

RESEARCH ARTICLE

Hybrid Algorithms for Energy Minimizing Vehicle Routing Problem: Integrating Clusterization and Ant Colony Optimization

NICOLÁS FRÍAS^{ID}, FRANKLIN JOHNSON^{ID}, AND CARLOS VALLE^{ID}

Departamento de Ciencia de Datos e Informática, Facultad de Ingeniería, Universidad de Playa Ancha, Valparaíso 2360002, Chile

Corresponding author: Nicolás Frías (nicolas.friasrojas@gmail.com)

This work was supported by Universidad de Playa Ancha under Grant DGI APFON 03-22.

ABSTRACT In the field of engineering, complex problems often arise that require solutions. The implementation of these algorithms plays a crucial role in achieving favorable outcomes with the available resources. The Vehicle Routing Problem (VRP) has been a central topic in distribution and logistics for decades. New VRP models and tools are developed to address the challenges of modern logistics. The Energy Minimizing Vehicle Routing Problem (EMVRP) is a “green”-oriented variant of the VRP where the objective is to minimize the total amount of energy consumed by a fleet of vehicles. The VRP literature has focused on solving the problem using a variety of approaches and techniques, including exact methods, heuristics, metaheuristics, and hybrid algorithms. Hybrid algorithms combine different techniques to obtain more effective and better solutions. This work presents four innovative hybrid algorithms to address the EMVRP problem. These algorithms combine Machine Learning (ML) clustering techniques with metaheuristic approaches inspired by an Ant Colony Optimization (ACO). The proposed algorithms are: Free Ant + K-Means, Free Ant + K-Medoids, Restricted Ant + K-Means, and Restricted Ant + K-Medoids. Each of them combines the benefits of clustering with the optimization capacity of ACO. Proposed algorithms were subjected to testing using instances from CVRPLIB. Both Free Ant and Restricted Ant efficiently solved EMVRP problems. The results obtained were analyzed and compared with the proposals of other authors in the literature. Overall, the results are promising, but they also indicate a significant scope for experimentation and parameter tuning of the proposed algorithms.

INDEX TERMS Complex problems, free ant, green-vrp, heuristics, k-means, k-medoids, machine learning, metaheuristics, restricted ant, vehicle routing problem (VRP).

I. INTRODUCTION

One of the most widely known and oldest combinatorial optimization problem is the Traveling Salesman Problem (TSP), which involves determining the shortest possible route that visits all cities (from a city list) exactly once and returns to the starting city. Over time, the idea of optimizing routes for a finite number of cities has been extended to find optimal routes for vehicles from one or more depots to a set of locations or customers, thus being understood as the natural

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan^{ID}.

evolution of the TSP. Since its inception, this problem has generated significant interest among the Operations Research and mathematical communities. These problems are known as Vehicle Routing Problems (VRP) and belong to the field of combinatorial optimization.

Introduced in the early 1960s, the VRP has been extensively studied by the scientific community, primarily because of its practical applications in the field of distribution and logistics. Additionally, the VRP presents inherent complications that exist in this type of combinatorial optimization problem, similar to the TSP, making it part of the NP-Hard problem set. Whereas the TSP seeks the shortest route to visit

a certain number of cities and return to the origin, the VRP aims to determine the optimal set of routes for vehicles, visit all customers once, and serve their associated demands per vehicle. In the VRP, there is a finite number of routes, one for each available vehicle, and the goal is to determine the optimal set of routes. This characteristic makes the VRP a generalization of TSP.

The formal definition of a VRP states that m vehicles initially located at a depot must serve positive amounts of goods to n customers. The objective is to determine the K optimal routes used by the fleet of vehicles to fulfill the demands of the n customers. VRP have many practical applications, especially in transportation and distribution logistics. Consequently, various VRP variants have been developed to better represent our reality and involve variables that approximate the actual costs associated with traversing different vehicle routes. Models such as the Capacitated Vehicle Routing Problem (CVRP), Dynamic Vehicle Routing Problem (DVRP), and Vehicle Routing Problem with Time Windows (VRPTW) are only a few current frameworks within this problem class, and they can differ significantly from one another or involve constraints from other types of problems.

This article presents the results of a research project focused on the Energy Minimizing Vehicle Routing Problem (EMVRP), which is one of the most interesting models in the field of vehicle routing optimization. This model stands out for its ability to minimize both the financial costs and environmental impact of a fleet of vehicles while traversing different routes. Therefore, the EMVRP has emerged as a valuable tool for addressing logistical and environmental challenges in route planning and distribution. Originally published by Kara et al. [1], the EMVRP is a key problem within the Green VRP domain. In this study, we propose an algorithm to solve EMVRP problems using constructive metaheuristic techniques and clustering algorithms. The study is divided into three parts: the EMVRP problem, clustering algorithms, and the use of the Ant Colony Optimization (ACO) metaheuristic technique. Our hypothesis in this work is that we believe that mixed strategies combining metaheuristics and machine learning propose better solutions than these techniques separately. We present two innovative metaheuristic algorithms for problem resolution based on the Best-Worst Ant System algorithm proposed by Cordón et al. [2]. These algorithms, named Free Ant and Restricted Ant, are combined with clustering techniques, specifically the K-Means and K-Medoids algorithms with capacity restriction, inspired by the work of Geetha et al. [3].

