587	0	0.36
rc1	87098	84541	96321	94573	3.02	1.85
r2	80589	79709	83630	83620	1.1	0.01
c2	54517	51532	58069	58412	5.79	-0.59
rc2	66532	66532	70549	70769	0	-0.31
r3	107269	106752	114002	114182	0.48	-0.16
c3	90373	90373	102711	102667	0	0.04
rc3	96065	96065	105850	107007	0	-1.08

Source: Author's own elaboration

Table 5 reports the results of the Hybrid Heuristic when relaxing *w* (*w* – *w* + 1) during Heuristic 2 and then solving M_{fix} with the original *w,* compared with the results without relaxing *w.* The relaxation is done to increase the possible movements of Heuristic 1, as the heuristic can make more changes to the solution, especially in instances * 3, where the average is closer to *w* than in the other instances.

Table 5. Results of relaxing *w*

	Relaxing w (*w* ← *w* + 1)			Without relaxing *w*			Gap	
Instance	Best value	Mean	SD	Best value	Mean	SD	Best value	Mean
r1	83561	83820	349	83281	83689	325	-0.34	0.16
c1	60221	61780	1537	60081	61432	1670	-0.23	-0.56
rc1	77083	77850	871	80709	82341	1321	4.7	5.77
r2	77231	77850	641	77017	77823	812	-0.28	-0.03
c2	52610	54458	1235	51392	53965	1424	-2.31	-0.91
rc2	63230	63976	714	63338	63871	862	0.17	-0.16
r3	99841	100350	804	102561	104830	1202	2.08	4.46
c3	85090	89471	4244	99085	100390	2981	12.57	12.2
rc3	87852	92532	3860	92140	100671	4320	9.05	8.24
re3a	85230	85633	779	9522	91980	5210	9.05	8.24

Source: Author's own elaboration

4.4. Sensitivity analysis

4.4.1. Parameters ρ, ϕ, γ

To analyze the importance of the values of ρ and ϕ, we test different values of these parameters by running Algorithm 1 solely with Algorithm 3 or 4 as improvement procedures. For this purpose, we choose instances r1, rc2 and c3, which provide different characteristics regarding the geographical position of the customers and the percentage of customers with each frequency. In Tables 6 and 7, we report the best, average and worst deviations from the best results of Heuristic 1 in Table 3 over 10 runs. The parameter γ is just to speed up the algorithm, because there might be some customers that are too far from each other, and it is not worth doing a permutation with them. We found no typical behavior in the results of Table 6; nevertheless, $\rho = 8$ is the best value in general. Conversely, ϕ shows the best results when set equal to 6.

Table 6. Sensitivity analysis

		r1			rc2			c3	
ρ Value	Mean	best	Worst	Mean	best	Worst	Mean	best	Worst
4	3.24%	2.03%	4.81%	7.42%	5.81%	9.24%	10.54%	7.59%	12.20%
5	2.78%	2.01%	3.96%	6.75%	4.57%	8.35%	8.97%	6.03%	12.79%
6	**2.16%**	1.13%	3.34%	5.51%	3.34%	8.19%	5.97%	0.93%	**7.24%**
7	2.26%	0.83%	**3.33%**	5.14%	3.32%	**6.76%**	6.25%	3.12%	0.88%
8	2.33%	**-0.89%**	3.99%	**4.95%**	**2.40%**	6.87%	5.17%	0.13%	10.05%
9	2.54%	1.80%	3.90%	5.61%	3.10%	6.90%	5.24%	1.10%	8.32%
10	2.97%	2.06%	3.98%	5.14%	3.23%	7.01%	**5.05%**	**0.52%**	11.80%

