

AILS-II: An Adaptive Iterated Local Search Heuristic for the Large-scale Capacitated Vehicle Routing Problem

Vinícius R. Máximo^a, Jean-François Cordeau^b, Mariá C. V. Nascimento^{a,*}

^a*Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo (UNIFESP)*

Av. Cesare M. G. Lattes, 1201, Eugênio de Mello, São José dos Campos-SP, CEP: 12247-014, Brazil

^b*HEC Montréal and GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada*

Abstract

A recent study on the classical Capacitated Vehicle Routing Problem (CVRP) introduced an adaptive version of the widely used Iterated Local Search (ILS) paradigm, hybridized with a path-relinking strategy (PR). The solution method, called AILS-PR, outperformed existing meta-heuristics for the CVRP on benchmark instances. However, tests on large-scale instances of the CVRP suggested that PR was too slow, making AILS-PR less advantageous in this case. To overcome this challenge, this paper presents an Adaptive Iterated Local Search (AILS) with two phases in its search process. Both phases include the perturbation and local search steps of ILS. The main difference between them is that the reference solution in the first phase is found by the acceptance criterion, while in the second phase it is selected from a pool of the best solutions found in the search process, the so-called elite set. This algorithm, called AILS-II, is very competitive on smaller instances, outperforming the other methods from the literature with respect to the average gap to the best known solutions. Moreover, AILS-II consistently outperformed the state of the art on larger instances with up to 30,000 vertices.

Keywords: Combinatorial Optimization, Capacitated Vehicle Routing Problem (CVRP), Adaptive Iterated Local Search (AILS)

*Corresponding author

Email addresses: vinyamax10@gmail.com (Vinícius R. Máximo), jean-francois.cordeau@hec.ca (Jean-François Cordeau), mcv.nascimento@unifesp.br (Mariá C. V. Nascimento)

1. Introduction

The Capacitated Vehicle Routing Problem (CVRP) is a widely studied combinatorial optimization problem first introduced by [Dantzig and Ramser \(1959\)](#). It consists in finding a set of routes that minimizes the cost of making deliveries to a set of customers with a homogeneous vehicle fleet based at a central depot. There is a great interest in providing efficient solutions to this problem, given its important role in many supply chains. [Toth and Vigo \(2001\)](#) carried out a study that reported potential savings of around 5 to 20% in total transportation costs with the use of computer systems in real routing problems. Most of these problems are difficult to solve in reasonable time for large instances with hundreds or thousands of customers. In these cases, heuristic methods are often preferred as they provide high-quality feasible solutions when well designed.

A number of successful heuristic methods have been employed to tackle the CVRP, among which we highlight the recently proposed hybrid genetic search (HGS) ([Vidal, 2022](#)), slack induction by string removals (SISRs) ([Christiaens and Vanden Berghe, 2020](#)) and adaptive iterated local search with path-relinking (AILS-PR) ([Máximo and Nascimento, 2021](#)). In particular, AILS-PR outperformed the state-of-the-art metaheuristics considering small- and medium-sized instances, with up to 1000 vertices, proposed by [Uchoa et al. \(2017\)](#), [Christofides et al. \(1979\)](#) and [Golden et al. \(1998\)](#). However, recent experiments indicate that AILS-PR is too slow on large-scale instances. More specifically, the path-relinking (PR) strategy shows poor performance in these cases.

The great advantage of using PR in AILS lies in the diversification that it allows by intensifying the search in the region of an elite solution set. In AILS-PR, this set contains the best solutions found in the different iterations of the algorithm. Bearing this in mind, we propose in this paper to embed the core idea of PR in AILS, deriving a new strategy called AILS-II.

AILS-II has two phases, one that considers the traditional AILS flow and another that intensifies the search using the solutions of the elite set. This approach has an equivalent computational cost in the two phases regardless of the problem size. Computational experiments with small- and medium-size instances show that AILS-II is very competitive with AILS-PR and HGS. However, the experiments on instances with more than 700 customers indicate that AILS-II is consistently better than the other algorithms.

The rest of the paper is organized as follows. Section 2 presents the description of the CVRP and the main notations used throughout the paper. Section 3 introduces AILS-II and presents different approaches for controlling the perturbation degree and acceptance criteria. Section 4 reports

the results of computational experiments comparing AILS-II with AILS-PR and HGS. Finally, in Section 5, we draw some conclusions and suggest some future research directions.

2. Problem description

The Capacitated Vehicle Routing Problem (CVRP) can be modeled on an undirected graph $G = (V, E)$, where $V = \{0, 1, \dots, n\}$ is the set of $n + 1$ vertices and E the set of edges. The depot is represented by the vertex 0 and the other vertices in the subset $V_c = V \setminus \{0\}$ represent the n customers. Each edge $\{i, j\} \in E$ has a non-negative weight $d_{i,j}$ that represents the cost associated with a vehicle moving between vertices i and j .

Each customer $i \in V_c$ has a non-negative demand q_i that must be satisfied (the depot has demand $q_0 = 0$). To meet the demands of customers, m identical vehicles with a capacity of \bar{q} are used. To ensure the feasibility of the problem, the demand of each customer is assumed to be smaller than or equal to the vehicle capacity, that is, $q_i \leq \bar{q} \forall i \in V_c$. In the CVRP, each route is represented by a closed loop with no node repetition. A closed loop is represented by a cyclic sequence of vertices where a pair of vertices is adjacent if they are consecutive in the sequence and not adjacent otherwise. The CVRP, therefore, consists in finding a set of m routes that minimize the sum of the edge weights. The routes of a solution s are described by $\mathcal{R} = \{R_1^s, R_2^s, \dots, R_{m_s}^s\}$, where m_s represents the number of routes in the solution s , $R_i^s = \{v_0^i, v_1^i, \dots, v_{t_i}^i\}$, $t_i + 1$ is the length of route R_i^s , $v_0^i = v_{t_i}^i = 0$, $R_i^s \cap R_j^s = \{0\}$, for $i \neq j$, and $\cup_{i=1}^{m_s} R_i^s = V$.

The CVRP is NP-Hard (Lenstra and Rinnooy Kan, 1981) since it generalizes the Traveling Salesman Problem (TSP), which seeks to minimize the length of a Hamiltonian tour.

3. Adaptive Iterated Local Search II (AILS-II)

In this paper, we propose the Adaptive Iterated Local Search II (AILS-II) inspired by AILS. AILS is an adaptation of iterated local search (ILS) possessing diversity control mechanisms as an attempt to escape from local optima. A version of AILS was recently developed for the Heterogeneous Fleet Vehicle Routing Problem by Máximo et al. (2021). As mentioned earlier, Máximo and Nascimento (2021) proposed an AILS and PR hybrid to solve the CVRP, called AILS-PR. PR plays the role of intensifying the search for high-quality solutions in the region between pairs of elite solutions. AILS-PR performs very well on instances with up to 1000 vertices. However, it

does not scale well and empirical tests with large-scale instances indicate that PR becomes very time-consuming.

The AILS-II has two phases, and each of them follows the classical steps of ILS: perturbation and local search. The first phase works the same as in the AILS proposed by [Máximo and Nascimento \(2021\)](#). The perturbation strategy consists of a procedure that removes and adds vertices from and to the routes by removal and insertion heuristics, respectively. After a given number of iterations in the first phase, AILS-II switches to the second phase with a probability of 50%. The second phase feeds the method with a new reference solution randomly selected from an elite solution set \mathcal{E} to go through the perturbation and local search. The main reason for the two phases is to ensure that both *exploration* and *exploitation* are achieved. While the first phase has a more exploratory behavior, the second one is more focused on exploring the search space around the elite set of solutions.

Flowchart 1 presents AILS-II as a general framework. After reading the instance data, it creates an empty elite set \mathcal{E} and constructs an initial solution, s . The construction of an initial solution is the same strategy proposed in [Máximo and Nascimento \(2021\)](#). Then, a local search starting from solution s returns a local optimum in the neighborhood of s . The resulting solution is assigned to s_1^r , the variable that keeps the reference solution of the first phase of AILS-II, called Phase 1. Then, the main loop starts. The first iteration runs Phase 1, so the reference solution s^r of AILS-II is updated with s_1^r . The reference solution is modified by a vertex removal heuristic \mathcal{H}_k chosen at random in the perturbation step. The perturbation strategy is described in Section 3.1 and aims to generate a solution s different from s^r . Next, a local search looks for the best solution in the neighborhood of solution s . If the resulting solution is better than s^* , it becomes the incumbent. The adjustment of the perturbation degree $\omega_{\mathcal{H}_k}$ related to the selected removal heuristic \mathcal{H}_k occurs as described in Section 3.4. The perturbation degree concerns the number of elements to be removed by a removal heuristic. Then, the convergence speed of the algorithm is checked: if it is low, the second phase, called Phase 2, becomes active. A low convergence speed means that the number of iterations without improvement in the incumbent is greater than 2000. This mechanism allows the metaheuristic to promote a lighter initial convergence at the beginning of AILS-II. The elite set \mathcal{E} is updated as described in Section 3.3 only when Phase 2 is active. If Phase 1 is running, it checks the acceptance criterion: if it is met, then the reference solution of the first phase, s_1^r , is updated; otherwise, the acceptance criterion, described in Section 3.5, is updated. It is worth noting that

checking if the solution meets the acceptance criterion happens only in Phase 1. Then, AILS-II examines if the stopping criterion has been reached. If so, it returns the best solution, s^* , and halts. Otherwise, a phase is randomly picked if Phase 2 is active and $\mathcal{E} \neq \emptyset$. If not, the method picks Phase 1, starting a new iteration of such a phase having s_1^r as the reference solution. If Phase 2 is chosen, the reference solution is selected from \mathcal{E} .