II. ENERGY MINIMIZING VEHICLE ROUTING PROBLEM

Let us consider an arbitrary Capacitated Vehicle Routing Problem (CVRP). This problem is defined by a graph $G = (V, A)$, where $V = v_1, v_2, \dots, v_i$ is the set of nodes representing the customers in the problem, with v_0 is the depot, and v_i is the i -th node or vertex. These nodes have positions with coordinates represented as $c_i = (x_i, y_i)$ in a

Cartesian plane. $A = (i, j) : i, j \in V, i \neq j$ is the set of arcs or edges connecting each node in set V , where (i, j) represents the arc connecting node v_i to node v_j .

Additionally, we have the following characteristics that complete the EMVRP model:

- d_{ij} is the distance between node i and node j ,
- q_i is the positive integer load of node v_i , i.e., its demand or supply,
- m is the number of heterogeneous vehicles,
- Q_0 is the tare weight (weight of the vehicle when empty) of these vehicles,
- Q is the maximum capacity of a vehicle.

Kara et al. [1] defined the Energy Minimizing Vehicle Routing Problem (EMVRP) as a vehicle routing problem, such that:

- Each node v_i is served by exactly one vehicle,
- Each route starts and ends at the depot v_0 ,
- The load on the arcs accumulates as the sum of quantities q_i supplied by the preceding nodes (in the case of pickup) and decreases as the sum of quantities q_i demanded by the preceding nodes (in the case of delivery),
- The load of a vehicle must not exceed the capacity Q ,
- The objective is to find a set of K vehicle routes with the minimum total cost, that is, the minimum total energy.

Furthermore, the following decision variables are defined for the formulation of this problem:

- 1) $X_{ij} = 1$ if the arc (i, j) is part of a route, and 0 otherwise,
- 2) Y_{ij} is the weight of the vehicle when traveling from node v_i to node v_j , and 0 otherwise.

As can be inferred, Y_{ij} is the component that gives meaning to this new energy-oriented approach. The weight of the vehicle on the first arc of any route must have a predetermined value; for example, tare weight Q_0 , and it should always increase (or decrease) by units of q_i immediately after visiting node v_i . In the case of pickup, the flow of the variable exhibits an incremental step function, whereas in the delivery case, it exhibits a decremental step function. Therefore, a model constructed for one case may not be suitable for its counterpart. However, Kara et al. demonstrated that when distances are symmetric, the optimal route for the delivery (or pickup) case is equal to the optimal route for the pickup (or delivery) case, but traversed in the reverse order. Thus, if the distances are symmetric, there is no need to differentiate between delivery and pickup, because the solution for one case is the solution for the other, but in the reverse order. Therefore, the objective function to be minimized is as follows:

$$\min \sum_{i=0}^n \sum_{j=0}^n d_{ij} Y_{ij}. \quad (1)$$

This objective function (1) is subject to the following constraints:

$$\sum_{i=1}^n X_{0i} = m. \quad (2)$$

$$\sum_{i=1}^n X_{i0} = m. \quad (3)$$

$$\sum_{i=0}^n X_{ij} = 1, \quad j = 1, 2, 3, \dots, n. \quad (4)$$

$$\sum_{j=0}^n X_{ij} = 1, \quad i = 1, 2, 3, \dots, n. \quad (5)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^n Y_{ij} - \sum_{\substack{j=0 \\ j \neq i}}^n Y_{ji} = q_i, \quad i = 1, 2, 3, \dots, n. \quad (6)$$

$$Y_{0i} = Q_0 X_{0i}, \quad i = 1, 2, 3, \dots, n. \quad (7)$$

$$Y_{ij} \leq (Q + Q_0 - q_j) X_{ij}, \quad (i, j) \in A. \quad (8)$$

$$Y_{ij} \geq (Q_0 + q_i) X_{ij}, \quad \forall (i, j) \in A. \quad (9)$$

$$X_{ij} = 0 \text{ o } 1, \quad (i, j) \in A. \quad (10)$$

The cost of traversing an arc (i, j) is the product of the distance d_{ij} (the distance between nodes v_i and v_j) and the weight Y_{ij} carried by the vehicle on this arc, which is satisfied by the objective function given in (1). Furthermore, Constraints (2) and (3) ensure that all m vehicles are used. Constraints (4) and (5) represent the degrees of restriction for each node. Constraint (6) represents the classical conservation of the flow equation, which balances the inflow and outflow of each node (in terms of demand or supply) and prohibits any illegal sub-trips. Constraint (7) initializes the flow in the first arc of each route. Constraint (8) handles the capacity constraints and forces the value of Y_{ij} to be zero when arc (i, j) is not a part of any route. Constraint (9) sets the lower bounds for flow on any arc. The integrity constraint is given in (10), and corresponds to the binary variable for each arc.

III. BACKGROUND

The state-of-the-art analysis in this work focused on addressing contemporary challenges in logistics and distribution, using the Energy Minimizing Vehicle Routing Problem (EMVRP) as a case study. These logistical problems encompass a wide array of challenges and complexities. We believe that the application of heuristics, metaheuristics and hybrid algorithms, such as ant colony optimization and clustering, has the potential to provide optimal or near-optimal solutions for logistics and distribution-related issues. The combination of these techniques can be beneficial in approaching these problems from various perspectives:

- 1) Optimization: Metaheuristics can find optimal or near-optimal solutions for logistics and distribution problems.
- 2) Adaptability: Metaheuristics can adapt to changing situations and cater to different types of problems, depending on current needs.
- 3) Complex Problems: Hybrid algorithms combine different techniques, such as machine learning and optimization, to effectively address challenging problems.