Source: Author's own elaboration

Table 7. φ sensitivity analysis

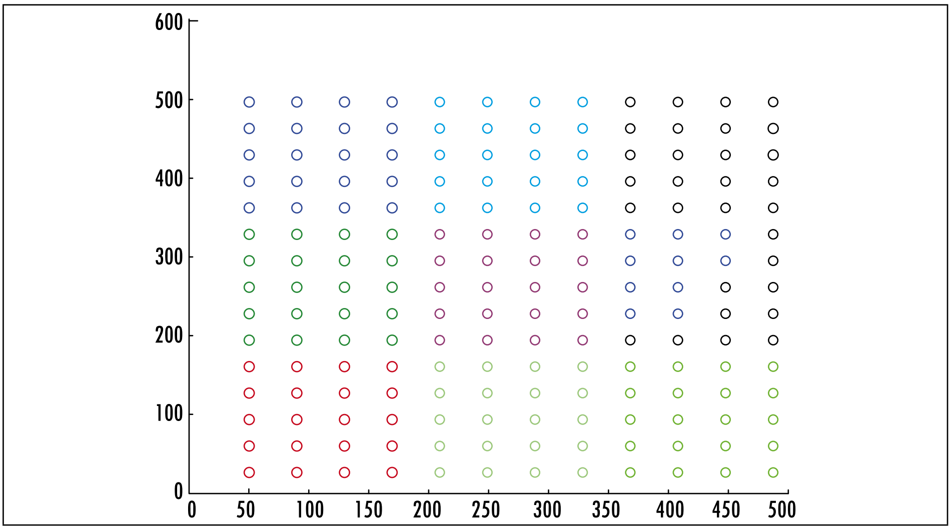
	r1			rc2			c3		
ϕ Value	Mean	Best	Worst	Mean	Best	Worst	Mean	Best	Worst
0	6.98%	5.61%	8.09%	8.34%	6.30%	11.10%	13.00%	8.38%	18.35%
1	4.20%	3.40%	5.23%	6.78%	4.12%	9.23%	10.45%	4.67%	17.40%
2	2.51%	1.86%	2.76%	5.17%	1.71%	8.34%	8.26%	2.48%	18.85%
3	2.49%	0.39%	4.47%	3.85%	2.32%	5.74%	5.51%	1.37%	10.49%
4	1.38%	-0.26%	3.07%	3.28%	1.92%	5.18%	5.16%	0.98%	13.58%
5	1.69%	0.83%	2.47%	2.81%	0.43%	6.64%	4.85%	1.30%	11.75%
6	1.04%	-0.68%	2.23%	2.30%	0.10%	4.27%	3.50%	-1.41%	10.70%
7	1.10%	-0.29%	2.09%	2.43%	1.15%	3.27%	4.63%	1.28%	9.30%
8	1.65%	0.73%	2.82%	2.72%	1.89%	4.76%	3.87%	0.68%	10.42%

Source: Author's own elaboration

4.4.2. Initial seeds

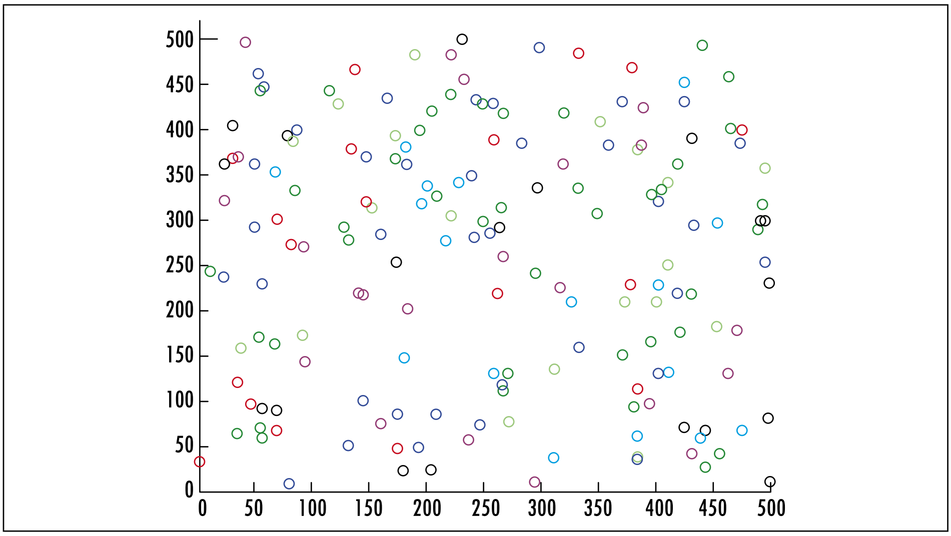
The initial seed location has proven to have an important role in the final results. In Figure 3, we show our initial arrangement where the 20 initial seeds for each truck are placed close to each other. In Figure 4, the 20 initial seeds for each truck are randomly spread over all regions. The final results of both initial arrangements are shown in Figs. 5 and 6, where the customers assigned to the same day are marked in the same color (for a better visualization, we just show the assignation for trucks 1 to 5). The results are more visually appealing, and the objective function is usually 5% to 10% better. Even though the creation of the routes is not part of the scope of this work, initial experiments have shown lower route costs when the seeds are arrange as in Figure 3.