3.1. Perturbation

The perturbation strategy employed in AILS-II consists in removing ω vertices from a given solution and then re-inserting them in other positions in the routes.

Algorithm 1 describes the perturbation method of a reference solution s^r employed in AILS-II. The input to this algorithm is the removal heuristic \mathcal{H}_k , which is randomly chosen. First, a copy of s^r is assigned to s' . Then, an insertion heuristic $\mathcal{A}_j \in \mathcal{A}$ is selected at random, and the perturbation process of removing and re-inserting the $\omega_{\mathcal{H}_k}$ vertices in s' happens. At the end of this process, the resulting solution s' may be infeasible, so we apply a feasibility algorithm to make it feasible. The removal heuristics employed in AILS-II are presented next.

- **Concentric removal, \mathcal{H}_1 :** This heuristic removes a set of $\omega_{\mathcal{H}_1}$ concentric vertices whose central node is chosen randomly.
- **Sequential removal, \mathcal{H}_2 :** This heuristic consists of removing a sequence of $\omega_{\mathcal{H}_2}$ consecutive vertices from a route. The starting point of the sequence is chosen at random. If during the removal process all elements of the route have been removed, then a new starting point is selected in another route and the process repeats.

AILS-II uses two insertion heuristics. Both heuristics restrict the addition of vertices in the same adjacency present in s^r . In this way, we force the generation of a solution s' different from s^r . If vertex v_i^j is adjacent to v_{i-1}^j and v_{i+1}^j in solution s^r , then the method must guarantee that the insertion heuristic does not add vertex v_i so that it is adjacent to v_{i-1}^j and v_{i+1}^j .

- **Cost-based insertion, \mathcal{A}_1 :** This heuristic consists in adding a vertex a into the position that results in the lowest possible cost. Let $\delta(a)$ be the set of neighbors of a . The vertex a is then inserted in the route \hat{j} , positioned immediately after \hat{i} in the sequence, where \hat{i} and \hat{j} are calculated as $(\hat{i}, \hat{j}) = \arg \min_{(i,j): v_i^j \in \mathcal{R}_i^s \cap \delta(a)} c(v_i^j, v_{i+1}^j, a)$ where $c(v_i^j, v_{i+1}^j, a) = d(v_i^j, a) + d(v_{i+1}^j, a) - d(v_i^j, v_{i+1}^j)$.

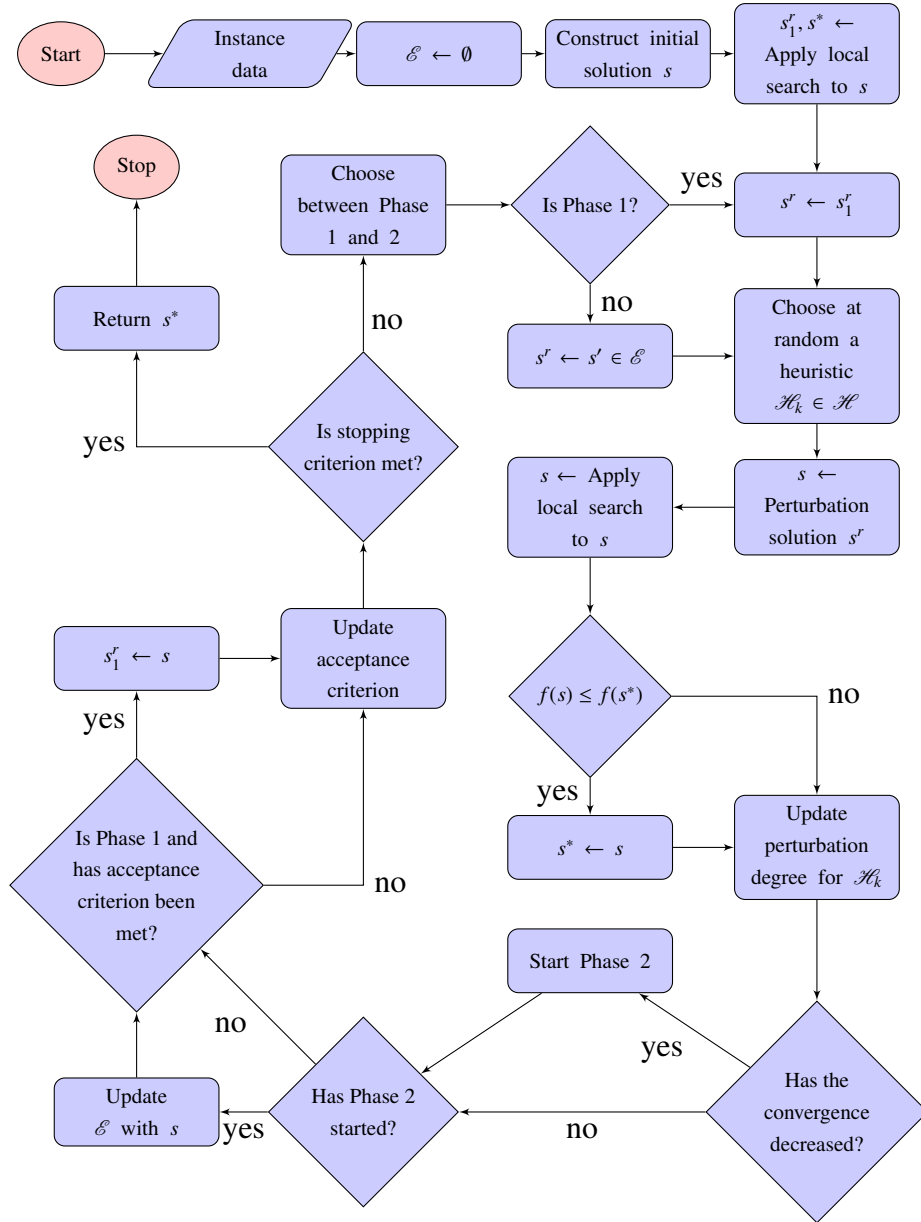


Figure 1: Flowchart of AILS-II.

- **Distance-based insertion, \mathcal{A}_2 :** This heuristic consists in inserting a vertex v in the position that makes it adjacent to its closest neighbor.

The employed feasibility procedure is briefly discussed in the next section.

Algorithm 1: Perturbation

Data: Solution s^r , \mathcal{H}_k

Result: Solution s'

```
1  $s' \leftarrow s^r$ 
2 Randomly select an insertion heuristic  $\mathcal{A}_j \in \mathcal{A}$ 
3 for  $l = 1, \dots, \omega_{\mathcal{H}_k}$  do
4   | Remove a vertex  $v_i$  from  $s'$  following heuristic  $\mathcal{H}_k$ 
5   | Insert  $v_i$  according to heuristic  $\mathcal{A}_j$ 
6 end
7  $s' \leftarrow \text{Feasibility}(s')$ 
```

3.2. Feasibility and Local Search

The feasibility and local search strategies used in AILS-II are the same as those presented by [Máximo and Nascimento \(2021\)](#). To make the current solution s feasible, the method changes the positions of the vertices in s to reduce the violation of vehicle capacity constraints. The inter-route movements used are:

- **Shift₁**: This movement consists in moving a vertex to a route different from the one that visits it in solution s .
- **Swap***: This movement was proposed by [Vidal \(2022\)](#) and consists of exchanging the routes of two vertices. The vertices are inserted in the least cost position. Therefore, the vertices do not necessarily swap their positions.
- **Cross**: This movement consists in swapping a sequence of vertices in a pair of distinct routes from a pair of reference points of the routes. Let v_k^i and v_l^j be the reference points to be swapped in routes R_i^s and R_j^s , respectively. According to this movement, $R_i^s = \{v_0^i, v_1^i, \dots, v_{k-1}^i, v_k^i, v_{k+1}^i, \dots, v_{t_i}^i\}$ and $R_j^s = \{v_0^j, v_1^j, \dots, v_{l-1}^j, v_l^j, v_{l+1}^j, \dots, v_{t_j}^j\}$ will become $\{v_0^j, v_1^j, \dots, v_{l-1}^j, v_l^i, v_{l+1}^j, \dots, v_{t_j}^j\}$ and $\{v_0^i, v_1^i, \dots, v_{k-1}^i, v_k^j, v_{k+1}^i, \dots, v_{t_i}^i\}$, respectively.

The intra-route movements used are:

- **Shift₁**: This movement consists of relocating the vertex position within the route.

- **Swap₁**: This movement consists in swapping the positions of a pair of vertices in the same route.
- **2 – opt**: This movement consists in the cross movement, but considering a pair of reference points of the same route.

In all movements, a pair of vertices v_i and v_j is used as reference points to perform the moves. We limit the neighborhood of a vertex v_i to investigate only those v_j among the φ vertices closest to v_i . We impose restrictions because empirical tests indicated that most of the best choices are between vertices close to each other. This approach is more relevant for large-scale instances. The choice of moves follows the best improvement strategy, optimizing the quality of the moves and the gain in feasibility, as discussed in [Máximo and Nascimento \(2021\)](#). If the investigated neighborhood of the given solution has only infeasible solutions, the procedure adds an empty route to the infeasible solution and restarts the neighborhood search until it finds a feasible solution. Local search is very similar to the feasibility procedure. The difference is that local search optimizes the solution value and is applied only to feasible solutions.