Before delving into the development of the algorithms proposed in this study, it is essential to introduce the terms that will be frequently used throughout the following chapters. Therefore, this section presents some fundamental concepts necessary to comprehend the processes involved in the algorithms proposed in this study. By doing so, readers will be able to gain a clear understanding of the terms employed in the analysis of Free Ant and Restricted Ant.

A. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) acts as a tool for solving complex combinatorial optimization problems and was originally designed to find the optimal path within a graph, specifically a Traveling Salesman Problem (TSP) graph. ACO can be understood as a bio-inspired metaheuristic with a general-purpose application, arising from the genuine observation that a colony of ants is capable of finding an optimal path between their nest and a food source. Initially, real ants, explore the area around their nest in a random manner when searching for food, leaving behind a trail of pheromones. This pheromone trail affects subsequent ants, gradually causing them to converge towards an optimal path.

The foundation of the Ant Colony Optimization theory was introduced by Marco Dorigo in his doctoral thesis in 1992. However, it was not until the mid-1990s that Dorigo, M., Maniezzo, V., and Colomi, A. [4] published it as a novel metaheuristic called Ant System (AS). AS employs a fixed number of artificial ants (w) where each ant a (considered a computational agent) constructs a solution to the problem, previously represented by a graph. In this graph, each arc (i, j) corresponds to one of the possible choices available to the ant to transition from its current state to a new state.

Ant a is only aware of the information associated with these arcs (i, j) , which consists of the following:

- 1) Heuristic information η , which determines the heuristic preference of an ant to move from one state to another. This information typically refers to the cost of transitioning from node v_i to node v_j , and in practice, it remains unchanged throughout the execution of the algorithm.
- 2) Pheromone trail information τ , which measures the learned preference for state transitions among the w ants. This information aims to mimic the real pheromone deposited by natural ants and is a data value that will be modified during the course of the execution of the algorithm, with the purpose of aiding future ants in selecting better routes.

The development of a generic ACO algorithm requires artificial ant a to have memory M_a , where it stores each visited node v_i . This list is commonly known as the taboo list because it contains all nodes that cannot be revisited (as they are part of a route). The list of the remaining nodes that the ant can visit is denoted as J_a . To determine the next node v_j that the ant will visit, the transition rule (11), also known as the standard probability distribution p_a of AS, is used to

determine the next node v_j that the ant will visit.

$$p_a(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha [\eta(r, s)^\beta]}{\sum_{u \in J_a(r)} [\tau(r, u)]^\alpha [\eta(r, u)]^\beta}, & \text{if } s \in J_a(r), \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where r corresponds to node v_i and s corresponds to node v_j . τ represents the pheromone trail on arc (i, j) , and η denotes the heuristic information on the same arc, typically the inverse of the distance $d(r, s)$ ($\frac{1}{d(r, s)}$). $J_a(r)$ represents the set of nodes that are yet to be visited by ant a starting from node r . Furthermore, α and β are adjustable parameters that determine the relative importance of the pheromone trail versus the relative importance of the heuristic information, respectively.

In ACO, it is necessary to modify the pheromone trails associated with arcs in each iteration. For example, in AS, at the end of each iteration, there is an evaporation of the pheromone trails, followed by the deposition of a determined number of pheromones by the w ants. The update of the pheromone τ_{rs} , associated with the arc (i, j) connecting nodes v_i and v_j , occurs as follows:

$$\tau_{rs} \leftarrow (1 - \rho) \cdot \tau_{rs} + \sum_{a=1}^w \Delta \tau_{rs}^a \quad (12)$$

where ρ is the evaporation coefficient (a predefined value within the range $[0, 1]$), and $\Delta \tau_{rs}^a$ is the amount of pheromone deposited on arc (i, j) by ant a . The amount of pheromone $\Delta \tau_{rs}^a$ that ant a can deposit can be calculated as follows:

$$\Delta \tau_{rs}^a = \begin{cases} \frac{1}{L_a} & \text{if ant } a \text{ uses arc } (i, j) \text{ in its solution,} \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where L_a is the length of the solution constructed by ant a .

Since the introduction of AS, various variants of ACO algorithms have been proposed to solve combinatorial optimization problems. Some are more closely inspired by AS, whereas others diverge further. However, their main ideas are similar:

- 1) The use of a colony of “ants” that cooperate with each other,
- 2) Artificial “pheromone” trails for local communication,
- 3) A sequence of local movements, and
- 4) A stochastic decision policy that utilizes local information.

Some well-known ACO algorithms include the Ant Colony System (ACS) [5], AS_{rank} [6], and MAX-MIN Ant System (MMAS) [7], among others. For more details on Ant Colony Optimization, please refer to [8].

B. MAX-MIN ANT SYSTEM

As presented by Stützle, T. & Hoos, H. [7], the MAX-MIN Ant System (MMAS) differs from standard Ant Colony Optimization (ACO) algorithms in that it limits the amount of pheromone that can be deposited on an arc (i, j) . This is

achieved by setting a minimum value τ_{min} and a maximum value τ_{max} of pheromone for each arc and updating the pheromone value only if it falls within this range. This restriction ensures that the algorithm does not converge rapidly to suboptimal solutions. The determination of τ_{min} and τ_{max} in MMAS are given by:

$$\tau_{max} = \frac{1}{1 - \rho} \frac{1}{f(S^{gb})} \quad (14)$$

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[n]{P_{best}})}{(avg - 1)\sqrt[n]{P_{best}}} \quad (15)$$

Here, $f(S^{gb})$ represents the value of the best global solution of the problem, P_{best} is the probability of convergence to the best-found solution, which is typically assumed to be 0.05. Finally, avg denotes the average choice among the different components of the solution, i.e., $avg = n/2$.