Figure 3. Organized initial seeds



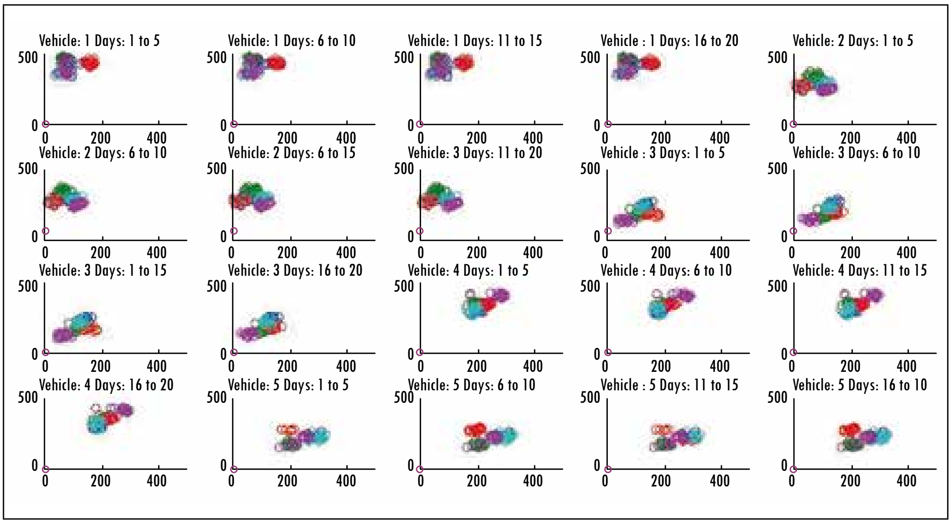
Source: Author's own elaboration

Figure 4. Random initial seeds



Source: Author's own elaboration

Figure 5. Instance c3 solution with organized initial seeds


Vehicle: 1 Days: 1 to 5

Vehicle: 1 Days: 6 to 10

Vehicle: 1 Days: 11 to 15

Vehicle : 1 Days: 16 to 20

Vehicle: 2 Days: 1 to 5

Vehicle: 2 Days: 6 to 10

Vehicle: 2 Days: 6 to 15

Vehicle: 3 Days: 11 to 20

Vehicle : 3 Days: 1 to 5

Vehicle: 3 Days: 6 to 10

Vehicle: 3 Days: 1 to 15

Vehicle: 3 Days: 16 to 20

Vehicle: 4 Days: 1 to 5

Vehicle : 4 Days: 6 to 10

Vehicle: 4 Days: 11 to 15

Vehicle: 4 Days: 16 to 20

Vehicle: 5 Days: 1 to 5

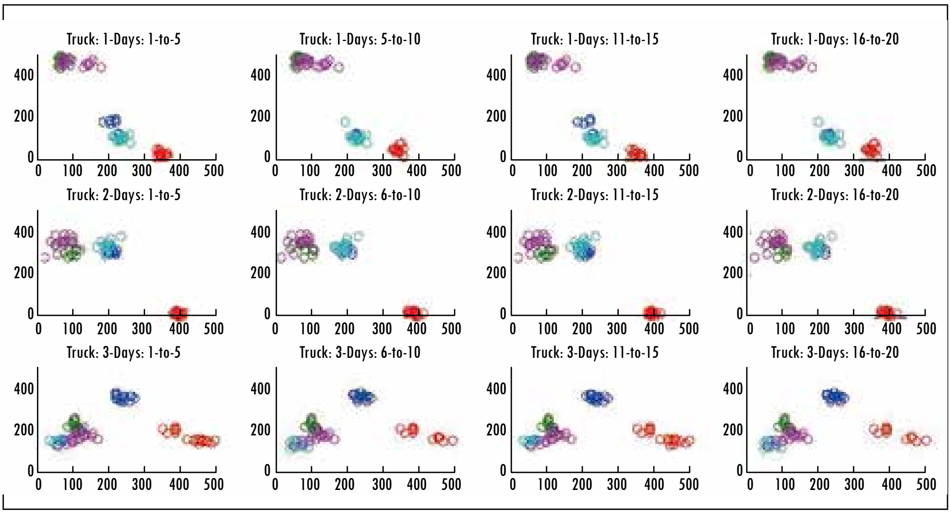
Vehicle: 5 Days: 6 to 10

Vehicle : 5 Days: 11 to 15

Vehicle: 5 Days: 16 to 10

Source: Author's own elaboration

Figure 6. Instance c3 solution with random initial seeds



Source: Author's own elaboration

4.5. Computational times

We now present a computational time analysis, introducing some new instances in Table 8. We reduce the number of customers to 400 and the number of trucks to 4 and 8.

Table 8. New instances

Instance	Days	Trucks	Visits	Average	w	Customers by frequencies			
						BW	W	BM	M
r1 (400)a	20	4	1271	7.94	8	2.25	53	43	1.75
c1 (400)a	20	4	1273	7.96	8	2.25	54	40.5	3.25
rc1 (400)a	20	4	1258	7.86	8	2.5	51.25	43.25	3
r1 (400)b	20	8	1271	15.88	16	2.25	53	43	1.75
c1 (400)b	20	8	1273	15.91	16	2.25	54	40.5	3.25
rc1 (400)b	20	8	1258	15.72	16	2.5	51.25	43.25	3.48