3.3. Update Elite \mathcal{E}

The elite set \mathcal{E} stores the best solutions found during the search. These solutions are used in Phase 2 of AILS-II to intensify the search. This set is limited to σ solutions. Algorithm 2 describes the update strategy used in AILS-II.

Let $MD_+^s = \{s_e \in \mathcal{E} \mid f(s_e) \geq f(s) \text{ and } d(s, s_e) \leq d_\beta\}$ and $MD_-^s = \{s_e \in \mathcal{E} \mid f(s_e) < f(s) \text{ and } d(s, s_e) \leq d_\beta\}$ be the sets of solutions whose distance from solution s is less than or equal to d_β . The set MD_+^s has solutions with a quality equal to or worse than s whereas MD_-^s has the best quality solutions. Consider also \hat{s}_e as the worst solution of the elite set \mathcal{E} . One of the requirements for a solution s to be part of the elite set is that $MD_-^s = \emptyset$. This means that solutions close to s with worse objective function values are not included in the elite set. Moreover, at least one of the following two conditions needs to be met for s to be included in the elite set: (i) the limit size of the elite set has not been reached; (ii) the solution s is better than or equal to the worst solution \hat{s}_e . If s has the credentials to be in the elite set, it is included in \mathcal{E} as long as every solution from MD_+^s is excluded from \mathcal{E} . However, if $|MD_+^s| = 0$ and $|\mathcal{E}| = \sigma$, the worst solution \hat{s}_e is removed \mathcal{E} .

The criterion for adding and removing solutions is that there is a minimum distance d_β between

the solutions in a subset. The goal of this approach is to diversify the characteristics of the solutions to be explored by AILS-II.

Algorithm 2: Updating of elite set \mathcal{E} .

Data: Solution s , \mathcal{E}

Result: Updated set \mathcal{E}

```

1 Consider  $MD_+^s = \{s_e \in \mathcal{E} \mid f(s_e) \geq f(s) \text{ and } d(s, s_e) \leq d_\beta\}$ 
2 Consider  $MD_-^s = \{s_e \in \mathcal{E} \mid f(s_e) < f(s) \text{ and } d(s, s_e) \leq d_\beta\}$ 
3 Consider  $\hat{s}_e = \arg \max_{s_e \in \mathcal{E}} f(s_e)$ 
4 if  $MD_-^s = \emptyset$  and  $(|\mathcal{E}| < \sigma \text{ or } f(s) \leq f(\hat{s}_e))$  then
5   if  $|MD_+^s| > 0$  then
6      $\mathcal{E} \leftarrow (\mathcal{E} \setminus MD_+^s) \cup \{s\}$ 
7   end
8   else
9     if  $|\mathcal{E}| < \sigma$  then
10       $\mathcal{E} \leftarrow \mathcal{E} \cup \{s\}$ 
11    end
12    else
13       $\mathcal{E} \leftarrow (\mathcal{E} \setminus \{\hat{s}_e\}) \cup \{s\}$ 
14    end
15  end
16 end

```

3.4. Perturbation Degree

The perturbation degree establishes the number of changes applied to the reference solution s' to achieve a different solution. The greater the perturbation degree, the larger the distance between these solutions. Thus, the control of the perturbation degree enables the method to manage the diversity of the search algorithm. For this reason, the perturbation degree control can be seen as a mechanism of great relevance for metaheuristics, considering that adequate control can allow the algorithm to escape from local optima. High diversity means that the algorithm will be able to escape from a local optimum more easily, but it might be costly to identify local optima (Lourenço

et al., 2010). On the other hand, low diversity yields a higher chance that the algorithm will get stuck in a local optimum. Avoiding these two scenarios is a difficult task, and this paper presents and evaluates some options for controlling the perturbation degree. As mentioned before, $\omega_{\mathcal{H}_k}$ denotes the number of vertices to be removed by the removal heuristic $\mathcal{H}_k \in \{\mathcal{H}_1, \mathcal{H}_2\}$ in the perturbation step. Accordingly, the larger $\omega_{\mathcal{H}_k}$ is, the greater the perturbation degree. Next, we present four mechanisms that will establish the value of $\omega_{\mathcal{H}_k}$. To simplify the notation, when the perturbation degree is the same for all removal heuristics, we refer to $\omega_{\mathcal{H}_k}$ as simply ω .

3.4.1. Fixed, \mathcal{D}_1 :

A fixed value for ω is defined for AILS-II. This means that all heuristics will have the same value for ω , a constant scalar throughout the search.

3.4.2. Relative, \mathcal{D}_2 :

The ω value is proportional to the instance size. Therefore, $\omega = \nu n$, where ν is a metaheuristic parameter. This criterion is the same used in [Morais et al. \(2014\)](#).

3.4.3. Random, \mathcal{D}_3 :

Choosing a single value for ω hardly is the best option for all instances. For this reason, [den Besten et al. \(2001\)](#) proposed to define random values for ω chosen within a range $[\underline{\omega}, \bar{\omega}]$ at each iteration. In this case, the values of $\underline{\omega}$ and $\bar{\omega}$ will be parameters of AILS-II.

3.4.4. Distance, \mathcal{D}_4 :

This approach adjusts the value of ω in such a way that the distance between the solution s obtained by the local search and the reference solution s^r is controlled by the parameter d_β . This process is calculated for each heuristic of $\mathcal{H}_k \in \{\mathcal{H}_1, \mathcal{H}_2\}$ so there is a $\omega_{\mathcal{H}_k}$ for each heuristic. If this distance is greater than d_β , then the value of $\omega_{\mathcal{H}_k}$ will be decreased; if it is lower, then we increase it. Thus, the algorithm will have a controlled diversity as a function of the distance $d(s, s^r)$ between the two solutions s^r and s . The distance metric used for this purpose consists in evaluating the number of different edges in the two solutions according to Equation (1). We define E^s as the set of edges of a solution s :

$$d(s, s^r) = |E^s \Delta E^{s^r}|. \quad (1)$$

The value of $\omega_{\mathcal{H}_k}$ is adjusted every γ iterations of the heuristic \mathcal{H}_k as follows: $\omega_{\mathcal{H}_k} = \min\{n, \max\{1, \omega_{\mathcal{H}_k} d_{\beta} / d_{\mathcal{H}_k}\}\}$, where $d_{\mathcal{H}_k}$ represents the average of the values of $d(s, s^r)$ since the last update. This mechanism for adjusting the perturbation degree is the same used in [Máximo and Nascimento \(2021\)](#).

3.5. Acceptance Criterion

The acceptance criterion establishes the rules for the iteration solution s to replace the current reference solution s'_1 of Phase 1. The acceptance criterion is an important diversity control mechanism of the algorithm ([Lourenço et al., 2003](#)). On the one hand, a very restrictive criterion implies less diversity. On the other hand, a very relaxed criterion provides a high diversity for the algorithm. For this reason, finding the ideal criterion is an important task.

There are many possibilities to define the acceptance criterion. One of the best known is the acceptance criterion of the Simulated Annealing (SA) algorithm proposed by [Kirkpatrick et al. \(1983\)](#). According to this criterion, solutions worse than the current one are accepted with a probability that decreases throughout the search process. [Dueck and Scheuer \(1990\)](#) proposed the Threshold Accepting (TA), which is a metaheuristic that uses a convergent threshold as an acceptance criterion. The main difference between TA regarding SA is that all solutions below the threshold are accepted. There is no probabilistic test for every solution. In TA, the threshold starts with a high value and throughout the search this value decreases, generating a convergent effect. The results presented in [Dueck and Scheuer \(1990\)](#) show that TA obtained better solutions while consuming less computing time when compared to SA. Backtracking Adaptive Threshold Accepting (BATA) is an improved version of TA, proposed by [Tarantilis et al. \(2004\)](#). This algorithm has a convergent threshold along the same lines as the TA, however, at certain times, the threshold may increase instead of decreasing steadily.

In this paper, we investigate seven different acceptance criteria and evaluate the performance of each one in AILS-II. Next, we present each of the criteria considered.

3.5.1. Threshold, \mathcal{C}_1 :

In this criterion, a threshold indicates the minimum quality that the solution s must have to update the solution s^r . This threshold is represented by $\bar{b} = \underline{f} + \eta(\bar{f} - \underline{f})$, where $\eta \in [0, 1]$ is a parameter. This parameter establishes a percentage between the average quality of the solutions

obtained by the local search called \bar{f} and the best solution found in the last γ iterations, referred to as \underline{f} . The average \bar{f} is calculated dynamically, according to Equation (2):

$$\bar{f} = \begin{cases} \bar{f}(1 - \gamma^{-1}) + f(s)\gamma^{-1}, & \text{if } it > \gamma \\ (\bar{f}(it - 1) + f(s))it^{-1}, & \text{if } it \leq \gamma \end{cases} \quad (2)$$

Thus, the solution s will be accepted if $f(s) \leq \bar{b}$.

3.5.2. Flow, \mathcal{C}_2 :

In this criterion, the volume of accepted solutions is controlled. For this, it employs the same mechanism found in \mathcal{C}_1 , by varying the η value so that the ideal flow established by the parameter $\kappa \in [0, 1]$ is achieved. The value of η is dynamically adjusted by induction at every γ accepted solutions so that $\eta = \kappa\eta/\kappa^r$, where κ^r represents the actual percentage of accepted solutions since the last update.