In general, the MAX-MIN Ant System is an effective and widely used optimization algorithm that has been successfully applied to a broad range of problems, including the Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), and Vehicle Routing Problem (VRP).

C. BEST-WORST ANT SYSTEM

The Best-Worst Ant System (BWAS) algorithm is an ACO type algorithm developed by Cordón et al. [2], which incorporates concepts from evolutionary computation. This algorithm utilizes the AS transition rule (11) and does not require online pheromone updates (12), thus making it an optional step. Furthermore, this algorithm relies on three essential actions performed by an external program separate from the ant colony. This program is a special type of non-interactive computational process known as a “daemon.” We briefly outline these actions as follows:

- 1) Only the best global solution, referred to as “best-global,” and the current worst solution, referred to as “worst-current,” are considered for performing positive and negative pheromone updates, respectively (offline update). This update is applied once the w ants have constructed their solutions and before the iteration end.
 - a) The offline update is applied as follows:

$$\tau_{rs} \leftarrow (1 - \rho) \cdot \tau_{rs} + \Delta \tau_{rs} \quad (16)$$

$$\begin{cases} \Delta \tau_{rs} = f(C(S_{best-global})), \\ \quad \text{if } (r, s) \in S_{best-global}, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

where $f(C(\cdot))$ corresponds to a function that evaluates the quality of any given solution.

- b) On the other hand, the penalty of arcs in the current worst solution, “worst-current,” is performed as follows:

$$\forall (r, s) \in S_{worst-current} \text{ and } (r, s) \notin S_{best-global}, \quad \tau_{rs} \leftarrow (1 - \rho) \cdot \tau_{rs} \quad (18)$$

- 2) The daemon process also incorporates a pheromone matrix reset procedure. This occurs whenever a defined stagnation condition is satisfied.
- 3) The pheromone matrix underwent a series of stochastic mutations that were loosely inspired by evolutionary computations. The purpose of these mutations was to introduce greater diversity into the search process:
 - a) Each row of the pheromone matrix is mutated (with a probability of P_m) as follows:

$$\tau'_{rs} = \begin{cases} \tau_{rs} + mut(it, \tau_{threshold}), & \text{if } e = 1 \\ \tau_{rs} - mut(it, \tau_{threshold}), & \text{if } e = 0 \end{cases} \quad (19)$$

$$\tau_{threshold} = \frac{\sum_{(r,s) \in S_{best-global}} \tau_{rs}}{|S_{best-global}|} \quad (20)$$

where e is a random value in $\{0, 1\}$, it represents the current iteration, and $\tau_{threshold}$ is the average pheromone value of the best global solution.

- b) Furthermore, the function $mut()$ is described as follows:

$$mut(it, \tau_{threshold}) = \frac{it - it_r}{N_{it} - it_r} \cdot \sigma \cdot \tau_{threshold} \quad (21)$$

where N_{it} is the maximum number of iterations, it_r is the last iteration where a pheromone matrix reset is performed, and σ is a constant that specifies the intensity of the mutation relative to the number of iterations conducted so far.

D. K-MEANS

One of the most commonly used algorithms for clustering datasets is K-Means, which is known for its simplicity and effectiveness. This algorithm (also known as Lloyd's Algorithm) was originally designed by Lloyd, S. P. [9], although the term "K-Means" was coined in 1967 by MacQueen, J. B. [10]. K-Means belongs to the field of unsupervised machine learning and takes a set of elements and assigns each of them to one of the K groups. This is based on the inherent features of each element, that is, their values x_j . The goal of the algorithm is to find the optimal K centroids that minimize the sum of the squared distances between all the objects in a group and their centroid.

In the classical sense of this algorithm, the centroids are initially placed randomly in space, and the algorithm iteratively assigns each element v_i to one of the K groups based on its similarity or proximity to the centroids. Each element is assigned to group G with the smallest distance to its respective centroid, where $G = G_1, G_2, G_3, \dots, G_K$ represents the set of groups. The equation for minimizing the sum of squared distances across all groups G_i , is given by:

$$argmin \sum_{i=1}^K \sum_{c_j \in G_i}^n \|c_j - \mu_i\|^2 \quad (22)$$

where n corresponds to the number of elements in group G_i , c_j represents the Cartesian coordinates of the j -th element (i.e., the values x_i), and μ_i is the average of the Cartesian coordinates among the nodes in group G_i , representing the coordinates of the i -th centroid belonging to that group. The K centroids (1, 2, 3, ..., K) are then recalculated as the average of the coordinates of all nodes within each respective group in G . Because it is an iterative algorithm, the groups are redefined as the K centroids are recalculated, and the nodes are reassigned to the groups organically after each iteration until convergence is reached.

E. K-MEDOIDS

Another algorithm in the field of unsupervised machine learning is the Partition Around Medoids (PAM), originally proposed by Kaufman, L., & Rousseeuw, P. J. [11], and is commonly referred to as "K-Medoids." Similar to K-Means, this algorithm works by dividing the dataset into K groups. The idea behind K-Medoids arises, in part, from the fact that K-Means is often sensitive to outliers. This sensitivity occurs because the centroids, denoted as i , can be strongly influenced by extreme values (elements whose coordinates are significantly distant from the rest), because the centroids are calculated as the average coordinates of the elements. Thus, K-Medoids is a classification technique that segments the elements of a dataset; however, unlike K-Means, where the centroids are not necessarily part of the groups, in K-Medoids, an object from the dataset is selected as the centroid of its group. This central object is referred to as a medoid.