Source: Author's own elaboration

Table 9. Results of new instances

	Heuristic 2	Heuristic 1	GAP	Heuristic 1	Heuristic 2
	Best	Best		Average computational	Time (minutes)
r1 (400)a	21302	20963	1.62%	10.23	2.26
c1 (400)a	18675	17832	4.73%	11.18	2.48
rc1 (400)a	20082	20290	-1.3%	10.45	2,61
r1 (400)b	15286	13533	12.95%	12.10	4.32
c1 (400)b	10522	9932	5.94%	11.97	3,95
rc1 (400)b	12663	12442	1.78%	13.03	3.48

Source: Author's own elaboration

Table 10. Solution times for Heuristic 1 for different numbers of customers and trucks

Customers	Trucks	Days	Heuristic 2	Computational time
2000	9	20		28.33
2000	4	20		26.72
1000	9	20		15.17
1000	4	20		13.42
400	9	20		12.05
400	4	20		10.62

Source: Author's own elaboration

5. Discussion

Comparing instances of types * 1, * 2 and * 3, Table 3 shows that taking into account both the best value found and the average of the values of the objective function, they increase in the order c, rc, and r, according to the geographical distribution of customers. The standard deviation (SD) generally has no typical behavior. These results seem logical for the types of distribution of the points and their possible proximity to the associated centroids. A significant exception is found when comparing the results of Heuristic 1 (also Hybrid) with the instances of type * 3, which increase in the order rc, c, and r. However, with other instances similar to those of type * 3, the trend seen in the other examples is reflected.

