LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

# A VARIABLE NEIGHBORHOOD SEARCH-BASED ALGORITHM FOR THE ELECTRIC VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

**André Luyde da Silva Souza**
Instituto Federal de São Paulo
Rua Zélia de Lima Rosa, 100 - Recanto das Primaveras I, Boituva - SP
andre.luyde@ifsp.edu.br

**Puca Huachi Vaz Penna, Marcone Jamilson Freitas Souza**
Universidade Federal de Ouro Preto
Departamento de Computação, ICEB, Campus Universitário, Morro do Cruzeiro - Ouro Preto/MG
{puca, marcone}@ufop.edu.br

**RESUMO**

Este trabalho apresenta um algoritmo flexível baseado em busca em vizinhança variável (VNS), nomeado Flexi-VNS, para o problema de roteamento de veículos elétricos com janelas de tempo (E-VRPTW). Este problema é uma variante do Problema de Roteamento de Veículos Elétricos clássico, acrescido da restrição de janela de tempo. Flexi-VNS inclui o *Randomized Variable Neighborhood Descent* (RVND) como procedimento de busca local. O RVND incorpora um novo procedimento intra-RVND, que é chamado toda vez que uma solução é modificada, sendo aplicado somente às rotas alteradas. Avaliamos o desempenho do Flexi-VNS em instâncias de referência do E-VRPTW e o comparamos com os melhores algoritmos da literatura. Os resultados mostram que o algoritmo Flexi-VNS encontra todas as melhores soluções da literatura em instâncias de menor porte. Dentre as 56 instâncias de grande porte, ele melhorou 7 e igualou os resultados em 22.

**PALAVRAS CHAVE. Veículos Elétricos. Roteamento de Veículos. Janelas de Tempo.**

**Tópicos: CRIS (Cidades e Regiões Inteligentes e Sustentáveis). L&T – Logística e Transportes**

**ABSTRACT**

This paper presents a flexible algorithm based on variable neighborhood search (VNS), called Flexi-VNS, for the electric vehicle routing problem with time windows (E-VRPTW). This problem is a variant of the classic Electric Vehicle Routing Problem, with the addition of the time window constraint. Flexi-VNS includes Randomized Variable Neighborhood Descent (RVND) as a local search procedure. RVND incorporates a new intra-RVND procedure, which is called every time a solution is modified, being applied only to the changed routes. We evaluated the performance of Flexi-VNS on E-VRPTW reference instances and compared it with the best algorithms in the literature. The results showed that the Flexi-VNS algorithm found all the best solutions in the literature on small-scale instances. Among the 56 large-scale instances, it improved seven and equaled the results in 22.

**KEYWORDS. Electric Vehicle. Vehicle Routing. Time Windows.**

**Paper topics: CRIS (Smart and Sustainable Cities & Regions). LT - (Logistics and Transport).**

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

## 1. Introduction

Governments around the world have been formulating carbon reduction strategies for freight transportation. These strategies typically involve analyzing Greenhouse Gas (GHG)/CO2 emissions from transportation operations. Road transportation has tended to gain attention, as it is the predominant mode of freight transport and is responsible for the most significant part of pollutant emissions within nations [McKinnon e Piecyk, 2009]. According to Koç e Karaoglan [2016], the green logistics concept has emerged as an extension of the Vehicle Routing Problem (VRP). Green logistics involves the consideration of alternative fuels for vehicle fleets to reduce the emission of carbon dioxide (CO2). Several studies have proposed methods for the incorporation of these alternative recharge methods into VRPs.

Transportation represents a significant component of logistics. The environmental logistics policy requires that logistics change the transportation scheme and shift it into a distribution network with minimal environmental impact. This highlights several problems concerning green transportation, including alternative fuels, electric vehicles, green intelligent transportation systems, and other eco-friendly infrastructures. Green logistics deals with activities that reduce energy and measure the environmental effects of different strategies. Based on these activities, VRP variants are identified from an operations research perspective and classified as the Green Vehicle Routing Problem (G-VRP) [Lin et al., 2014]. G-VRP is a variant of the VRP that harmonizes the environmental and economic costs by implementing effective routes to meet the ecological concerns and the usual objectives of VRP [Rodrigue et al., 2017]. According to Zhang et al. [2017], the VRP focuses on finding the optimal route to minimize transportation costs, and the G-VRP emphasizes economic costs and sustainability issues. The sustainable problems are described as decreasing GHG emissions by reducing the fuel consumption of vehicles and using alternative fuel-powered vehicles that depend on fuel sources such as electricity, natural gas, and hydrogen, among others.

In this way, the EVRP can be classified as a variant of G-VRP. Electric vehicles can decrease transportation costs and pollution effects compared to other vehicles used in the transportation system. However, the limited battery capacity, reduced number of charging or swapping stations, long recharging battery time, and limited range driving make the charging process challenging compared with fossil fuel vehicles [Kucukoglu et al., 2021]. Schneider et al. [2014] defined the EVRP as an extension of the classic VRP. The EVRP considers the vehicle fleet to be composed exclusively of electric cars. The differences between the problems are related to the battery constraints and charging operations.

This paper addresses the E-VRPTW, this problem is a variant of EVRP that has been extensively researched in the literature. Schneider et al. [2014] introduced the problem by incorporating the recharging plan and the time window restriction on the EVRP. The authors considered the time window constraints essential to get the problem closer to the real-world logistics application.

To solve the E-VRPTW, we used an algorithm based on the General Variable Neighborhood Search (GVNS) metaheuristic, called Flexi-VNS. Its local search is the main difference between our proposed algorithm and the classical GVNS. Our local search is done through the Randomized Variable Neighborhood Descent (RVND) procedure, a Variable Neighborhood Descent (VND) variant [Mladenović e Hansen, 1997]. Our algorithm is adapted to solve the problem, and it is detailed in Section 4.2. To test it, we submitted our algorithm on instances used by Schneider et al. [2014]. The results were compared with the best-known solutions reported in the literature.

The remaining of this work is organized as follows. Section 2 presents a brief review of studies related to the E-VRPTW. Section 3 describes the mathematical formulation. Section 4 details the Flexi-VNS components. Section 5 shows the computational experiments. Finally, Section 6 concludes this work.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

## 2. Literature review on E-VRPTW

The E-VRPTW is one variant extensively researched in the literature. Several authors studied it and the extensions provided by the problem. Following, we present some studies regarding the problem and its variants.

Schneider et al. [2014] introduced an E-VRPTW, which incorporates the possibility of recharging at any of the available stations using an appropriate recharging scheme. Vehicle fleets with limited capacity and time windows are essential constraints in real-world logistics applications. The authors considered these constraints to their problem. They proposed a hybrid heuristic combining the Tabu Search (TS) heuristic with the VNS heuristic. To test their approach, they designed a benchmark based on Vehicle Routing Problem with Time Windows (VRPTW) instances proposed by Solomon [1987]. To demonstrate the performance of their metaheuristic, they made tests on instances of the related problems Multi-depot Vehicle Routing Problem with Inter-depot Routes (MDVRPI), G-VRP, and VRPTW. Their method assisted in routing and recharging decisions for Electric Commercial Vehicles (ECVs) employed in real-world delivery operations.