The working mechanism of K-Medoids follows an iterative process similar to that of K-Means. In the initial step, the medoids of each group are initialized randomly. Each element is then associated with one of the groups in G based on the closest proximity to its respective medoid M_i . In each subsequent iteration, the K medoids ($M_1, M_2, M_3, \dots, M_K$) are recalculated in terms of the similarity among the elements within the group. In other words, the new medoid M_i of the i -th group is determined as the object that, on average, has the lowest distance to the rest of the objects within the same group. The cost of group G_i in K-Medoids is given by:

$$cost_{G_i} = \sum_{c_j \in G_i} |c_j - M_i| \quad (23)$$

where G represents the set of groups, n is the total number of non-medoid elements in a group, M_i denotes the coordinates of the medoid belonging to group G_i , and c_j represents the coordinates of the j -th non-medoid element within group G_i .

F. VARIABLE NEIGHBORHOOD SEARCH

Variable Neighborhood Search (VNS) is commonly described as a framework rather than a specific local search algorithm, as it provides guidelines for implementing more effective local searches. The VNS was originally introduced by Hansen, P. & Mladenovic, N. [12] in 1997 as a method for constructing local search heuristics. It is based on systematic changes of neighborhoods, involving a descent phase to reach

a local minimum and a shaking phase to escape the current valley, that is, to search for a better local minimum.

The VNS is considered a metaheuristic that exploits the idea of changing neighborhoods to find improved solutions. To achieve this, the algorithm relies on the following three observations:

- 1) A local minimum with respect to one neighborhood structure may not be the same for another.
- 2) A global minimum is a local minimum with respect to all possible neighborhood structures.
- 3) For many problems, the local minimum with respect to one or more neighborhoods is relatively close to each other.

This empirical observation implies that a local optimum often provides some information about the global optimum.

G. CLARKE AND WRIGHT'S ALGORITHM

The Clarke & Wright algorithm (also known as Saving's Algorithm) was introduced by Clarke, G. & Wright, J.W. [13] in 1964. It is one of the most popular heuristic techniques for solving the Vehicle Routing Problem (VRP). The central idea of this algorithm is to combine two routes into a single new route, ensuring that "savings" are achieved. The potential savings from combining two routes is given by:

$$g_{ij} = d_{i0} + d_{0j} - d_{ij}. \quad (24)$$

Here, g_{ij} represents the savings value when two nodes are integrated into the same route (greater values indicate higher savings); d_{i0} is the distance between node v_i and the depot v_0 , d_{0j} is the distance between depot v_0 and node v_j , and d_{ij} is the distance between node v_i and node v_j .

The algorithm operates as follows: first, each customer is assigned to a vehicle. Then, it attempts to combine two routes, provided that the vehicle's capacity allows it. The order in which the routes are combined is determined using (24). These savings values are sorted in descending order, and the route that generates the highest savings is selected and added to the merged route. The Clarke & Wright algorithm has two approaches: Parallel and Sequential. The Parallel approach creates routes simultaneously, whereas the Sequential approach creates one route at a time.

IV. METHODOLOGY

This section presents the working methodology and the processes involved in each of the proposed algorithms: Free Ant and Restricted Ant. Furthermore, the notation k is used interchangeably to refer to both centroids and medoids M from this point onward.

The overall behavior of the proposed algorithms in this work can be observed in Fig. 1, which presents, in a flowchart, the clustering stage and then the ant colony optimization stage.

A. CLUSTERING ALGORITHMS

The purpose of these algorithms is to obtain information that can assist the ants in making better decisions during

state transitions. These clustering algorithms are used to generate groups of nodes that can potentially form a single route. Therefore, the clustering algorithms are executed at the beginning of the main program execution, once the data for an instance has been loaded, various parameters that control the algorithms have been set, and the distances between all nodes, as well as the energy cost C_{ij} between each node, have been calculated.

1) IMPROVED K-MEANS ALGORITHM FOR CAPACITATED CLUSTERING PROBLEM

The K-Means algorithm typically does not consider any constraints when generating clusters because it focuses solely on identifying similar elements. To maintain the vehicle capacity constraint within each route, an improved version of K-Means called the Improved K-Means Algorithm for Capacitated Clustering Problem, has been utilized. This algorithm was proposed by Geetha et al. [3] and its main advantage lies in considering the maximum capacity Q of the m vehicles in a Capacitated Vehicle Routing Problem (CVRP). To avoid breaking this capacity constraint, nodes are grouped using a prioritization system.

The Improved K-Means Algorithm incorporates several modifications to better handle situations where there is an intra-cluster maximum capacity constraint. The customer nodes are assigned to the nearest groups in a similar manner to any K-Means algorithm, but the assignment occurs in an ordered manner based on a characteristic of each customer: their demand. Customers with higher demands q_i and minimum distances d_{ik_j} with respect to the centroids are given higher priority when being assigned to a group. Therefore, it is necessary to sort the customers (based on their respective demands) at the beginning of the algorithm in descending order.

Once the nodes are sorted, they are assigned to the groups whose centroid are closest to them. To determine the initial centroid locations, the coordinates of the nodes with higher demand that are farthest from the depot and the already selected centroids are considered. This approach is based on the work of Arthur, D. and Vassilvitskii, S. [14], who demonstrated that the initial position of the centroids directly affects the final result of the clustering process. They refer to this improved version of the K-Means algorithm as K-Means++.