Comparing the instances of types r *, c * and rc *, Table 3 also shows that for all algorithms implemented, taking into account both the best value found and the average value, the values of the objective function increase in the order 2, 1, and 3 of the frequency distribution. The SD generally behaves in the same way. This could be in relation to the total number of visits to be made, which decreases in the same order.

5.1. Algorithm performance

The best results are obtained by the Hybrid Heuristic when relaxing the upper bound ($w - w + 1$) during the Heuristic 2 phase. Because Heuristic 2 performs better if most of the clusters are not saturated, the algorithm moves more freely through the solution space. Experiments show that if w is too close to the average of customers by sub-cluster, worse results are obtained even though the number of visits may increase, such as in the case of instances rc3 and rc3a.

It is important to note that if all the customers have a monthly frequency of visits and

$w \leftarrow \infty$, both Algorithms 2 and 6 perform as k-means algorithms [23].

Heuristic 2 is the repetition for a given number of iterations of a solver of a mixed integer linear programming (MILP) problem with binary variables (greater percentage) and continuous variables. The solution time depends mainly on the number of binary variables. The binary variables depend on the number of customers and the number of trucks (clusters). When one of these two parameters increases, the size of each MILP problem increases, and the number of binary variables and the solution time of each iteration increase. For example, in the computational experiments carried out, when varying the number of clients from 400 to 1 000 and keeping the remaining parameters fixed, the average time in the solution of each MILP problem varies from 0.02 to 0.05 seconds.

In the same way, setting the number of customers at 1000 and varying the number of trucks from 4 to 9, the average time for the solution of each MILP problem varies from 0.05 to 0.107 seconds. In relation to Heuristic 1 and its execution time, the results are given in Table 10.

In terms of solution quality, by increasing the number of trucks and maintaining the same number of customers, it is logical that the solution improves in both heuristics, decreasing the value obtained from the objective function (Table 9).

5.2. Balancing workload

The model (7)-(11) does not have a lower bound for the number of customers belonging to the sub-clusters; this means that it could be the case that no customer is assigned to a particular sub-cluster. If a more balanced workload among all routes is desired, a lower bound can be introduced into the model. The solution approaches do not change much, for instance, the Hybrid Heuristic can be applied by letting Heuristic 1 remain as is and, at the end, solving M_{fix}

with the lower bound additional constraint. If the average of customers per sub-cluster is very close to *w,* this balance is automatic.

6. Conclusions and future work

We presented novel solution approaches to assign customers to trucks and days in a PVRP like problem with the additional constraint that every customer must always be visited by the same truck.

Our method is very useful for instances with a great number of allowable combinations of visit days. The generalization for capacity constraints in the daily routes is straight forward and can be done in all Heuristics with *w* equal to the maximum capacity.

Future work consists of the creation of routes for each sub-cluster that correspond to a problem of type TSP (traveling salesman problem), which can be solved by a simple insertion heuristic and further improvements in the solutions with local search procedures and meta-heuristics.

References

[1] M. L. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109-124, Jun. 1981. doi: 10.1002/net.3230110205

[2] B. E. Gillett and L. R. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Operations Res.*, vol. 22, no. 2, pp. 340-349, Apr. 1974. doi: 10.1287/opre.22.2.340

[3] N. Suthikarnnarunai, "A sweep algorithm for the mix fleet vehicle routing problem," in *Proc. IMECS 2008*, vol. 2, Hong Kong, 19-21 March, 2008.

[4] K. Shin and S. Han, "A centroid-based heuristic algorithm for the capacitated vehicle routing problem," *Comput. Informat.*, vol. 30, no. 4, pp. 721-732, 2011.

[5] I. Stankovianska, "An effective approach to routing problems," *Scient. Papers Univ. Pardubice. Series D, no. 10 (2006)*, pp. 148-151, 2006.

[6] I. D. Giosa, I. L. Tansini, and I. O. Viera, "New assignment algorithms for the multi-depot vehicle routing problem," *J. Oper. Res. Soc.*, vol. 53, no. 9, pp. 977-984, 2002.