Keskin e Çatay [2016] studied the variant of the E-VRPTW; the problem was the Electric Vehicle Routing Problem with Time Windows and Partial Recharge (E-VRPTW-PR). This difference consists in relaxing the full recharge restriction and allowing partial recharging. This way, the problem is more practical in the real world due to the shorter recharging duration. To solve the problem, they developed an Adaptive Large Neighborhood Search (ALNS) algorithm. They used benchmark instances from the literature to test their approach. Computational results show that their proposed method is effective in finding high-quality.

Hiermann et al. [2016] introduced the Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations (E-FSMFTW). The problem consists of finding a tour for different vehicle types to serve customers and minimize transportation costs and the distance traveled. They proposed a formulation based on Bräysy et al. [2008] for the Fleet Size and Mix Vehicle Routing Problem with Time Windows (FSMVRPTW) and Schneider et al. [2014] for the E-VRPTW. They proposed a Branch-and-price solver and a hybrid heuristic, which combines an ALNS extended with an embedded intensification mechanism to solve the problem. The authors proposed a benchmark based on the instances of Schneider et al. [2014] to test their approach. The Branch-and-price obtained optimal solutions for smaller instances and lower bounds for larger instances. For the small instances, the ALNS found all optimal solutions. Their heuristic solver found solutions within a gap of around one percent to the best-known solution for larger instances. Several of these best-known solutions have been proven optimal by the branch-and-price solver.

Recently, Keskin et al. [2021] worked on an extension named Electric Vehicle Routing Problem with Time Windows and Stochastic Waiting Times at Recharging Stations (E-VRPTW-STRS). Their approach differs from the others by the waiting time at the recharging station. This time is stochastic; thus, the range time is previously known, but not the exact time. This time is revealed when the vehicle arrives at the recharge station. They proposed a simulation-based ALNS heuristic to solve the problem. To test the performance of their approach, they solved the version deterministic of the problem, comparing the results and computing the expected recourse cost. They used a discrete event simulation to calculate and evaluate the objective function. The results show efficiency in finding solutions within a reasonable computer time frame.

More details about E-VRP, E-VRPTW, and related variants are presented in the following reviews: Erdelic et al. [2019], Martins et al. [2021], Qin et al. [2021], Kucukoglu et al. [2021].

## 3. Electric Vehicle Routing Problem with Time Windows

The E-VRPTW is a variant of the EVRP with time windows as an additional constraint to the problem. Another difference in this variant is that the charging stations have their location previously defined. Therefore, there is no cost to build the charging stations [Schneider et al., 2014].

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

The problem is defined as follows. Let $V'$ be a set of vertices $V' = \{V \cup F\}$, where $V = \{1, \cdots, n\}$ represents the set of $n$ customers and $F$ is the set of charging stations $F = \{1, \cdots, |F|\}$. Vertices 0 and $n+1$ represent the depot, and each route must begin and end at the same depot. Moreover, we define $F_0 = F \cup \{0\}, V'_0 = V' \cup \{0\}$, $V'_{n+1} = V' \cup \{n+1\}$, and $V'_{0,n+1} = V' \cup \{0, n+1\}$. The EVRP can be represented on a fully connected undirected graph $G = \left(V'_{0,n+1}, E\right)$, where the set of edges $E = \{(i,j)\,|i,j \in V'_{0,n+1}, i \neq j\}$. For each edge $(i,j) \in E$, there is a shipping cost $d_{ij}$, travel time $\delta_{ij}$, and a constant battery consumption rate $h$ (per unit distance), i.e., traversing this edge consumes $h \times d_{ij}$ battery charge. The fleet of homogeneous EV is represented by $K$, and for each EV $k \in K$, a demand capacity $Q$, a duration time $L$, and a battery capacity $B$ have to be respected. The battery of EVs is fully charged when they leave the depot. Each vertex $i \in V'_{0,n+1}$ has a positive demand $q_i$, which is 0 if $i \notin V$, a service time $s_i$ $(s_0 = s_{n+1} = 0)$, and a time window $[e_i, l_i]$ that must be respected. The service time $s_i$ cannot begin before $e_i$, which might cause a waiting time. It is not authorized to start after $l_i$, but can finish later. At a charging station, the recharging time depends on the energy level of the EV when it arrives at the station. The difference between the present battery level and the battery capacity $B$ is recharged with a recharging rate of $g$ (i.e., assuming the full recharging policy). Each customer must be visited once by precisely one vehicle, i.e., split delivery is not permitted. A solution to the problem is called route plan and consists of $|R|$ $(|R| \leq |K|)$ feasible routes, where $R = \{r_1, r_2, \cdots, r_k, \cdots, r_{|R|}\}$. Thus, the number of routes equals the number of EVs used in the solution, which can be less or equal to the number of available EVs. A feasible route $r_k$ is executed by EV $k$ that departures from the depot, attends a subset of $V$, and then returns to the depot. The route total demand cannot surpass $Q$. We define the decision variables $\tau_{ik}$ as arrival time, $u_{ik}$ as remaining load, and $y_{ik}$ as the remaining battery level on arrival at vertex $i \in V'_{0,n+1}$ for the vehicle $k \in K$. Let $x_{ijk}$ $\left(i \in V'_0, j \in V'_{n+1}, i \neq j, k \in K\right)$ be a binary decision variable equals 1 if an edge is traveled by a vehicle $k$ and 0 otherwise. The objective function of the problem is to minimize the total traveling distance. The mixed-integer linear programming formulation for the E-VRPTW based on the EVRP model presented on Schneider et al. [2014] is described in supplementary material, available here.

## 4. Solution Approach

Since the mathematical model of the problems described in the previous chapter cannot solve big-size instances of the E-VRPTW, we developed a heuristic algorithm to solve these instances. The algorithm used to solve the problem is named Flexi-VNS, it is a VNS-based algorithm adapted from the classic VNS, first introduced by Mladenović e Hansen [1997] and improved in other studies like [Hansen e Mladenović, 2001; Hansen et al., 2008, 2010, 2019].

### 4.1. Solution representation

We represent the E-VRPTW solution by a matrix of $|R|$ rows, where each row informs the route assigned to a vehicle (the $r$-th row refers to the route of the $r$-th vehicle). Each route begins and ends at the depot. The customers are inserted in the route, satisfying the problem constraints.

Figure 1 shows a solution representation for an instance with five vehicles and 20 customers. The numbers with $(CS)$ represent the charging station. The depot is represented by node 0.

So, the solution $S$ of Figure 1 is $S = R_1 \cup R_2 \cup \ldots R_5$, where $R_4 = \{0, 10, 12, 23, 13, 0\}$, for example.