3.5.3. Distance, \mathcal{C}_3 :

This acceptance criterion uses the average distance of the accepted solutions concerning their respective reference solutions to adjust the threshold of the acceptance criterion. This criterion also uses the threshold mechanism defined in \mathcal{C}_1 . The particularity lies in the value of η , which is adjusted at each γ accepted solutions. The ideal distance is represented by μd_β , where $\mu \in [0, 1]$ is an acceptance criterion parameter. The adjustment of η is given by $\eta = \mu d_\beta \eta / d_a^r$, where d_a^r indicates the average distance of the accepted solutions to s^r since the last update.

3.5.4. Convergent, \mathcal{C}_4 :

In this criterion, we use a convergent behavior throughout the search. In the first iterations, the method imposes a more relaxed criterion that gradually becomes more restrictive as the search process evolves. This criterion was inspired by the TA algorithm proposed in [Dueck and Scheuer \(1990\)](#). To define \mathcal{C}_4 , we use the constraint mechanism \mathcal{C}_1 but update the η value at every iteration. Initially, $\eta = 1$. Then, at each iteration, we decrease its value by the following rule: $\eta \leftarrow \eta\alpha$, where $\alpha \in]0, 1[$.

The value of α is adjusted so that the value of η starts at 1 and ends at a small value, let us say ϵ . Therefore, α is adjusted at every λ iterations as $\alpha \leftarrow \epsilon^{t_a/\lambda}$, where t_a represents the elapsed

running time, t_f is the total running time of the algorithm (it is fixed) and it is the number of iterations executed so far. In this paper, ϵ takes the value 0.01.

3.5.5. *Relative Threshold*, \mathcal{C}_5 :

In this acceptance criterion, a threshold $\bar{b} = f(s^*) + \theta f(s^*)$ restricts the solutions that will be accepted. In this case, a solution s will be accepted if $f(s) \leq \bar{b}$. The value of \bar{b} is calculated using the parameter $\theta \in [0, 1]$ that indicates the percentage of the objective function of the best solution found s^* that must be added. This criterion is the same as used in [Morais et al. \(2014\)](#).

3.5.6. *k-best*, \mathcal{C}_6 :

This acceptance criterion uses the best solution s obtained in the k last iterations to update the reference solution s^r . This criterion does not take the quality of s^r into account, that is, it can improve or deteriorate every k iterations. This mechanism is similar to the roulette wheel mechanism implemented in genetic algorithms.

3.5.7. *Best*, \mathcal{C}_7 :

This acceptance criterion only updates the solution s^r if the presented solution s has a quality equal to or higher than the best solution found s^* . This criterion has a behavior of great intensity and little diversity, being more suitable for shorter execution times ([den Besten et al., 2001](#)). This criterion is the same used in [Penna et al. \(2013\)](#) and [Stützle \(2006\)](#).

4. Computational Experiments

The computational experiments were performed on a cluster with 104 nodes, each of them with 2 Intel Xeon E5-2680v2 processors running at 2.8 GHz, 10 cores, and 128 GB DDR3 of 1866 MHz RAM. AILS-II was developed in Java. In the experiments, we investigate the performance of the proposed algorithm using the set of benchmark instances presented in [Uchoa et al. \(2017\)](#). This set contains 100 CVRP instances whose size ranges from 100 to 1000 vertices. To assess the performance of AILS-II on larger instances we also consider the 10 instances proposed in [Arnold et al. \(2019\)](#) which have between 3000 and 30,000 vertices. All experiments were carried out using as stopping criterion $10n$ seconds, where n is the number of vertices of the instance.

Next, we show the impact of the different acceptance criteria presented in the previous section.

4.1. Assessment of Acceptance Criteria

This section presents the performance evaluation of AILS-II using the acceptance criteria introduced in Section 3.5. In this experiment, we only show the results of the perturbation degree control mechanism Distance \mathcal{D}_4 . The reason behind this is that AILS-II achieved the best results with this approach. The parameter tuning of AILS-II considering each acceptance criterion was performed to better evaluate the tested criteria. The tuning was carried out in the same way presented in Section 4.3 and Table 1 reports the results. This table shows the best parameter values achieved for each acceptance criterion. After the tuning, we calculate the average gap of 10 runs of the algorithm for the 100 benchmark instances to obtain the box-plot presented in Figure 2.

Table 1: Parameters and values assessed for each acceptance criterion.

Criterion	Parameter	Type	Tested Values	Best Value
Threshold, \mathcal{C}_1	η	Real	{0.1, 0.2, ..., 0.7}	0.4
Flow, \mathcal{C}_2	κ	Real	{0.1, 0.2, ..., 0.7}	0.4
Distance, \mathcal{C}_3	μ	Real	{0.2, 0.3, ..., 0.8}	0.5
Convergent, \mathcal{C}_4	–	–	–	–
Relative Threshold, \mathcal{C}_5	θ	Real	{0.001, 0.005, 0.01, 0.015, 0.02}	0.005
k -best, \mathcal{C}_6	k	Integer	{3, 4, ..., 8}	6
Best, \mathcal{C}_7	–	–	–	–

One can observe that AILS-II with the acceptance criterion Best \mathcal{C}_7 obtained the worst results. One of the possible reasons is that this approach may allow little diversity and high intensity. Therefore, AILS-II was more likely to get stuck in local optima. Criterion \mathcal{C}_5 had the second-worst result. This acceptance criterion has a threshold that represents a percentage of the quality of the best solutions found. It did not fit properly when the magnitude of the solution values varied in the different instances. AILS-II achieved the third-worst results when it adopted the acceptance criterion \mathcal{C}_3 . Despite a very similar performance with AILS-II considering \mathcal{C}_1 , \mathcal{C}_1 was slightly better than \mathcal{C}_3 . Criterion \mathcal{C}_2 , presented by [Máximo and Nascimento \(2021\)](#), is based on the flow of accepted solutions and presented the third best results. AILS-II obtained the second-best results considering criterion \mathcal{C}_6 , which achieved a good trade-off between intensity and diversity. AILS-II, however, found the best results with the convergent acceptance criterion, \mathcal{C}_4 . This criterion provided a reasonable diversity at the beginning of AILS-II, enabling the method to escape from

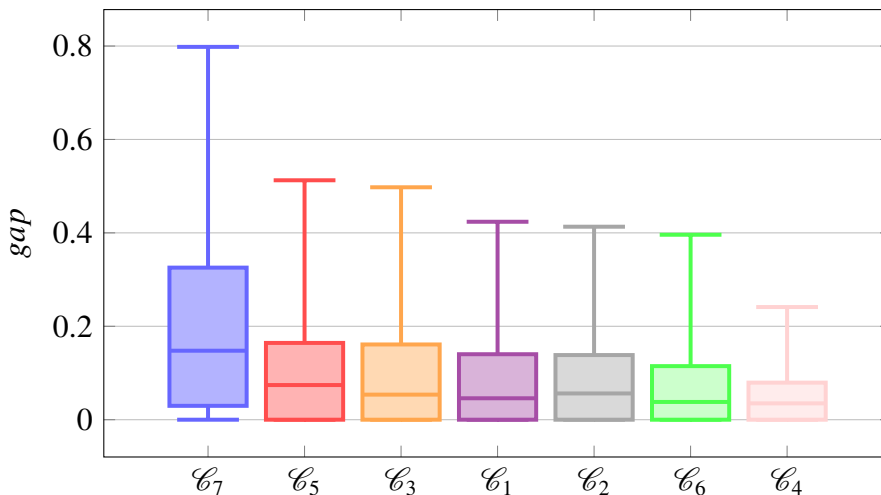


Figure 2: Box-plots of the gaps of AILS-II in 100 benchmark instances considering each of the 7 investigated acceptance criteria with the best parameter set up.

local optima in this phase, driving a more intensification-like search in the final iterations of AILS-II. Therefore, we chose \mathcal{C}_4 as the acceptance criterion of AILS-II in the other experiments.

4.2. Assessment of Perturbation Degree Mechanisms

In this section, we evaluate the performance of the approaches presented in Section 3.4 to control the perturbation degree of AILS-II. To assess these mechanisms, we tune each one following the strategy shown in Section 4.3. The acceptance criterion used was the convergent \mathcal{C}_4 and Table 2 reports the results. This table presents the mechanisms, the parameters, the values considered for the parameterization followed by the obtained best parameter value. We ran each of the 100 instances 10 times and the gaps obtained are displayed in the box plots in Figure 3.

Table 2: Parameters and values assessed for each mechanism for the perturbation degree.

Mechanism	Parameter	Type	Tested Values	Best Value
Fixed, \mathcal{D}_1	ω	Integer	{5, 10, ..., 40}	15
Relative, \mathcal{D}_2	ν	Real	{0.01, 0.02, ..., 0.05}	0.03
Random, \mathcal{D}_3	$\underline{\omega}, \bar{\omega}$	Integer	{1, 5, 10, 15, 20, 25, 30, 35, 40}	$\underline{\omega} = 1, \bar{\omega} = 30$
Distance, \mathcal{D}_4	d_β	Integer	{10, 15, ..., 40}	25

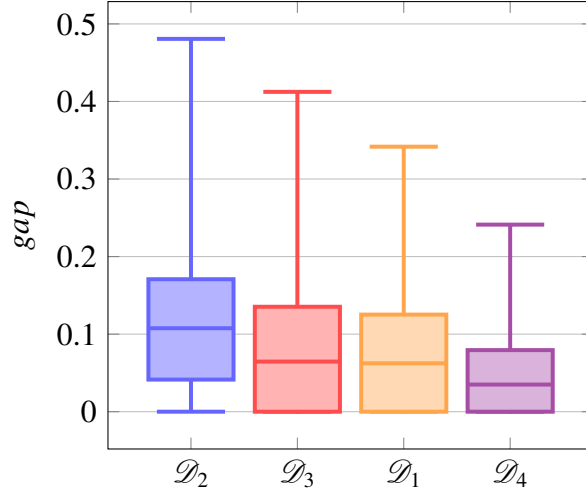


Figure 3: Box plots of the gaps of AILS-II in 100 benchmark instances considering each of the four investigated perturbation criteria with the best parameter set up.