[7] J. J. Miranda-Bront, B. Curcio, I. Méndez-Díaz, A. Montero, F. Pousa, and P. Zabala, "A cluster- first route-second approach for the swap body vehicle routing problem," *Ann. Oper. Res.*, vol. 253, no. 2, pp. 935-956, 2017. doi: 10.1007/s10479-016-2233-1

[8] R. Russell and W. Igo, "An assignment routing problem," *Networks*, vol. 9, no. 1, pp. 1-17, 1979. doi:10.1057=palgrave.jors.2601426

[9] C. C. R. Tan and J. E. Beasley, "A heuristic algorithm for the period vehicle routing problem," *Omega*, vol. 12, no. 5, pp. 497-504, 1984. doi: 10.1016/0305-0483(84)90050-1

[10] S. Coene, A. Arnout, and F. C. R. Spieksma, "On a periodic vehicle routing problem," *J. Oper. Res. Soc.*, vol. 61, no. 12, pp. 1719-1728, Dec. 2010. doi: 10.1057/jors.2009.154

[11] I. M. Chao, B. L. Golden and E. Wasil, "An improved heuristic for the period vehicle routing problem," *Networks*, vol. 26, no. 1, pp. 25-44, Aug. 1995. doi: 10.1002/net.3230260104

[12] M. G. Baldoquin de la Peña, A. Escalera Fariñas, and R. Linfati, "A model and solution method for solving the real-world and complex problem of scheduling visits to customers," *J. Appl. Res. Technol.*, vol. 12, no. 3, pp. 333-342, Jun. 2014. doi: 10.1016/S1665-6423(14)71616-5

[13] A. Escalera Fariñas and M. G. Baldoquin de la Peña, "Sistema soporte a la decisión para el agrupamiento de clientes de brascuba sa," *Ing. Ind.*, vol. 34, no. 2, pp. 143-154, May-Ago. 2013.

[14] A. M. Campbell and J. H. Wilson, "Forty years of periodic vehicle routing," *Networks*, vol. 63, no. 1, pp. 2-15, Jan. 2014. doi: 10.1002/net.21527.

[15] N. Christofides and J. E. Beasley, "The period routing problem," *Networks*, vol. 14, no. 2, pp. 237-256, 1984. doi: 10.1002/net.3230140205.

[16] R. A. Russell and D. Gribbin, "A multiphase approach to the period routing problem," *Networks*, vol. 21, no. 7, pp. 747-765, Dec. 1991. doi: 10.1002/net.3230210704.

[17] J. F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105-119, Sep. 1997. doi: 10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G

[18] V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl, "A variable neighborhood search heuristic for periodic routing problems," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 791-802, Jun. 2009. doi: 10.1016/j.ejor.2007.08.048.

[19] O. Lum, C. Cerrone, B. Golden, and E. Wasil, "Partitioning a street network into compact, balanced, and visually appealing routes," *Networks*, Vol. 69, no. 3, pp. 290-303, May. 2017. doi: 10.1002/net.21730.

[20] J. Pacheco and O. Valencia, "Análisis de nuevos métodos de clasificación. Un ejemplo ilustrativo de su uso en la agrupación de los municipios de Castilla y León," *Est. Econ. Apl.*, vol. 23, no. 3, pp. 711-729, 2005.

[21] É. D. Taillard, "Heuristic methods for large centroid clustering problems," *J. Heuristics*, vol. 9, no. 1, pp. 51-73, Jan. 2003. doi: 10.1023/A:1021841728075

[22] P. Hansen and N. Mladenovic, "J-MEANS: a new local search heuristic for minimum sum of squares clustering," *Pattern Recogn.*, vol. 34, no. 2, pp. 405-413, Feb. 2001. doi: 10.1016/S0031-3203(99)00216-2

[23] H-H. Bock, "Origins and extensions of the k-means algorithm in cluster analysis," *JEHPS*, vol. 4, no. 2, pp. 48-49, Dec. 2008.