### 4.2. Variable Neighborhood Search

The VNS algorithm was proposed by Mladenović e Hansen [1997]. The VNS is a metaheuristic that explores neighborhoods of the solution and moves to a neighbor if it improves the

DOI: 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $R_1$ | 0 | 9 | 4 | 22(CS) | 2 | 15 | 17 | 6 | 19 | 0 |
| $R_2$ | 0 | 3 | 5 | 24(CS) | 1 | 14 | 18 | 0 | | |
| $R_3$ | 0 | 7 | 8 | 20 | 21(CS) | 0 | | | | |
| $R_4$ | 0 | 10 | 12 | 23(CS) | 13 | 0 | | | | |
| $R_5$ | 0 | 11 | 16 | 0 | | | | | | |

Figure 1: Example of a solution $S$.

current solution. These moves are executed during a limited number of times. VNS-based algorithms have three main steps that have to be specified: the initial solution, the shaking procedure, and the local search procedure.

In the initial solution, we adopt a random approach to increase the diversification capability of the proposed Flexi-VNS method. Our initial solution works as follows: The algorithm first selects a vehicle and then allocates a randomly selected non-assigned customer to this vehicle respecting its capacity constraint. Suppose the sum of customers' demand for the vehicle plus the demand of the selected customer is lower than the vehicle's capacity. In that case, the customer is added to the vehicle route and removed from the set of unassigned customers. Otherwise, a new random non-assigned customer is selected. When no customer can be added to the selected vehicle route, the vehicle returns to the depot. After that, a new vehicle is selected, and a route for this vehicle will be built similarly. These steps are repeated until all the customers are assigned. If all vehicles are used and unallocated customers remain, a dummy vehicle ($EV_{k+1}$) is created, and all unallocated customers are added to this vehicle. This vehicle is penalized at the objective function.

The shaking procedure is used to avoid getting stuck in a local optimum and diversify the search. This procedure has three parameters: a solution $S$, the set $NS$ of shaking operators, and the maximum number of iterations *iter*. The shaking operators have the same probability of being selected. In each iteration, the procedure chooses a shaking operator from the set $NS$, then applies the move to $S'$ and stores it in $S''$. The obtained solution is set to be the new incumbent solution $S$. This procedure is repeated until it meets a pre-specified number of consecutive modifications in the solution $S'$, given by *levelMax*. To avoid generating completely random solutions, the value of this parameter was calibrated and limited to $\textit{levelMax} = 4$.

In order to obtain a local optimum solution, the local search procedure is applied to the solution obtained in the shaking phase. We perform a local search by applying the RVND method presented in Souza et al. [2010] using the set $NL$ of local search operators. The RVND is a VND variant, first introduced in Mladenović e Hansen [1997]. Unlike in the classic VND, in the RVND, the exploration order of the neighborhoods is not defined in advance. The random exploration order of the neighborhoods avoids looking for the best order of neighborhoods, which is dependent on the problem instance [Subramanian et al., 2010].

We follow the idea presented in Penna et al. [2019] and apply a new intensification procedure to the set of improved routes in the incumbent solution $S$. We named this procedure intra-RVND. Firstly we create the subset $S_i$, which is a partial solution containing the improved routes of the solution $S$. The remained routes are allocated to a subset $S^-$. Then, a random neighborhood from $NI$ is selected and then applied to the partial solution $S_i$. If the solution returned $S_i'$ improves the partial solution $S_i$, then $S_i$ is updated, and $NI$ is restarted with all $|NI|$ neighborhoods. Otherwise, if $S_i'$ does not improve $S_i$ the neighborhood $NI_l$ is removed from $NI$. Finally, after applying all $|NI|$ neighborhoods without improvement in the partial solution $S_i$, the method returns the complete solution $S$ composed of the routes in the subset $S^-$ and the partial solution $S_i$.

The proposed VNS-based algorithm, named Flexi-VNS, is presented in Algorithm 1.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

Firstly, Flexi-VNS receives the maximum number of iterations (*iterMax*) as a parameter. Next, it generates the initial solution $S_0$ (line 1). This step is followed by the creation of three sets of neighborhood structures, one for the shaking operators $NS = \{NS_1, ..., NS_a\}$ (line 3), one for the local search operators $NL = \{NL_1, ..., NL_b\}$ (line 4), and one for the intra-route neighborhood used by the intra-RVND procedure located inside the local search procedure $NI = \{NI_1, ..., NI_c\}$ (line 5). These structures are described in Section 4.3. In the shaking procedure (line 8), solution $S'$ is obtained by applying the neighborhoods $NS$ on the current solution $S$. The number of modifications is determined by *iter*. Thus, the RVND method is applied in the solution $S'$ using the neighborhood structures $NL$ to obtain a new solution $S''$. If $S''$ is feasible and its objective value is better than the incumbent solution $S$, $S$ is replaced by $S''$ (line 11). When there are *iterMax* consecutive iterations without improvement, the algorithm ends and returns the best solution found in line 17.

**Algorithm 1:** Flexi-VNS ($S$, *iterMax*)

1 $S_0 \leftarrow initialSolution()$
2 $S \leftarrow S_0$
3 $NS \leftarrow \{NS_1, ..., NS_a\}$
4 $NL \leftarrow \{NL_1, ..., NL_b\}$
5 $NI \leftarrow \{NI_1, ..., NI_c\}$
6 *iter* $\leftarrow 1$
7 **while** *iter* $\leq$ *iterMax* **do**
8 $S' \leftarrow$ Shaking$(S, NS,$ *iter*$)$
9 $S'' \leftarrow$ RVND$(S', NL, NI)$
10 **if** $f(S'') < f(S)$ **then**
11 $S \leftarrow S''$
12 *iter* $\leftarrow 1$
13 **else**
14 *iter* $\leftarrow$ *iter* $+ 1$
15 **end**
16 **end**
17 **return** $S$

### 4.3. Neighborhood structures

To explore the search space of the solution problem, five inter-route neighborhoods and five intra-route neighborhoods were implemented.

#### 4.3.1. Inter-route Neighborhoods

These neighborhoods involve the relocation or exchange of customers/battery stations between two different vehicles. We use five classical VRP neighborhoods, i.e.: (i) Shift(1,0) – one customer is removed from one vehicle and inserted into another; (ii) Shift(2,0) – two consecutive customers are removed from one vehicle and inserted in another one; (iii) Swap(1,1) – one customer selected in one route is exchanged by another customer selected in a second route; (iv) Swap(2,1): Two subsequent customers selected in one vehicle are exchanged by one customer in another vehicle; (v) Swap(2,2): Two subsequent customers are selected in one vehicle and two subsequent in the other vehicle and exchanged.

#### 4.3.2. Intra-route Neighborhoods

These neighborhoods consist of changing customers/battery stations in the routes. They are described below and applied in a unique vehicle at a time. We use five different moves, i.e.: (i) AddStation: one battery station is selected and inserted in a determined position in the route;

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

(ii) RemoveStation: one battery station is removed from the vehicle's route, and the adjacent customers of the battery station are connected; (iii) Relocate: one customer or battery station is selected and relocated into another position in the vehicle's route; (iv) Relocate2: Two subsequent customers/battery stations are selected and relocated into another position in the vehicle's route; (v) Swap: Two customers/battery stations are selected and exchanged the positions in the vehicle's route.