AILS-II achieved the worst results considering mechanism \mathcal{D}_2 . The parameterization result indicates that the ideal ω value did not increase according to the instance size for this problem. For this reason, we believe that this approach has not had a good result. AILS-II with the mechanism \mathcal{D}_1 returned results slightly better than AILS-II with \mathcal{D}_2 . Given the result, the mechanism \mathcal{D}_1 is more advantageous, as it has a single parameter. AILS-II with the mechanism \mathcal{D}_4 obtained the best results. So, \mathcal{D}_4 was the strategy adopted in AILS-II in the remaining experiments.

4.3. Parameter tuning

AILS-II has five parameters considering the convergent acceptance criterion \mathcal{C}_4 and the perturbation degree adjustment mechanism \mathcal{D}_4 .

- d_β : indicates the minimum distance between the solutions of the elite set \mathcal{E} and in the control of the perturbation degree in \mathcal{D}_4 .
- γ : indicates the amount of sampled data needed to update the perturbation degree.
- σ : sets the maximum size that a set \mathcal{E} may have.
- φ : sets the maximum size that a set $\delta(v_k)$ may have.

The tuning experiments of the AILS-II parameters involved a set of 100 instances proposed by Uchoa et al. (2017). The parameter adjustment optimized the gap of the average solution considering 10 executions in each instance. Such an average solution value is called *Avg*. The gap was calculated as $gap = 100(Avg - BKS)/BKS$. Next, we describe the values considered for each parameter:

- $d_\beta \in \{10, 15, \dots, 40\}$
- $\sigma \in \{20, 40, \dots, 120\}$
- $\gamma \in \{10, 20, \dots, 60\}$
- $\varphi \in \{20, 40, \dots, 100\}$

The parameter tuning strategy involved varying the parameters in their respective intervals keeping the other parameters fixed. Initially, the parameters were set with the following configuration: $d_\beta = 25$, $\sigma = 60$, $\gamma = 30$ and $\varphi = 60$. These values were obtained in preliminary experiments optimizing the gaps. We chose the value that presented the best gap and adopted this value for the following rounds. This process was repeated until all parameters were fixed. In line with this, the parameters with greater sensitivity, detected in preliminary tests and shown in Figure 4, were d_β and φ , in this order. Then, γ and σ , which were not as sensitive as the other parameters, were tuned, respectively. Figure 4 illustrates the results of the parameter set up.

Table 3 shows the parameter configuration reached at the end of this process.

Table 3: Values employed in AILS-II after the set up of parameters.

Parameter	Type	Interval	Value
d_β	Integer	[10, 40]	25
σ	Integer	[20, 120]	60
γ	Integer	[10, 60]	30
φ	Integer	[20, 100]	40

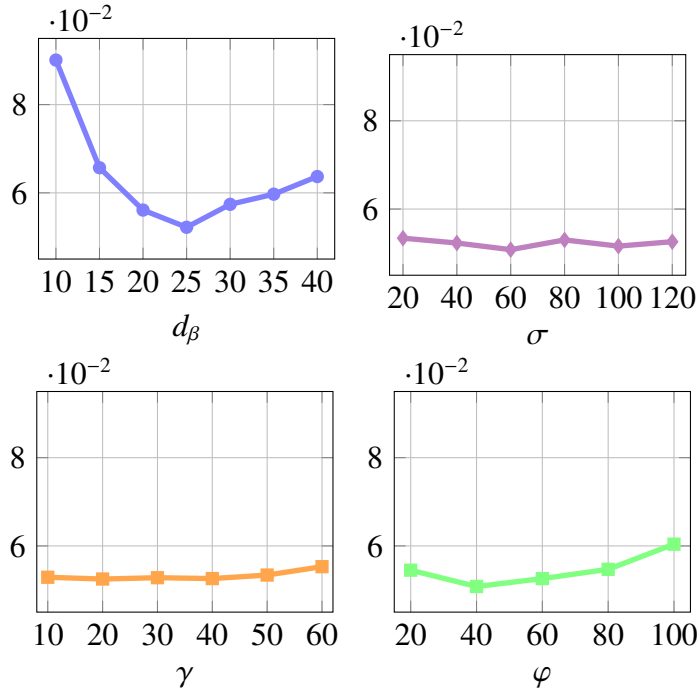


Figure 4: Final result obtained in the configuration of the parameters.

4.4. Experiments

The tests were performed on two sets of instances: the 100 small- and medium-sized instances proposed by Uchoa et al. (2017) – Experiment I; and the large instances introduced by Arnold et al. (2019) – Experiment II. In both experiments, we compare the proposed AILS-II with AILS-PR (Máximo and Nascimento, 2021) and HGS (Vidal, 2022) (all tests were run on the same micro-computer and adopting the stopping criterion of $10n$ seconds. The information considered in these experiments were:

- BKS: the objective function value of the best known solutions. Values highlighted in bold are optimal solutions according to the information available in the repository <http://vrp.atd-lab.inf.puc-rio.br>.
- Avg: average objective function value of the solutions obtained in 50 runs (instances Uchoa et al. (2017)) and 10 runs (instances of Arnold et al. (2019)).
- *gap*: gap between Avg and BKS, calculated as explained earlier in this section.

- Best: objective function value of the best solution obtained in the runs.
- T (min): average time in minutes that the algorithm took to find the best solution in each run.

The results of Experiment I are presented in Tables 4, 5 and 6. AILS-II, HGS and AILS-PR presented a better average gap for 71%, 46% and 34% of the instances, respectively. This result is quite relevant, as both HGS and AILS-PR are the state of the art for the CVRP. Moreover, AILS-II achieved the best solutions in 88% of the instances while HGS and AILS-PR obtained the best results in 57% and 59%, respectively. As the algorithms were executed on the same machine and with the same stopping criterion, the computational time was relatively close for the two algorithms.

To analyze the result of the algorithms in relation to the size of the instances, we divided the 100 instances considered in Experiment I into four groups containing 25 instances each: X-n101-214, X-n219-331, X-n336-573 and X-n586-1001. The average gap obtained for each group is illustrated in the boxplot graph shown in Figure 5. For small instances, AILS-II does not present a good result in relation to the other algorithms. However, as the size of the instances increases, AILS-II significantly outperforms the other two algorithms.

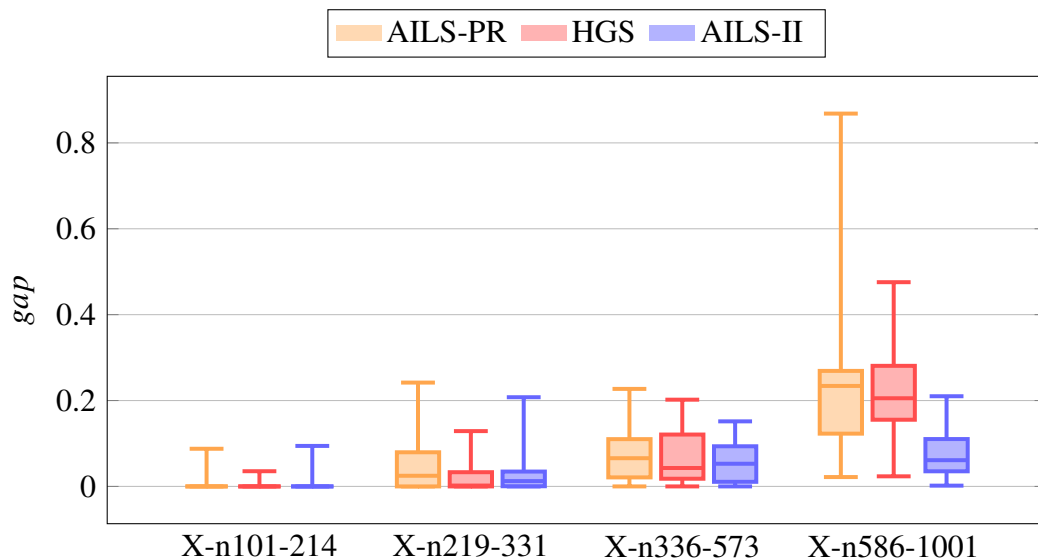


Figure 5: Boxplots of the average gap per group of instances – Experiment I.

Experiment II demonstrates the behavior of the algorithms on large instances proposed by Arnold et al. (2019). Table 7 presents the results of this experiment. AILS-II achieved a smaller

Table 4: Part I of the results considering the instances introduced by [Uchoa et al. \(2017\)](#).