To determine the optimal number of routes, that is, the optimal K , the following equation is used:

$$K = \sum_{i=1}^n \frac{q_i}{Q}. \quad (25)$$

On the other hand, to calculate the priority of each node v_i , the next equation is used:

$$\text{Priority } P_i = d_{ik_j}/q_i. \quad (26)$$

Here, d_{ik_j} corresponds to the distance between node v_i and the k -th centroid of each j -th group. Once all nodes have been



FIGURE 1. Generalized flowchart of the proposed hybrid algorithms.

assigned to a group, the coordinates of each centroid k_j is recalculated. If convergence is not achieved, the priority P_i needs to be recalculated for a new iteration.

In the context of VRP problems, the optimal number of clusters that clustering algorithms should generate will always be equal to the number of routes, that is, the optimal K . Therefore, the number of clusters is determined by (25).

2) IMPROVED K-MEDOIDS ALGORITHM FOR CAPACITATED CLUSTERING PROBLEM

The K-Medoids algorithm for Capacitated Clustering follows the same principles as the algorithm proposed by Geetha et al. [3], but it is governed by the criteria of K-Medoids. Once again, we employ the idea of K-Means++ to select the most distant nodes as the initial medoids.

Each node is assigned to its nearest group, that is, the group whose medoid is closest in terms of distance. Similar to the Capacitated K-Means approach, a priority system based on distance and demand was used for node assignment. Once all nodes have been assigned to a group, the K medoids must be recalculated. To choose the new medoid for each group in G_j , it is necessary to determine which node in the group has the lowest average cost compared with the other members of the same group, according to (23).

B. ACO ALGORITHMS

The ACO Algorithms developed in this work are Free Ant and Restricted Ant, and both are based on the BWAS system. The execution of these algorithms begins after the clustering process.

Algorithm 1 Clustering Algorithms

```

1: Initialization of variables
2:  $K$ 
3:  $totalCost$ 
4: function GetFirstCentroids( $K$ )
5:   for  $idx_k$  in  $K$  do
6:     if  $centroidsList$  is empty then
7:       Find the farthest node from the depot
8:        $centroidsList \leftarrow node$ 
9:     else
10:      Find the farthest node from the rest of
centroids
11:       $centroidsList \leftarrow node$ 
12:    end if
13:  end for
14:  return  $centroidsList$ 
15: end function
16: while  $actualIteration < maxIterations$  do
17:    $unassignedNodes$ 
18:   Calculate distances from every node to every
centroid
19:   for  $node$  in  $nodesList$  do
20:     if  $node_r$  exists in  $unassignedNodes$  then
21:       Gets closest clusters to  $node_r$ 
22:       for  $cluster$  in  $closestClusters$  do
23:         Calculate distances from every node to
 $cluster$ 
24:         Calculate priority for every node to
 $cluster$ 
25:         Assign to this  $cluster$  the node with
highest priority
26:         Remove this node from  $unassignedNodes$ 
27:         if  $node_r$  not exists in  $unassignedNodes$ 
then
28:           Exit this FOR loop
29:         end if
30:       end for
31:     end if
32:   end for
33:   function CalculateTotalCost
34:     Sum all distances from every node to its centroid
35:     return  $newCost$ 
36:   end function
37:   if  $newCost < totalCost$  then
38:      $bestClusters \leftarrow actualClusters$ 
39:      $bestCentroids \leftarrow actualCentroids$ 
40:      $totalCost \leftarrow newCost$ 
41:   end if
42:   Recalculate centroids
43:   if  $actualCentroids = bestCentroids$  then
44:     Exit this WHILE loop
45:   end if
46: end while

```

The pseudo-code for the ACO Algorithms is outlined in Algorithm 2.

Algorithm 2 ACO Algorithms

```

1: Initialization of variables
2:  $unvisitedNodes$ 
3:  $routes$ 
4: while length of  $unvisitedNodes > 0$  do
5:    $newRoute$ 
6:    $actualNode \leftarrow depot$ 
7:    $maxCapacity$ 
8:    $vehicleWeight \leftarrow tare$ 
9:   Calculate all the  $validNodes$  from  $unvisitedNodes$ 
10:  while length of  $validNodes > 0$  do
11:    Calculate the  $heuristics$  from  $actualNode$  to
 $validNodes$ 
12:    Calculate the  $pheromones$  from  $actualNode$  to
 $validNodes$ 
13:     $pseudoRandomChoice$ 
14:    if  $pseudoRandomChoice \leq l_0$  then
15:      Calculate  $bestNode$ 
16:       $nextNode \leftarrow bestNode$ 
17:    else
18:      Calculate all the probabilities with  $heuristics$ 
and  $pheromones$ 
19:      Choose a  $randomNode$  based on these
probabilities
20:       $nextNode \leftarrow randomNode$ 
21:    end if
22:    Append  $nextNode$  to  $newRoute$ 
23:     $maxCapacity$  subtract  $demand$  of  $nextNode$ 
24:     $vehicleWeight$  sum  $demand$  of  $nextNode$ 
25:     $unvisitedNodes$  remove  $nextNode$ 
26:     $actualNode \leftarrow nextNode$ 
27:    Recalculate all the  $validNodes$  from
 $unvisitedNodes$ 
28:  end while
29:  Append  $newRoute$  to  $routes$ 
30: end while
31: return  $routes$ 

```

The algorithm begins by initializing the variables and setting up the necessary data structures. The main loop iterates until all the nodes have been visited. In each iteration, a new route is constructed. The algorithm selects the next node to visit based on the heuristics and pheromone information. The selection is influenced by a pseudo-random choice using parameter l_0 . If the pseudo-random choice is less than or equal to l_0 , then the best node is chosen. Otherwise, the selection is based on the probabilities calculated from the heuristics and pheromones. The selected node is added to the new route and the algorithm updates the remaining capacity and weight of the vehicle. The selected node is removed from the list of unvisited nodes, and the process continues until all the nodes have been visited. The resulting routes are returned as the outputs of the algorithm.