### 4.4. Solution evaluation

According to Vidal et al. [2013], using infeasible solutions during the search for routing problems contributes significantly to the performance of heuristics. Furthermore, solution penalization is used to explore the search space on moves efficiently. Thus, we penalized battery, load, and time window violations for the E-VRPTW. So, solutions that do not respect these constraints are allowed and handled through a dynamic penalty mechanism.

This way, the evaluation function of a solution $S$ is determined according to Eq. (1):

$$f(S) = z + \alpha_1 \times dm(S) + \alpha_2 \times EY(S) + \alpha_3 \times tw(S) + \alpha_4 \times vh(S), \tag{1}$$

where $z$ represents the objective function of the problem described in the supplementary material. In turn, $\alpha_1$, $\alpha_2$, $\alpha_3$, and $\alpha_4$ are the penalization values for a vehicle load, battery capacity, time window violation, the route, and an extra vehicle used in the initial solution, respectively.

The values of $EY(S)$, $dm(S)$, $tw(S)$, and $vh(S)$ are detailed below:

- $EY(S)$: Two variables are needed to calculate the battery violations. The first $Y^{\rightarrow}_{vi}$ represents the cumulative battery charge necessary to reach $v_i$ from the last recharging station visited or the depot. The second is $Y^{\leftarrow}_{vi}$, which denotes the amount of charge to reach the next charge station or depot beginning from $v_i$. Equations (2) and (3) describe them.

$$Y^{\rightarrow}_{v_i} = \left\{ \begin{array}{ll} \text{if } v_{i-1} \in F'_0 & = h \times d_{v_{i-1}v_i} \\ Otherwise & = Y^{\rightarrow}_{v_{i-1}} + r \times d_{v_{i-1}v_i} \end{array} \right\} \tag{2}$$

$$Y^{\leftarrow}_{v_i} = \left\{ \begin{array}{ll} \text{if } v_{i+1} \in F'_{n+1} & = h \times d_{v_i v_{i+1}} \\ Otherwise & = Y^{\leftarrow}_{v_{i+1}} + h \times d_{v_i v_{i+1}} \end{array} \right\} \tag{3}$$

  The sum of extra energy needed to satisfy the limit of battery constraint of the solution is calculated by Equation (4).

$$EY(S) = \sum_{k \in K} \max\{0, \sum_{i \in R_k} Y^{\rightarrow}_{v_i} - B\} \tag{4}$$

- $dm(S)$: The sum of the load capacity violation in the solution $S$, calculated by Equation (5).

$$dm(S) = \sum_{k \in K} \max\{0, \sum_{i \in R_k} q_i - Q\} \tag{5}$$

- $tw(S)$: It represents the sum of the time windows violation in the solution $S$. The function 6 shows the $tw(S)$ calculation.

$$tw(s) = \sum_{i \in V'_{0,n+1}} (\tau_i - l_i)^+ \tag{6}$$

- $vh(S)$: The vehicle value ($EV_{k+1}$) used in the initial solution, calculated by Equation (7):

$$vh(S) = \max\{0, \bar{v} - |K|\}, \tag{7}$$

  where $\bar{v}$ is the total number of vehicles used in the solution $S$.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

## 5. Computational Experiments

This section presents the results and analyses of the proposed solution approach on benchmark instances of the three problems. These benchmarks are described in Subsection 5.1. All experiments were implemented in C++ and performed on an Intel Core i7-6700HQ CPU @ 2.60 GHz $\times$ 8 with 8 GB RAM. The parameter calibration is available in Subsection 5.2. Subsection 5.3 reports the results obtained executing our proposed algorithm in the literature instances. The code and the solutions are available in `https://github.com/AndreLuyde/EVRPTW-SBPO`.

### 5.1. Data Set

The E-VRPTW data sets used were generated by Schneider et al. [2014]. They used the Solomon [1987] instances for VRPTW as a base to create these instances. They made a group of 56 large instances with 100 customers each, 23 recharging stations, and another set of 36 small instances with 5, 10, and 15 customers. The instances were divided into three classes according to the geographic distribution of the customer's locations. The classes are R, C, and RC. Class R represents random customer distribution, class C denotes clustered customer distribution, and class RC merges the other two classes. They are also subdivided into groups. Groups R1, C1, and RC1 have a short scheduling horizon, which means they usually need more vehicles, and R2, C2, and RC2 have a long scheduling horizon.

All the instances have a recharging station at the depot. The other stations are randomly distributed, limited to generate feasible and meaningful instances, i.e., every customer can be reached from the depot using at most two different recharging stations. Two values are considered to define the battery capacity. The first is the amount of charge needed to travel, which is 60% of the average route length of the best-known solution of the correspondent VRPTW instance. The second is twice the necessary battery charge to cross the longest arc between a customer and a recharging station. The battery capacity is defined as the bigger of these values. This procedure ensures the feasibility of the instance and assurances that recharging stations have to be used. The consumption rate $h$ is defined as 1.0 and the recharging rate $g$ is defined to ensure that a full charge requires three times the average customer service time of the instance.

### 5.2. Parameter Calibration

The performance of a metaheuristic is highly affected by the values of relevant parameters. Thus, it is important to calibrate the best values for the algorithm parameters so that the algorithm can reach its best performance. We determine suitable values for the parameters of the proposed solution approach by using the calibration software *irace* presented in López-Ibáñez et al. [2016]. This tool has been widely adopted in EVRP studies, such as in Bruglieri et al. [2022], Pan et al. [2021], and Peng et al. [2019].

Table 1 shows the Flexi-VNS parameters and the *irace* results for the E-VRPTW. In this table, the first column represents the parameter's name, followed by their description in the second column. The third column presents the range of possible values for each parameter. Finally, the last column highlights the returned values by *irace*.

Table 1: Flexi-VNS parameter values provided for the *irace* and the best ones returned by it.

| Parameter | Description | Range of Values | Returned Value |
|---|---|---|---|
| $\alpha_1$ | Unit penalty for load capacity violation | [10, 1000] | 849 |
| $\alpha_2$ | Unit penalty for battery capacity violation | [10, 1000] | 919 |
| $\alpha_4$ | Unit penalty for time window violation | [10, 1000] | 692 |
| lvlShake | Maximum level for shake procedure | [1, 7] | 4 |

The parameter $\alpha_4$ was not submitted to the *irace* tool because it represents an extra vehicle. Thus, it was set to a high value (10000) to ensure the algorithm could remove this vehicle.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

### 5.3. Results on E-VRPTW instances

The VNS proposed algorithm showed excellent performance, finding all BKS solutions available in the literature execution in the small-size instances of Schneider et al. [2014]. The detailed results on those instances are described in supplementary material, available here.