Instance	AILS-PR (Máximo and Nascimento, 2021)				HGS (Vidal, 2022)			AILS-II		
	BKS	Avg (<i>gap</i>)	Best	T (min)	Avg (<i>gap</i>)	Best	T (min)	Avg (<i>gap</i>)	Best	T (min)
X-n101-k25	27591*	27591.00 (0.0000)	27591	0.12	27591.00 (0.0000)	27591	0.02	27591.00 (0.0000)	27591	0.45
X-n106-k14	26362*	26362.00 (0.0000)	26362	1.81	26371.32 (0.0354)	26362	7.90	26362.00 (0.0000)	26362	0.96
X-n110-k13	14971*	14971.00 (0.0000)	14971	0.02	14971.00 (0.0000)	14971	0.01	14971.00 (0.0000)	14971	0.07
X-n115-k10	12747*	12747.00 (0.0000)	12747	0.04	12747.00 (0.0000)	12747	0.02	12747.00 (0.0000)	12747	0.32
X-n120-k6	13332*	13332.00 (0.0000)	13332	0.19	13332.00 (0.0000)	13332	0.13	13332.00 (0.0000)	13332	0.35
X-n125-k30	55539*	55539.00 (0.0000)	55539	0.73	55539.00 (0.0000)	55539	0.35	55539.00 (0.0000)	55539	2.81
X-n129-k18	28940*	28940.00 (0.0000)	28940	3.13	28940.00 (0.0000)	28940	0.13	28940.00 (0.0000)	28940	0.55
X-n134-k13	10916*	10916.00 (0.0000)	10916	0.36	10916.00 (0.0000)	10916	0.43	10916.00 (0.0000)	10916	0.54
X-n139-k10	13590*	13590.00 (0.0000)	13590	0.20	13590.00 (0.0000)	13590	0.02	13590.00 (0.0000)	13590	0.58
X-n143-k7	15700*	15700.00 (0.0000)	15700	2.12	15700.00 (0.0000)	15700	0.17	15700.00 (0.0000)	15700	0.87
X-n148-k46	43448*	43448.00 (0.0000)	43448	0.11	43448.00 (0.0000)	43448	0.08	43448.00 (0.0000)	43448	0.50
X-n153-k22	21220*	21220.30 (0.0014)	21220	7.91	21224.70 (0.0221)	21220	1.41	21220.90 (0.0042)	21220	9.77
X-n157-k13	16876*	16876.00 (0.0000)	16876	0.12	16876.00 (0.0000)	16876	0.08	16876.00 (0.0000)	16876	0.71
X-n162-k11	14138*	14138.00 (0.0000)	14138	0.42	14138.00 (0.0000)	14138	0.04	14138.00 (0.0000)	14138	0.54
X-n167-k10	20557*	20557.00 (0.0000)	20557	0.43	20557.00 (0.0000)	20557	0.29	20557.00 (0.0000)	20557	1.25
X-n172-k51	45607*	45607.00 (0.0000)	45607	0.36	45607.00 (0.0000)	45607	0.09	45607.00 (0.0000)	45607	4.34
X-n176-k26	47812*	47812.00 (0.0000)	47812	0.44	47812.00 (0.0000)	47812	0.34	47812.00 (0.0000)	47812	0.87
X-n181-k23	25569*	25569.12 (0.0005)	25569	6.92	25569.00 (0.0000)	25569	0.85	25569.00 (0.0000)	25569	1.49
X-n186-k15	24145*	24145.04 (0.0002)	24145	7.63	24145.00 (0.0000)	24145	0.31	24145.00 (0.0000)	24145	3.56
X-n190-k8	16980*	16980.00 (0.0000)	16980	7.03	16981.04 (0.0061)	16980	14.72	16980.00 (0.0000)	16980	6.14
X-n195-k51	44225*	44225.00 (0.0000)	44225	0.71	44225.00 (0.0000)	44225	2.68	44231.68 (0.0151)	44225	17.43
X-n200-k36	58578*	58578.00 (0.0000)	58578	2.51	58578.00 (0.0000)	58578	5.69	58578.00 (0.0000)	58578	6.69
X-n204-k19	19565*	19565.00 (0.0000)	19565	5.86	19565.00 (0.0000)	19565	1.61	19565.00 (0.0000)	19565	2.20
X-n209-k16	30656*	30656.00 (0.0000)	30656	2.59	30656.00 (0.0000)	30656	5.07	30656.06 (0.0002)	30656	8.27
X-n214-k11	10856*	10865.52 (0.0877)	10856	13.38	10859.64 (0.0336)	10856	15.78	10866.24 (0.0943)	10856	12.81
X-n219-k73	117595*	117595.00 (0.0000)	117595	0.27	117595.00 (0.0000)	117595	5.34	117595.00 (0.0000)	117595	0.36
X-n223-k34	40437*	40437.00 (0.0000)	40437	12.16	40437.00 (0.0000)	40437	3.89	40437.16 (0.0004)	40437	9.49
X-n228-k23	25742*	25742.76 (0.0030)	25742	7.28	25742.26 (0.0010)	25742	12.41	25742.00 (0.0000)	25742	5.58
X-n233-k16	19230*	19230.60 (0.0031)	19230	11.89	19230.00 (0.0000)	19230	2.90	19230.32 (0.0017)	19230	7.38
X-n237-k14	27042*	27042.88 (0.0033)	27042	8.14	27042.00 (0.0000)	27042	2.53	27043.04 (0.0038)	27042	5.50
X-n242-k48	82751*	82818.38 (0.0814)	82764	15.19	82778.45 (0.0332)	82751	16.43	82759.36 (0.0101)	82751	20.46
X-n247-k50	37274*	37274.00 (0.0000)	37274	0.93	37274.00 (0.0000)	37274	14.80	37274.00 (0.0000)	37274	4.35
X-n251-k28	38684*	38715.06 (0.0803)	38684	24.74	38684.79 (0.0020)	38684	14.59	38695.04 (0.0285)	38684	16.74
X-n256-k16	18839	18879.28 (0.2138)	18864	1.32	18839.00 (0.0000)	18839	6.97	18878.12 (0.2077)	18839	2.65

Table 5: Part II of the results considering the instances introduced by [Uchoa et al. \(2017\)](#).

Instance	AILS-PR (Máximo and Nascimento, 2021)				HGS (Vidal, 2022)			AILS-II		
	BKS	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)
X-n261-k13	26558*	26559.40 (0.0053)	26558	20.04	26558.05 (0.0002)	26558	16.15	26558.96 (0.0036)	26558	17.49
X-n266-k58	75478*	75551.94 (0.0980)	75529	17.92	75575.22 (0.1288)	75478	21.10	75495.04 (0.0226)	75478	17.04
X-n270-k35	35291*	35299.74 (0.0248)	35291	14.88	35302.67 (0.0331)	35291	5.32	35303.20 (0.0346)	35291	15.68
X-n275-k28	21245*	21245.00 (0.0000)	21245	2.99	21245.00 (0.0000)	21245	0.58	21245.00 (0.0000)	21245	1.92
X-n280-k17	33503	33529.64 (0.0795)	33503	20.48	33515.24 (0.0365)	33503	23.34	33532.30 (0.0875)	33503	24.96
X-n284-k15	20215*	20252.02 (0.1831)	20225	25.27	20237.56 (0.1116)	20225	27.86	20236.86 (0.1081)	20215	28.16
X-n289-k60	95151*	95203.30 (0.0550)	95151	27.53	95244.55 (0.0983)	95185	24.22	95196.34 (0.0477)	95151	30.74
X-n294-k50	47161	47174.56 (0.0288)	47167	20.97	47172.48 (0.0244)	47161	26.91	47176.28 (0.0324)	47161	28.09
X-n298-k31	34231*	34231.00 (0.0000)	34231	4.93	34231.00 (0.0000)	34231	8.64	34235.86 (0.0142)	34231	11.18
X-n303-k21	21736	21749.56 (0.0624)	21746	18.44	21740.97 (0.0229)	21738	24.25	21746.92 (0.0502)	21736	28.32
X-n308-k13	25859	25921.52 (0.2418)	25859	22.79	25863.84 (0.0187)	25861	23.81	25873.44 (0.0558)	25861	23.58
X-n313-k71	94043	94084.56 (0.0442)	94046	31.13	94082.84 (0.0424)	94045	28.20	94062.72 (0.0210)	94044	33.03
X-n317-k53	78355*	78355.00 (0.0000)	78355	6.58	78355.83 (0.0011)	78355	14.79	78355.00 (0.0000)	78355	3.67
X-n322-k28	29834*	29845.74 (0.0394)	29834	24.01	29842.10 (0.0272)	29834	20.96	29837.70 (0.0124)	29834	18.58
X-n327-k20	27532	27559.54 (0.1000)	27532	23.39	27541.55 (0.0347)	27532	25.11	27567.28 (0.1281)	27532	27.12
X-n331-k15	31102*	31103.24 (0.0040)	31103	16.76	31102.97 (0.0031)	31102	19.98	31103.00 (0.0032)	31103	12.44
X-n336-k84	139111	139202.40 (0.0657)	139111	33.24	139260.93 (0.1078)	139174	35.82	139204.78 (0.0674)	139130	37.12
X-n344-k43	42050	42065.34 (0.0365)	42050	26.81	42067.18 (0.0409)	42050	26.06	42089.32 (0.0935)	42055	27.17
X-n351-k40	25896	25936.98 (0.1582)	25917	37.70	25942.56 (0.1798)	25925	36.86	25935.22 (0.1515)	25900	39.69
X-n359-k29	51505	51540.96 (0.0698)	51505	33.63	51601.33 (0.1870)	51512	36.95	51524.56 (0.0380)	51505	39.74
X-n367-k17	22814	22814.00 (0.0000)	22814	6.80	22814.00 (0.0000)	22814	3.07	22814.88 (0.0039)	22814	26.08
X-n376-k94	147713*	147713.00 (0.0000)	147713	4.54	147713.40 (0.0003)	147713	20.72	147713.00 (0.0000)	147713	2.75
X-n384-k52	65928	66022.52 (0.1434)	65941	41.44	66016.48 (0.1342)	65984	34.09	65974.30 (0.0702)	65938	43.71
X-n393-k38	38260*	38262.50 (0.0065)	38260	24.40	38260.00 (0.0000)	38260	7.44	38268.86 (0.0232)	38260	31.96
X-n401-k29	66154	66202.66 (0.0736)	66169	39.97	66215.33 (0.0927)	66179	29.67	66188.96 (0.0528)	66162	50.37
X-n411-k19	19712	19756.76 (0.2271)	19723	33.66	19716.70 (0.0238)	19712	36.02	19731.02 (0.0965)	19717	53.38
X-n420-k130	107798*	107837.44 (0.0366)	107818	37.92	107817.35 (0.0179)	107798	37.51	107809.56 (0.0107)	107798	39.53
X-n429-k61	65449	65484.60 (0.0544)	65453	33.49	65477.05 (0.0429)	65449	32.18	65495.94 (0.0717)	65457	48.42
X-n439-k37	36391*	36408.14 (0.0471)	36391	24.21	36395.09 (0.0112)	36391	15.96	36400.10 (0.0250)	36395	18.04
X-n449-k29	55233	55290.32 (0.1038)	55236	42.46	55334.05 (0.1829)	55296	40.67	55288.26 (0.1000)	55240	57.30
X-n459-k26	24139	24160.02 (0.0871)	24140	41.15	24152.71 (0.0568)	24139	30.59	24163.66 (0.1022)	24139	42.67
X-n469-k138	221824*	221958.92 (0.0608)	221841	50.16	222091.90 (0.1208)	221992	36.13	221943.28 (0.0538)	221842	63.41
X-n480-k70	89449	89467.74 (0.0210)	89449	52.73	89490.80 (0.0467)	89458	44.60	89456.70 (0.0086)	89449	39.71
X-n491-k59	66483	66556.18 (0.1101)	66487	59.37	66603.26 (0.1809)	66516	48.59	66545.86 (0.0946)	66489	61.62