1) FREE ANT

The Free Ant algorithm utilizes information obtained from the best K clusters to determine the nodes belonging to each

group. In addition, this information is used to construct the initial pheromone matrix. For this purpose, an amount of pheromone τ_{max} is assigned to each arc (i, j) connecting nodes within the same group, whereas arcs (i, j) connecting nodes from different groups are assigned an initial amount of pheromone τ_0 . The calculation of τ_0 is based on the equation developed by M. Dorigo and L.M. Gambardella in their work on the Ant Colony System [5]:

$$\tau_0 = \frac{1}{L_c}. \quad (27)$$

Here, L_c corresponds to the best total energy obtained by a greedy algorithm.

To prevent possible stagnation, the amount of pheromone associated with each arc (i, j) is limited to the ranges $\{\tau_{min}, \tau_{max}\}$. The specific values for the minimum pheromone amount τ_{min} and maximum pheromone amount τ_{max} are determined according to (15) and (14), proposed in the Max-Min Ant System (MMAS).

Once the pheromone matrix is created and initialized, iterative execution of the Free Ant algorithm begins. In each iteration, a limited number of ants are deployed, and to maximize exploration, each ant is placed in a different node to start its construction. However, to minimize computation time, a candidate node list is used. For the design of the ACO algorithms in this work, it was decided that the composition of this list would be a homogeneous mixture of the closest nodes to depot v_0 in each group, the first selected nodes in each route from the solutions with the best results obtained in the previous iteration, and a small number of randomly selected nodes.

Once ant a is placed in its initial node, it starts constructing one of the K routes. The ant continues to select nodes to visit as long as the accumulated total demand in route r is less than the maximum capacity Q of the vehicle. When it is no longer possible to add nodes without violating this constraint, or when all nodes have been visited, the ant returns to the depot. This means that although there are nodes that can be added to the list representing the current route, the size of this list continues to increase by adding more nodes. When no more nodes are available to visit, a new route is created for the remaining available nodes. Once all nodes have been visited, the quality of the generated solution is calculated.

The quality of the solution is determined by the sum of all energies required for each route. This objective function is given by (1). Thus, we can observe that each ant a in the iterative process of the Free Ant algorithm has the total freedom (probabilistically) to choose which node in the graph to visit. Hence its name.

To improve the performance and decision-making capability of each ant a , this work decided to complement the heuristic information η by considering not only the distance d_{ij} but also the energy cost C_{ij} and the savings value g_{ij} , such that:

$$\eta = [d_{ij}]^\beta \cdot [C_{ij}]^\gamma \cdot [g_{ij}]^\delta. \quad (28)$$

where γ and δ are constants that determine the relative importance of C_{ij} and g_{ij} , respectively. Thus, the standard probability distribution used by Free Ant and Restricted Ant is given by:

$$p_a(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha \cdot ([d_{ij}]^\beta [C_{ij}]^\gamma [g_{ij}]^\delta)}{\sum_{u \in J_a(r)} [\tau(r, u)]^\alpha \cdot ([d_{ij}]^\beta [C_{ij}]^\gamma [g_{ij}]^\delta)}, & \text{if } s \in J_a(r), \\ 0, & \text{otherwise,} \end{cases} \quad (29)$$

The details of this strategy, along with others (such as the use of a candidate node list), can be found in the application recommendations of ACO for VRP problems by Bullnheimer, B., Hartl, R., and Strauss, C. [15] and Rizzoli, A.E., Montemanni, R., Lucibello, E. et al. [16].

2) RESTRICTED ANT

In Restricted Ant, the global problem is decomposed into smaller sub-problems: groups. Each group corresponds to one of the K routes. In this manner, we transform the original VRP problem into K TSP problems of lower complexity.

In this version of the BWAS algorithm, all elements τ_{ij} are initialized with a value of τ_{max} .

The iterative process of this algorithm operates differently from the Free Ant algorithm, as the iterations are performed for each group. For each group G_i , a determined number of ant deployments are repeated, and each ant a creates only a partial solution to the problem, specifically the solution for that group G_i .

Once again, we work with a candidate node list, one list for each group, but this time it will consist of a homogeneous distribution of nodes that are closest to the depot, the initial nodes of the best solutions obtained from the previous iteration, and a small number of randomly selected nodes. Each ant a starts its tour at one of these candidate nodes and probabilistically creates a solution based on (29). Naturally, ant a can only visit the nodes belonging to the same group.

Once all iterations in each group are completed, the best solution found in each group is combined into a single final solution, which represents the solution to the original problem.

C. VNS ALGORITHMS

In this study, the VNS was used as a tool to guide the design and development of local search algorithms. In our case, these types of algorithms require an initial solution, provided by the ACO algorithms. Based on these solutions, the algorithm starts exploring the neighborhood space to find (or not) a better local optimum.

Two VNS-inspired algorithms were used in the program: one focused on improving the individual solution of a group, that is, the solution obtained by the Restricted Ant, and another to improve the solution of the entire problem provided by Free Ant.