Schneider et al. [2014] also proposed instances with 100 customers in the same three classes, C, R, and RC. Tables 2, 3, and 4 show the results of our algorithm's execution in these bigger instances. Column **Instances** shows the name of the instance tested. Columns **BKS** and **nv** present the best-known solution reported in the literature and the number of vehicles used in the BKS. Columns **sol**, **nv**, $gap$, and **time** show the best solution found, the number of vehicles used in the solution, the gap from the BKS, and the CPU time spent expressed in seconds referred at the columns' top, respectively. Columns $\overline{\textbf{sol}}$ and $\overline{gap}$ show the average solution calculated with ten executions and the gap of the $\overline{\textbf{sol}}$ to the BKS.

Table 2: Class C results for larger instances proposed by Schneider et al. [2014].

| Instances | BKS | nv | VNS/TS | | | ALNS$_1$ | | | | HGA | | | ALNS$_3$ | | | Flexi-VNS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Class C | | | sol | nv | $gap$ | sol | nv | time | $gap$ | sol | time | $gap$ | sol | nv | $gap$ | sol | nv | time | $gap$ | $\overline{sol}$ | $\overline{gap}$ |
| C101 | **1053.83** | 12 | **1053.83** | 12 | 0.00 | **1053.83** | 12 | 97.20 | 0.00 | **1053.83** | 228.60 | 0.00 | **1053.83** | 12 | 0.00 | **1053.83** | 12 | 83.20 | 0.00 | **1053.83** | 0.00 |
| C102 | **1051.38** | 11 | 1057.16 | 11 | 0.55 | **1051.38** | 11 | 156.60 | 0.00 | 1055.12 | 322.20 | 0.35 | 1056.12 | 11 | 0.45 | **1051.38** | 11 | 295.50 | 0.00 | 1064.49 | 1.23 |
| C103 | **1034.86** | 10 | 1041.55 | 10 | 0.64 | **1034.86** | 10 | 216.60 | 0.00 | 1034.86 | 595.80 | 0.00 | 1001.81 | 11 | -3.30 | **1034.86** | 10 | 356.95 | 0.00 | 1042.36 | 0.72 |
| C104 | **951.57** | 10 | 980.82 | 10 | 2.98 | 961.88 | 10 | 293.40 | 1.07 | 953.63 | 604.20 | 0.22 | **951.57** | 10 | 0.00 | **951.57** | 10 | 296.04 | 0.00 | 972.70 | 2.17 |
| C105 | **1075.37** | 11 | **1075.37** | 11 | 0.00 | **1075.37** | 11 | 109.20 | 0.00 | **1075.37** | 279.00 | 0.00 | **1075.37** | 11 | 0.00 | **1075.37** | 11 | 448.87 | 0.00 | 1085.03 | 0.89 |
| C106 | **1057.65** | 11 | 1057.87 | 11 | 0.02 | **1057.65** | 11 | 146.40 | 0.00 | **1057.65** | 315.00 | 0.00 | **1057.65** | 11 | 0.00 | **1057.65** | 11 | 238.58 | 0.00 | 1058.16 | 0.05 |
| C107 | **1031.56** | 11 | **1031.56** | 11 | 0.00 | **1031.56** | 11 | 163.20 | 0.00 | **1031.56** | 420.00 | 0.00 | **1031.56** | 11 | 0.00 | **1031.56** | 11 | 199.82 | 0.00 | 1045.70 | 1.35 |
| C108 | **1095.66** | 10 | 1100.32 | 10 | 0.42 | **1095.66** | 10 | 192.00 | 0.00 | 1015.68 | 508.20 | -7.87 | 1015.68 | 11 | -7.87 | 1015.68 | 11 | 187.67 | -7.87 | 1015.92 | -7.85 |
| C109 | **1033.67** | 10 | 1051.84 | 10 | 1.73 | **1033.67** | 10 | 170.40 | 0.00 | 1060.78 | 675.60 | 2.56 | 1069.16 | 10 | 3.32 | 1059.61 | 10 | 1115.36 | 2.45 | 1035.74 | 0.20 |
| C201 | **645.16** | 4 | **645.16** | 4 | 0.00 | **645.16** | 4 | 87.00 | 0.00 | **645.16** | 322.80 | 0.00 | **645.16** | 4 | 0.00 | **645.16** | 4 | 181.13 | 0.00 | **645.16** | 0.00 |
| C202 | **645.16** | 4 | **645.16** | 4 | 0.00 | **645.16** | 4 | 149.40 | 0.00 | **645.16** | 455.40 | 0.00 | **645.16** | 4 | 0.00 | **645.16** | 4 | 225.61 | 0.00 | 645.57 | 0.06 |
| C203 | **644.98** | 4 | **644.98** | 4 | 0.00 | **644.98** | 4 | 208.80 | 0.00 | **644.98** | 475.80 | 0.00 | **644.98** | 4 | 0.00 | **644.98** | 4 | 371.34 | 0.00 | 645.06 | 0.01 |
| C204 | **636.43** | 4 | **636.43** | 4 | 0.00 | **636.43** | 4 | 207.00 | 0.00 | **636.43** | 514.20 | 0.00 | **636.43** | 4 | 0.00 | **636.43** | 4 | 323.20 | 0.00 | 639.13 | 0.42 |
| C205 | **641.13** | 4 | **641.13** | 4 | 0.00 | **641.13** | 4 | 121.80 | 0.00 | **641.13** | 387.60 | 0.00 | **641.13** | 4 | 0.00 | **641.13** | 4 | 142.47 | 0.00 | **641.13** | 0.00 |
| C206 | **638.17** | 4 | **638.17** | 4 | 0.00 | **638.17** | 4 | 139.80 | 0.00 | **638.17** | 439.20 | 0.00 | **638.17** | 4 | 0.00 | **638.17** | 4 | 166.47 | 0.00 | **638.17** | 0.00 |
| C207 | **638.17** | 4 | **638.17** | 4 | 0.00 | **638.17** | 4 | 162.60 | 0.00 | **638.17** | 453.60 | 0.00 | **638.17** | 4 | 0.00 | **638.17** | 4 | 311.17 | 0.00 | **638.17** | 0.00 |
| C208 | **638.17** | 4 | **638.17** | 4 | 0.00 | **638.17** | 4 | 156.60 | 0.00 | **638.17** | 452.40 | 0.00 | **638.17** | 4 | 0.00 | **638.17** | 4 | 198.24 | 0.00 | **638.17** | 0.00 |
| Average | | | | | 0.37 | | | 163.41 | 0.06 | | 433.84$^a$ | 0.20$^a$ | | | 0.25$^a$ | | | 309.62$^a$ | 0.15$^a$ | 276.35$^a$ | 0.44$^a$ |

VNS/TS on Schneider et al. [2014], ALNS$_1$ on Goeke e Schneider [2015], HGA on Hiermann et al. [2019], and ALNS$_2$ on Keskin e Çatay [2016]
$^a$: The average computational times and gap values were calculated considering only the solutions with the same number of vehicles.