Table 6: Part III of the results considering the instances introduced by Uchoa et al. (2017).

Instance	AILS-PR (Máximo and Nascimento, 2021)				HGS (Vidal, 2022)			AILS-II		
	BKS	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)	Avg (gap)	Best	T (min)
X-n502-k39	69226	69234.96 (0.0129)	69226	42.59	69238.95 (0.0187)	69228	35.63	69231.64 (0.0081)	69226	52.39
X-n513-k21	24201	24229.80 (0.1190)	24201	44.99	24201.00 (0.0000)	24201	8.88	24229.02 (0.1158)	24201	32.70
X-n524-k153	154593*	154596.36 (0.0022)	154593	25.86	154650.40 (0.0371)	154605	39.76	154601.74 (0.0057)	154593	67.18
X-n536-k96	94846	94972.96 (0.1339)	94906	69.21	95037.56 (0.2020)	94977	53.29	94896.02 (0.0527)	94830	65.29
X-n548-k50	86700*	86801.28 (0.1168)	86704	53.01	86744.53 (0.0514)	86706	51.76	86719.22 (0.0222)	86700	47.30
X-n561-k42	42717	42754.86 (0.0886)	42719	62.08	42733.76 (0.0392)	42719	45.53	42751.34 (0.0804)	42728	72.24
X-n573-k30	50673	50763.96 (0.1795)	50701	68.02	50767.38 (0.1862)	50744	58.32	50730.34 (0.1132)	50680	81.46
X-n586-k159	190316	190397.26 (0.0427)	190336	67.26	190519.19 (0.1068)	190427	53.50	190379.70 (0.0335)	190316	72.57
X-n599-k92	108451	108586.38 (0.1248)	108455	69.63	108619.38 (0.1553)	108563	71.66	108553.66 (0.0947)	108472	77.41
X-n613-k62	59535	59583.44 (0.0814)	59544	75.86	59668.40 (0.2241)	59582	46.95	59569.56 (0.0580)	59536	74.25
X-n627-k43	62164	62326.34 (0.2611)	62221	74.96	62324.47 (0.2581)	62264	71.23	62205.46 (0.0667)	62168	89.10
X-n641-k35	63682	63831.40 (0.2346)	63752	77.06	63836.07 (0.2419)	63751	58.86	63752.26 (0.1103)	63696	79.87
X-n655-k131	106780*	106803.30 (0.0218)	106780	68.71	106805.07 (0.0235)	106780	49.13	106781.94 (0.0018)	106780	45.80
X-n670-k130	146332	146905.48 (0.3919)	146445	73.07	146742.92 (0.2808)	146606	63.03	146639.28 (0.2100)	146423	98.48
X-n685-k75	68205	68298.62 (0.1373)	68246	70.04	68345.00 (0.2053)	68310	70.69	68281.58 (0.1123)	68227	97.30
X-n701-k44	81923	82143.14 (0.2687)	81954	82.05	82158.31 (0.2872)	82038	65.68	81981.40 (0.0713)	81923	101.54
X-n716-k35	43373	43474.50 (0.2340)	43380	88.43	43492.31 (0.2751)	43467	53.27	43397.96 (0.0575)	43369	99.92
X-n733-k159	136187	136354.38 (0.1229)	136257	87.13	136393.08 (0.1513)	136315	42.73	136270.20 (0.0611)	136190	101.67
X-n749-k98	77269	77496.82 (0.2948)	77382	94.80	77616.83 (0.4502)	77513	64.42	77387.72 (0.1536)	77321	111.73
X-n766-k71	114417	114688.16 (0.2370)	114493	99.56	114712.67 (0.2584)	114647	79.62	114457.30 (0.0352)	114425	121.56
X-n783-k48	72386	72715.06 (0.4546)	72509	111.15	72730.25 (0.4756)	72580	83.51	72462.08 (0.1051)	72382	110.22
X-n801-k40	73305	73626.22 (0.4382)	73435	79.97	73444.50 (0.1903)	73374	79.55	73356.02 (0.0696)	73306	100.75
X-n819-k171	158121	158349.54 (0.1445)	158208	118.80	158432.64 (0.1971)	158332	58.30	158215.00 (0.0594)	158143	123.45
X-n837-k142	193737	194059.30 (0.1664)	193906	91.09	194133.36 (0.2046)	193902	77.74	193807.54 (0.0364)	193735	118.31
X-n856-k95	88965*	89060.46 (0.1073)	88967	93.89	89018.27 (0.0599)	88983	70.09	89016.98 (0.0584)	88965	88.97
X-n876-k59	99299	99539.58 (0.2423)	99419	125.68	99624.45 (0.3278)	99484	111.50	99413.80 (0.1156)	99338	133.27
X-n895-k37	53860	54058.18 (0.3680)	53932	95.17	54097.80 (0.4415)	54053	100.79	53968.12 (0.2007)	53854	121.42
X-n916-k207	329179	329562.74 (0.1166)	329351	116.67	329782.40 (0.1833)	329653	115.97	329277.62 (0.0300)	329180	134.71
X-n936-k151	132715	133072.02 (0.2690)	132883	125.49	133326.80 (0.4610)	133198	88.83	132933.16 (0.1644)	132803	146.13
X-n957-k87	85465	85647.98 (0.2141)	85514	126.17	85526.10 (0.0715)	85512	89.54	85489.56 (0.0287)	85464	121.11
X-n979-k58	118976	120009.00 (0.8682)	119244	147.63	119195.11 (0.1842)	119130	101.92	118992.68 (0.0140)	118961	151.30
X-n1001-k43	72355	72737.48 (0.5286)	72558	124.45	72654.67 (0.4142)	72601	115.18	72469.70 (0.1585)	72382	155.29
Average		(0.0977)		38.19	(0.0881)		31.79	(0.0456)		42.96

average gap in all 10 instances. The same behavior is observed in the best solutions. This result

Table 7: Results achieved by AILS-II and AILS-PR in the instances introduced by [Arnold et al. \(2019\)](#).