Table 3: Class R results for larger instances proposed by Schneider et al. [2014].

| Instances | BKS | nv | VNS/TS | | | ALNS$_1$ | | | | HGA | | | ALNS$_3$ | | | Flexi-VNS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Class C | | | sol | nv | $gap$ | sol | nv | time | $gap$ | sol | time | $gap$ | sol | nv | $gap$ | sol | nv | time | $gap$ | $\overline{sol}$ | $\overline{gap}$ |
| R101 | **1663.04** | 18 | 1672.55 | 18 | 0.57 | 1665.62 | 18 | 94.80 | 0.15 | **1663.04** | 464.40 | 0.00 | 1679.06 | 18 | 0.95 | **1663.04** | 18 | 599.70 | 0.00 | 1669.05 | 0.36 |
| R102 | **1484.57** | 16 | 1535.81 | 16 | 3.34 | 1487.41 | 16 | 135.60 | 0.19 | **1484.57** | 477.00 | 0.00 | 1524.14 | 16 | 2.60 | 1484.67 | 16 | 979.18 | 0.01 | 1484.88 | 0.02 |
| R103 | **1267.35** | 13 | 1299.64 | 13 | 2.48 | 1271.35 | 13 | 157.20 | 0.31 | 1268.88 | 713.40 | 0.12 | 1312.50 | 13 | 3.44 | 1268.29 | 13 | 769.60 | 0.07 | 1283.54 | 1.26 |
| R104 | **1088.43** | 11 | **1088.43** | 11 | 0.00 | **1088.43** | 11 | 145.80 | 0.00 | 1103.50 | 884.40 | 1.37 | 1071.89 | 12 | -1.54 | 1088.78 | 11 | 296.36 | 0.03 | 1092.78 | 0.40 |
| R105 | **1442.35** | 14 | 1473.59 | 14 | 2.12 | **1442.35** | 14 | 202.80 | 0.00 | 1382.70 | 813.60 | -4.31 | 1383.29 | 15 | -4.27 | 1447.88 | 14 | 714.50 | 0.38 | 1490.76 | 3.25 |
| R106 | **1321.08** | 13 | 1344.66 | 13 | 1.75 | 1324.10 | 13 | 158.40 | 0.23 | 1275.86 | 897.00 | -3.54 | 1276.15 | 14 | -3.52 | 1347.56 | 13 | 626.94 | 1.97 | 1369.73 | 3.55 |
| R107 | **1148.38** | 12 | 1154.52 | 12 | 0.53 | 1150.95 | 12 | 153.60 | 0.22 | 1148.38 | 826.80 | 0.00 | 1148.43 | 12 | 0.00 | **1147.71** | 12 | 518.48 | -0.06 | 1170.85 | 1.91 |
| R108 | **1049.12** | 11 | 1065.89 | 11 | 1.57 | 1050.04 | 11 | 157.80 | 0.09 | 1049.12 | 913.80 | 0.00 | 1051.59 | 11 | 0.23 | **1048.42** | 11 | 541.20 | -0.07 | 1067.27 | 1.70 |
| R109 | **1261.31** | 12 | 1294.05 | 12 | 2.53 | **1261.31** | 12 | 225.60 | 0.00 | 1214.63 | 865.80 | -3.84 | 1214.72 | 13 | -3.84 | 1302.89 | 12 | 441.51 | 3.19 | 1325.68 | 4.86 |
| R110 | **1119.50** | 11 | 1143.52 | 11 | 2.10 | **1119.50** | 11 | 221.40 | 0.00 | 1097.66 | 913.20 | -1.99 | 1097.89 | 12 | -1.97 | 1122.28 | 11 | 953.31 | 0.25 | 1187.87 | 5.76 |
| R111 | **1099.53** | 12 | 1124.06 | 12 | 2.18 | 1106.19 | 12 | 162.60 | 0.60 | **1099.53** | 883.20 | 0.00 | 1109.14 | 12 | 0.87 | 1108.00 | 12 | 504.46 | 0.76 | 1137.32 | 3.32 |
| R112 | **1016.63** | 11 | 1026.52 | 11 | 0.96 | **1016.63** | 11 | 163.80 | 0.00 | **1016.63** | 855.00 | 0.00 | 1038.74 | 11 | 2.13 | **1016.63** | 11 | 839.27 | 0.00 | 1049.82 | 3.16 |
| R201 | **1264.82** | 3 | **1264.82** | 3 | 0.00 | **1264.82** | 3 | 123.00 | 0.00 | 1267.14 | 588.60 | 0.18 | 1265.67 | 3 | 0.07 | 1271.12 | 3 | 676.44 | 0.50 | 1269.51 | 0.37 |
| R202 | **1052.32** | 3 | **1052.32** | 3 | 0.00 | **1052.32** | 3 | 162.60 | 0.00 | **1052.32** | 657.60 | 0.00 | **1052.32** | 3 | 0.00 | 1053.95 | 3 | 309.93 | 0.15 | 1081.45 | 2.69 |
| R203 | **895.54** | 3 | 912.86 | 3 | 1.90 | **895.54** | 3 | 188.40 | 0.00 | **895.54** | 741.00 | 0.00 | **895.54** | 3 | 0.00 | 901.42 | 3 | 488.48 | 0.65 | 941.08 | 4.84 |
| R204 | **779.49** | 2 | 790.57 | 2 | 1.40 | **779.49** | 2 | 273.00 | 0.00 | 784.77 | 1151.40 | 0.67 | 780.98 | 2 | 0.19 | 803.13 | 2 | 990.68 | 2.94 | 763.17 | -2.14 |
| R205 | **987.36** | 3 | 988.67 | 3 | 0.13 | **987.36** | 3 | 124.80 | 0.00 | **987.36** | 690.60 | 0.00 | **987.36** | 3 | 0.00 | **987.36** | 3 | 336.36 | 0.00 | 1019.60 | 3.16 |
| R206 | **922.19** | 3 | 925.20 | 3 | 0.33 | **922.19** | 3 | 156.00 | 0.00 | 925.34 | 850.80 | 0.34 | 922.70 | 3 | 0.06 | 922.70 | 3 | 1255.55 | 0.06 | 966.85 | 4.62 |
| R207 | **845.26** | 2 | 852.73 | 2 | 0.88 | **845.26** | 2 | 207.60 | 0.00 | 847.59 | 1084.80 | 0.27 | 847.14 | 2 | 0.22 | 811.48 | 3 | 866.82 | -4.16 | 841.74 | -0.42 |
| R208 | **736.12** | 2 | 736.60 | 2 | 0.07 | **736.12** | 2 | 211.20 | 0.00 | **736.12** | 1035.60 | 0.00 | **736.12** | 2 | 0.00 | 740.76 | 2 | 821.66 | 0.63 | 769.11 | 4.29 |
| R209 | **867.05** | 3 | 872.36 | 3 | 0.61 | **867.05** | 3 | 159.60 | 0.00 | 870.68 | 1009.20 | 0.42 | 871.22 | 3 | 0.48 | 872.11 | 3 | 599.90 | 0.58 | 889.24 | 2.50 |
| R210 | **843.65** | 3 | 847.06 | 3 | 0.40 | 846.20 | 3 | 186.60 | 0.30 | 846.62 | 1071.60 | 0.35 | **843.65** | 3 | 0.00 | 846.21 | 3 | 453.66 | 0.30 | 876.29 | 3.72 |
| R211 | **827.89** | 2 | 866.21 | 2 | 4.42 | **827.89** | 2 | 211.80 | 0.00 | 836.27 | 1167.00 | 1.00 | 761.56 | 3 | -8.71 | 874.32 | 2 | 1155.28 | 5.31 | 819.49 | -1.02 |
| Average | | | | | 1.31 | | | 173.22 | 0.09 | | 845.59$^a$ | 0.24$^a$ | | | 0.13$^a$ | | | 676.02$^a$ | 0.80$^a$ | 562.44$^a$ | 2.38$^a$ |

VNS/TS on Schneider et al. [2014], ALNS$_1$ on Goeke e Schneider [2015], HGA on Hiermann et al. [2019], and ALNS$_2$ on Keskin e Çatay [2016].
$^a$: The average computational times and gap values were calculated considering only the solutions with the same number of vehicles.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

Table 4: Class RC results for larger instances proposed by Schneider et al. [2014].

| Instances | BKS | nv | VNS/TS | | | ALNS$_1$ | | | | HGA | | | ALNS$_3$ | | | Flexi-VNS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Class C | | | sol | nv | *gap* | sol | nv | time | *gap* | sol | time | *gap* | sol | nv | *gap* | sol | nv | time | *gap* | $\overline{sol}$ | $\overline{gap}$ |
| RC101 | **1723.79** | 16 | 1731.07 | 16 | 0.42 | 1726.91 | 16 | 87.00 | 0.18 | **1723.79** | 781.20 | 0.00 | 1731.07 | 16 | 0.42 | **1723.79** | 16 | 649.19 | 0.00 | 1773.68 | 2.81 |
| RC102 | **1659.53** | 14 | 1554.61 | 15 | -6.75 | 1552.08 | 15 | 112.80 | -6.92 | 1551.28 | 874.20 | -6.98 | 1551.69 | 15 | -6.95 | **1649.23** | 14 | 385.91 | -0.62 | 1626.54 | -2.03 |
| RC103 | **1350.09** | 13 | 1353.55 | 13 | 0.26 | **1350.09** | 13 | 147.60 | 0.00 | 1350.55 | 891.60 | 0.03 | 1351.73 | 13 | 0.12 | **1350.09** | 13 | 773.85 | 0.00 | 1366.82 | 1.22 |
| RC104 | **1227.25** | 11 | 1249.23 | 11 | 1.76 | 1227.25 | 11 | 171.60 | 0.00 | 1230.92 | 807.60 | 0.30 | 1232.45 | 11 | 0.42 | **1224.81** | 11 | 611.54 | -0.20 | 1245.23 | 1.44 |
| RC105 | **1473.24** | 14 | 1483.38 | 14 | 0.68 | 1475.31 | 14 | 88.80 | 0.14 | 1473.24 | 727.80 | 0.00 | 1473.24 | 14 | 0.00 | **1470.46** | 14 | 808.67 | -0.19 | 1498.13 | 1.66 |
| RC106 | **1423.27** | 13 | 1440.19 | 13 | 1.17 | 1427.21 | 13 | 99.60 | 0.28 | **1423.27** | 588.60 | 0.00 | 1414.99 | 14 | -0.59 | **1423.27** | 13 | 521.72 | 0.00 | 1438.90 | 1.09 |
| RC107 | **1274.41** | 12 | 1275.89 | 12 | 0.12 | 1274.89 | 12 | 123.60 | 0.04 | 1274.41 | 823.20 | 0.00 | 1283.05 | 12 | 0.67 | **1274.25** | 12 | 556.35 | -0.01 | 1303.72 | 2.21 |
| RC108 | **1197.41** | 11 | 1238.81 | 11 | 3.31 | 1197.83 | 11 | 156.60 | 0.00 | **1197.41** | 535.20 | 0.00 | 1209.11 | 11 | 0.93 | **1197.41** | 11 | 691.37 | 0.00 | 1238.62 | 3.29 |
| RC201 | **1444.94** | 4 | 1447.20 | 4 | 0.16 | **1444.94** | 4 | 106.20 | 0.00 | 1446.03 | 606.00 | 0.08 | 1446.84 | 4 | 0.13 | 1450.68 | 4 | 652.34 | 0.40 | 1471.60 | 1.81 |
| RC202 | **1410.74** | 3 | 1412.91 | 3 | 0.15 | **1410.74** | 3 | 194.40 | 0.00 | 1421.34 | 706.80 | 0.75 | 1450.34 | 3 | 2.73 | 1447.06 | 3 | 417.20 | 2.51 | 1315.98 | -7.20 |
| RC203 | **1055.19** | 3 | 1078.28 | 3 | 2.14 | **1055.19** | 3 | 232.80 | 0.00 | 1057.16 | 691.20 | 0.19 | 1069.27 | 3 | 1.32 | 1080.79 | 3 | 461.32 | 2.37 | 1131.21 | 6.72 |
| RC204 | **884.72** | 3 | 889.22 | 3 | 0.51 | 884.80 | 3 | 244.80 | 0.01 | 884.72 | 787.20 | 0.00 | 887.45 | 3 | 0.31 | **884.60** | 3 | 1436.32 | -0.01 | 929.22 | 4.79 |
| RC205 | **1273.55** | 3 | 1321.75 | 3 | 3.65 | **1273.55** | 3 | 191.40 | 0.00 | 1280.33 | 760.80 | 0.53 | 1277.60 | 3 | 0.32 | 1328.95 | 3 | 1319.53 | 4.17 | 1210.48 | -5.21 |
| RC206 | **1188.63** | 3 | 1191.13 | 3 | 0.21 | **1188.63** | 3 | 161.40 | 0.00 | 1190.50 | 666.60 | 0.16 | 1207.64 | 3 | 1.57 | 1228.37 | 3 | 811.97 | 3.24 | 1227.35 | 3.15 |
| RC207 | **985.03** | 3 | 995.52 | 3 | 1.05 | **985.03** | 3 | 219.60 | 0.00 | 991.96 | 766.80 | 0.70 | 994.48 | 3 | 0.95 | 993.97 | 3 | 922.44 | 0.90 | 1051.17 | 6.29 |
| RC208 | **836.29** | 3 | 838.03 | 3 | 0.21 | **836.29** | 3 | 238.80 | 0.00 | 836.29 | 913.80 | 0.01 | 841.34 | 3 | 0.60 | 844.03 | 3 | 608.64 | 0.92 | 902.26 | 7.32 |
| Average | | | | | 1.04$^a$ | | | 164.28$^a$ | 0.03$^a$ | | 845.59$^a$ | 0.65$^a$ | | | 0.65$^a$ | | | 726.77 | 0.83 | 549.97 | 2.08 |

VNS/TS on Schneider et al. [2014], ALNS$_1$ on Goeke e Schneider [2015], HGA on Hiermann et al. [2019], and ALNS$_2$ on Keskin e Çatay [2016].
$^a$: The average computational times and gap values were calculated considering only the solutions with the same number of vehicles.