Instance	n	AILS-PR (Máximo and Nascimento, 2021)				HGS (Vidal, 2022)			AILS-II		
		BKS	Avg (<i>gap</i>)	Best	T (min)	Avg (<i>gap</i>)	Best	T (min)	Avg (<i>gap</i>)	Best	T (min)
Antwerp1	6000	477277	481978 (0.9850)	481493	981.19	482543.6 (1.1035)	482115	998.99	477886.4 (0.1277)	477642	995.55
Antwerp2	7000	291371	297493.6 (2.1013)	296906	1158.54	300460.1 (3.1194)	300253	1152.18	292466.6 (0.3760)	292351	1153.86
Brussels1	15000	501767	511513.4 (1.9424)	510973	2490.47	518255.1 (3.2860)	518003	2483.12	502207.9 (0.0879)	502109	2493.06
Brussels2	16000	345551	357326.2 (3.4077)	354899	2664.81	361062.8 (4.4890)	361000	2637.30	347223.9 (0.4841)	346949	2653.04
Flanders1	20000	7240845	7359931 (1.6446)	7348763	3332.40	7448609 (2.8693)	7446950	3308.83	7244956.3 (0.0568)	7243669	3330.73
Flanders2	30000	4375193	4600045 (5.1392)	4574321	4992.01	4642602 (6.1119)	4641730	4651.51	4423072.8 (1.0943)	4416035	4992.51
Ghent1	10000	469532	476001.2 (1.3778)	475517	1652.78	480354.5 (2.3050)	480349	1658.97	470024.3 (0.1048)	469860	1659.58
Ghent2	11000	257802	263584 (2.2428)	262391	1830.27	268288.6 (4.0677)	268025	1661.51	259669.6 (0.7244)	259023	1811.13
Leuven1	3000	192848	194417.5 (0.8139)	194272	412.48	194430.6 (0.8206)	194408	497.57	193121.5 (0.1418)	193031	486.99
Leuven2	4000	111397	113117.9 (1.5448)	112885	653.91	113424.1 (1.8197)	113300	654.70	112143.8 (0.6704)	111962	648.04
Average			(2.1200)		2016.89	(2.9992)		1970.47	(0.3868)		2022.45

demonstrates the best performance of AILS-II in large instances in relation to the state of the art. AILS-PR showed a better average gap for the 10 instances compared to HGS. The average gap of this experiment was analyzed using the performance profiles of [Dolan and Moré \(2002\)](#) illustrated in Figure 6. This profile illustrates the percentage of problems solved by a given algorithm as a function of a performance factor $\tau \in \mathbb{R}$ that relates the analyzed algorithm with the best overall performance. Therefore, the coordinate axis $\phi_s(\tau) = P(r_{ps} \leq \tau \mid 1 < s < n_s)$ represents the accumulated probability that the performance ratio r_{ps} associated with the s algorithm is within a performance factor τ indicated on the x-axis. It is possible to observe that the AILS-II performance profile curve dominates those of the other metaheuristics, and that the AILS-PR curve dominates the HGS curve. When a curve dominates the others, it means that this algorithm is able to have a higher percentage of problems solved for any performance factor compared to the others.

5. Final Remarks and Future Work

AILS is an adaptive ILS metaheuristic introduced by [Máximo and Nascimento \(2021\)](#) to solve the CVRP. The authors proposed a hybridization of the path-relinking strategy with AILS, called AILS-PR, to better explore the solution space of such a problem. As a result, AILS-PR achieved outstanding results on the tested instances, which consisted mostly of small- and medium-size instances.

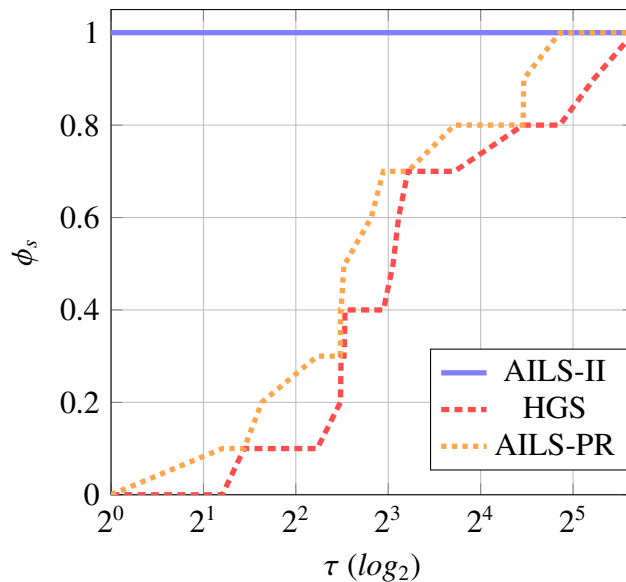


Figure 6: Performance profile (Dolan and Moré, 2002) comparing the average gaps obtained by the HGS, AILS-PR and AILS-II on Experiment II.

However, preliminary experiments pointed that neither AILS-PR nor AILS performed well on large-scale instances. On the one hand, AILS does not intensify the search much. On the other hand, AILS-PR was too slow on large instances.

Therefore, the main goal of this study was to present a more efficient approach to intensification for large instances of the CVRP. We thus introduced a two-phase version of AILS, called AILS-II. The first phase of the method roughly follows the steps of the original AILS, whereas the second phase provides restarts to prevent the method from getting stuck in local optima. We also present a study on diversity control that evaluates different approaches for controlling the perturbation degree and acceptance criteria. The results show that the adequate control of these two stages in the adaptive process is of great relevance to ensure good performance.

The computational experiments were performed using a benchmark dataset proposed by Uchoa et al. (2017) that contains 100 instances whose size ranges from 100 to 1000 vertices. We compared AILS-II with the AILS-PR and HGS. Considering this dataset, AILS-II presented a average solution better than or equal to the other algorithms on 71 of the 100 instances. The average gaps of AILS-II were 0.0456, whereas HGS and AILS-PR achieved average gaps of 0.0881 and 0.0977, respectively.

To evaluate the performance of AILS-II for larger instances, we carried out experiments with a dataset proposed by [Arnold et al. \(2019\)](#) that contains 10 large instances of CVRP. These instances have between 3000 and 30,000 vertices. The results of this experiment showed that AILS-II outperformed AILS-PR and HGS in all instances.

As future work, we intend to continue investigating more efficient mechanisms for controlling diversity. We also want to explore other search strategies that are robust and simpler. Other areas of interest involve machine learning to optimize the performance of metaheuristics.

Acknowledgments

The authors are grateful for the financial support provided by CNPq and FAPESP (2016/01860-1, 2019/22067-6); the Laboratory for Simulation and High Performance Computing (LaSCADo), funded by (FAPESP) through project 2010/50646-6; and the Center for Mathematical Sciences Applied to Industry (CeMEAI), funded by FAPESP by through the 2013/07375-0 project, for the availability of computer resources.

References

- Arnold, F., Gendreau, M., Sörensen, K., 2019. Efficiently solving very large-scale routing problems. *Computers & Operations Research* 107, 32–42. doi:[10.1016/j.cor.2019.03.006](https://doi.org/10.1016/j.cor.2019.03.006).
- den Besten, M., Stützle, T., Dorigo, M., 2001. Design of iterated local search algorithms, in: Boers, E.J.W. (Ed.), *Applications of Evolutionary Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 441–451.
- Christiaens, J., Vanden Berghe, G., 2020. Slack induction by string removals for vehicle routing problems. *Transportation Science* 54, 417–433.
- Christofides, N., Mingozzi, A., Toth, P., 1979. The vehicle routing problem, in: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), *Combinatorial Optimization*, Wiley, Chichester. pp. 315–338.
- Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. *Management Science* 6, 80–91.

- Dolan, E.D., Moré, J.J., 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming* 91, 201–213.
- Dueck, G., Scheuer, T., 1990. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics* 90, 161–175. doi:[10.1016/0021-9991\(90\)90201-B](https://doi.org/10.1016/0021-9991(90)90201-B).
- Golden, B.L., Wasil, E.A., Kelly, J.P., Chao, I.M., 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results, in: *Fleet Management and Logistics*. Springer, pp. 33–56.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Lenstra, J.K., Rinnooy Kan, A.H.G., 1981. Complexity of vehicle routing and scheduling problems. *Networks* 11, 221–227.
- Lourenço, H.R., Martin, O.C., Stützle, T., 2003. Iterated local search, in: *Handbook of Metaheuristics*. Springer US, Boston, MA. volume 57 of *International Series in Operations Research and Management Science*, pp. 320–353.
- Lourenço, H.R., Martin, O.C., Stützle, T., 2010. *Iterated Local Search: Framework and Applications*. Springer US, Boston, MA. pp. 363–397. doi:[10.1007/978-1-4419-1665-5_12](https://doi.org/10.1007/978-1-4419-1665-5_12).
- Máximo, V.R., Cordeau, J.F., Nascimento, M.C.V., 2021. An adaptive iterated local search heuristic for the heterogeneous fleet vehicle routing problem. arXiv preprint doi:[10.48550/ARXIV.2111.12821](https://doi.org/10.48550/ARXIV.2111.12821).
- Máximo, V.R., Nascimento, M.C.V., 2021. A hybrid adaptive iterated local search with diversification control to the capacitated vehicle routing problem. *European Journal of Operational Research* 294, 1108–1119. doi:[10.1016/j.ejor.2021.02.024](https://doi.org/10.1016/j.ejor.2021.02.024).
- Morais, V.W., Mateus, G.R., Noronha, T.F., 2014. Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications* 41, 7495–7506.

- Penna, P.H.V., Subramanian, A., Ochi, L.S., 2013. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics* 19, 201–232.
- Stützle, T., 2006. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research* 174, 1519–1539. doi:[10.1016/j.ejor.2005.01.066](https://doi.org/10.1016/j.ejor.2005.01.066).
- Tarantilis, C., Kiranoudis, C., Vassiliadis, V., 2004. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research* 152, 148–158. doi:[10.1016/S0377-2217\(02\)00669-0](https://doi.org/10.1016/S0377-2217(02)00669-0).
- Toth, P., Vigo, D., 2001. An overview of vehicle routing problems, in: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. pp. 1–26. doi:[10.1137/1.9780898718515.ch1](https://doi.org/10.1137/1.9780898718515.ch1).
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A., 2017. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research* 257, 845–858.
- Vidal, T., 2022. Hybrid genetic search for the CVRP: Open-source implementation and swap* neighborhood. *Computers & Operations Research* 140, 105643. doi:[10.1016/j.cor.2021.105643](https://doi.org/10.1016/j.cor.2021.105643).