This problem considered reducing the number of vehicles more important than reducing the distance traveled. Our algorithm was able to find reasonable solutions for large instances. We found nine of the BKS solutions reported in the literature. Some solutions have a smaller distance but a larger number of vehicles among the solutions found. These cases are considered better, the solution with fewer vehicles. The VNS found an equal number of vehicles in 53 of the 56 solutions reported in the literature. Among them, the solutions have an average gap of 0.15% for instances of class C, 0.80% for class R, and 0.83% for class RC.

Our algorithm performs better in instances of classes C and RC. The Flexi-VNS found 15 out of 17 BKS solutions in class C with an average gap of 0.16% and 9 of 16 BKS solutions in class RC with an average gap of 0.83%. To calculate the gap we consider only solutions with an equal number of vehicles from BKS. In class C, we only found one solution with more vehicles, which was the instance **C108**, and the distance was smaller than the BKS. In the RC instances, we get the same number of vehicles in all cases. In class R, Flexi-VNS improved or equaled the BKS solution value in 5 of 23 instances with a gap of 0.8% for instances with the same number of vehicles. In this class, Flexi-VNS gets the same number of vehicles as the literature, except for the instance **r207**.

## 6. Conclusion

This work addressed the Electric Vehicle Routing Problem with Time Windows, an important green logistics vehicle class variant. To solve the problem, a General Variable Neighborhood Search (GVNS)–based, named Flexi-VNS, was developed. The Flexi-VNS algorithm includes a Randomized Variable Neighborhood Descent (RVND) method as the local search procedure. The algorithm performance was tested using two benchmark sets presented by Schneider et al. [2014]. The first one, the small-size set, contains 36 instances, while the large-size instances contain 56 instances. The results obtained by Flexi-VNS were compared with the ones from several algorithms in the literature. Flexi-VNS found all BKS values reported in the literature for the small-size instances, which shows the method's efficiency. Concerning the large-size instances, our method was capable of improving the BKS value in seven instances and equaling the value in 22.

## Acknowledgements

The authors register their thanks to Universidade Federal de Ouro Preto (UFOP), Instituto Federal de São Paulo, and the Brazilian agencies CNPq (grant 311074/2023-5), FAPEMIG (grant PPM-CEX 676/17), and CAPES (finance code 001), for supporting the development of this work.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

**References**

Bräysy, O., Dullaert, W., Hasle, G., Mester, D., e Gendreau, M. (2008). An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle-routing problem with time windows. *Transportation Science*, 42(3):371–386. ISSN 15265447.

Bruglieri, M., Ferone, D., Festa, P., e Pisacane, O. (2022). A grasp with penalty objective function for the green vehicle routing problem with private capacitated stations. *Computers & Operations Research*, 143:105770. ISSN 0305-0548.

Erdelic, T., Carić, T., e Lalla-Ruiz, E. (2019). A survey on the electric vehicle routing problem: Variants and solution approaches. *Journal of Advanced Transportation*, 2019. ISSN 20423195.

Goeke, D. e Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99.

Hansen, P. e Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467.

Hansen, P., Mladenović, N., Brimberg, J., e Moreno Pérez, J. A. (2019). Variable neighborhood search. In Gendreau, M. e Potvin, J.-Y., editors, *Handbook of Metaheuristics*, volume 272 of *International Series in Operations Research & Management Science*, p. 57–97. Springer International Publishing, Cham. ISBN 978-3-319-91086-4.

Hansen, P., Mladenović, N., e Moreno Pérez, J. A. (2008). Variable neighbourhood search: Methods and applications. *40R*, 6(4):319–360.

Hansen, P., Mladenović, N., e Moreno Pérez, J. A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175:367–407.

Hiermann, G., Hartl, R. F., Puchinger, J., e Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1):235–248. ISSN 0377-2217.

Hiermann, G., Puchinger, J., Ropke, S., e Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018. ISSN 03772217.

Keskin, M. e Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65(216):111–127.

Keskin, M., Çatay, B., e 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.

Koç, Ç. e Karaoglan, I. (2016). The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing Journal*, 39:154–164. ISSN 15684946.

Kucukoglu, I., Dewil, R., e Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161(August):107650.

Lin, C., Choy, K. L., Chung, S. H., e Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, 41(4):1118–1138.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., e Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Martins, L. d. C., Tordecilla, R. D., Castaneda, J., Juan, A. A., e Faulin, J. (2021). Electric Vehicle Routing, Arc Routing, and Team Orienteering Problems in Sustainable Transportation. *Energies*, 14(16):5131.

McKinnon, A. C. e Piecyk, M. I. (2009). Measurement of CO2 emissions from road freight transport: A review of UK experience. *Energy Policy*, 37(10):3733–3742. ISSN 03014215.

Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100. ISSN 0305-0548.

Pan, B., Zhang, Z., e Lim, A. (2021). Multi-trip time-dependent vehicle routing problem with time windows. *European Journal of Operational Research*, 291(1):218–231. ISSN 0377-2217.

Peng, B., Zhang, Y., Gajpal, Y., e Chen, X. (2019). A memetic algorithm for the green vehicle routing problem. *Sustainability*, 11(21):6055.

Penna, P. H. V., Subramanian, A., Ochi, L. S., Vidal, T., e Prins, C. (2019). A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. *Annals of Operations Research*, 273:5–74.

Qin, H., Su, X., Ren, T., e Luo, Z. (2021). A review on the electric vehicle routing problems: Variants and algorithms. *Frontiers of Engineering Management*, 8(3):370–389.

Rodrigue, J.-P., Slack, B., e Comtois, C. (2017). Green logistics. In Brewer, A. M., Button, K. J., e Hensher, D. A., editors, *The Handbook of Logistics and Supply-Chain Management*, volume 2, p. 339–350. Emerald Group Publishing Limited.

Schneider, M., Stenger, A., e Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520. ISSN 15265447.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Souza, M. J. F., Coelho, I. M., Ribas, S., Santos, H. G., e Merschmann, L. H. d. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, 207(2):1041–1051.

Subramanian, A., Drummond, L. M. A., Bentes, C. B., Ochi, L. S., e Farias, R. d. S. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911.

Vidal, T., Crainic, T. G., Gendreau, M., e Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489.

Zhang, S., Gajpal, Y., e Appadoo, S. S. (2017). A meta-heuristic for capacitated green vehicle routing problem. *Annals of Operations Research*, 269(1-2):753–771.

**DOI:** 10.59254/sbpo-2024-193509
https://proceedings.science/p/193509?lang=pt-br