1) 2-OPT VNS

This first algorithm is a simple variant of VNS and was implemented to improve the solutions delivered by the ants of the Restricted Ant. As the solutions provided by each of these ants are solutions to specific groups, the design of this algorithm is based solely on two intra-cluster operators: “single-route-relocate” and “single-route-swap”.

The single-route-relocate operator relocates a node to a new position within the same route, whereas the single-route-swap operator exchanges the positions of two nodes, within the same route.

2) GENERAL VNS

This algorithm corresponds to a more comprehensive design of the VNS, and its implementation is aimed at improving the solutions obtained in the Free Ant.

In this version of the algorithm, the operators are divided into two groups: intra-cluster and inter-cluster. The intra-cluster operators are the same as in the 2-OPT VNS, while the inter-cluster operators are as follows: “two-routes-relocate,” which relocates a node from one route to a new position in another route; “two-routes-swap,” that exchanges two nodes between different routes; and finally, “two-routes-exchange,” which exchanges two consecutive nodes from one route with two consecutive nodes from another route.

The inter-cluster operators are only viable and applicable when the newly generated solution does not violate the capacity constraint Q for any of the K routes.

D. HYBRID ALGORITHMS

Due to the NP-Hard nature of many problems in the field of engineering, especially those of a combinatorial nature, various authors have proposed algorithms that successfully combine different techniques, such as heuristics, metaheuristics, machine learning, etc., in an attempt to solve these problems. A hybrid algorithm is one that combines two or more algorithms that solve the same problem, either by choosing one (based on the data) or switching between them during the course of the algorithm. This is generally done to combine desired features of each algorithm so that the final algorithm is better than its individual components. Some examples of hybrid algorithms include Machine Learning (ML) with Tabu Search (TS), iterative algorithms with integer programming, Genetic Algorithm (GA) with Local Search (LS), Ant Colony Optimization (ACO) with Local Search (LS), and so on.

Since the algorithms proposed in this work are hybrid algorithms, as they use a combination of clustering techniques with the ACO metaheuristic, a review of the state of the art related to these areas was conducted. We highlight the recent proposals of two studies focused on using K-Means to subdivide the search space for ACO.

One of these studies is by S.A. El-Khatib, Y.A. Skobtsov & S.I. Rodzin [17], who present an efficient algorithm for image segmentation using a method that leverages all the

TABLE 1. Specifications of the testing machine.

Component	Model
Operating System	Windows 10
CPU	AMD Ryzen 7 3700X
RAM	16GB DDR4
Programming Language	Python 3.7.1
IDE	Anaconda

advantages of K-Means and ACO algorithms. The work of Kusumahardhini, N., Hertono, G. F. & Handari, B. D. [18] was also reviewed, in which they combined K-Means with ACO to solve the Multiple Traveling Salesman Problem (MTSP).

V. EXPERIMENTAL RESULTS

In this section, we will present the results of experiments obtained with the Free Ant and Restricted Ant algorithms, which were conducted on a series of instances from the CVRPLIB library. The libraries used for experimentation correspond to “Set A (Augerat, 1995),” “Christofides, Mingozzi and Toth (1979),” and “Golden et al. (1998).”

The choice of these libraries was made because during the literature review stage of the EMVRP, two articles related to EMVRP were identified. The first corresponds to the authors Rathnakumara and Rupasinghe [19], who developed a Tabu Search (TS) algorithm guided by a Machine Learning (ML) model. This algorithm is called ML-TS and was tested by the authors on the “Set A (Augerat, 1995)” instance as an EMVRP problem.

The second article is by the authors Kramer et al. [20], who developed a metaheuristic that combines Local Search (LS) with an integer programming approach and a speed optimization algorithm. This metaheuristic is called ILS-SP-SOA and was tested by the authors on the instances “Christofides, Mingozzi and Toth (1979)” and “Golden et al. (1998)” as an EMVRP problem.

The specifications of the testing machine used for the benchmarks are listed in Table 1. The parameter settings are presented in Table 2. The choice of the values for these parameters is the result of an experimentation phase where empirically, values were sought that would generally yield the best results for both algorithms. The configuration used by Stützle, T., Hoos, H. [7] and that of Cordón et al. [2] served as a starting point, and the search for an optimal configuration began from there.

For this study, each instance of a CVRPLIB library was executed 10 times for each variant of the proposed algorithms. For each instance of each CVRPLIB library, “Set A (Augerat, 1995),” “Christofides, Mingozzi and Toth (1979),” and “Golden et al. (1998),” two tables are presented. The first table corresponds to the best result of each program execution, that is, the best total energy cost of all routes (objective function (1)) and the total computation time in seconds to obtain this solution. The calculation of time considers the time required by the clustering algorithm, added

TABLE 2. Parameter configuration.

Symbol	Parameter	Value	
		Free Ant	Restricted Ant
N	Number of iterations	200-300	100-200
tare	Ratio of maximum capacity to vehicle weight	0.0 - 0.15	0.0 - 0.15
α	Pheromone weight	2	2
β	Distance weight	1	1
γ	Energy weight	3	3
δ	Energy saving weight	1.5	1.5
ρ	Pheromone evaporation weight	0.2	0.2
P_m	Pheromone mutation probability	0.3	0.3
σ	Mutation intensity	3	2
l_0	Exploitation probability	0.3	0.2

to the time of the ACO algorithm, and the time of the VNS local searches. Additionally, a second table presents the averaged results of the 10 executions and their average Gap. The average Gap is calculated by:

$$Gap = 100 \cdot \frac{S_{avg} - S_{best}}{S_{best}} \tag{30}$$

where S_{avg} corresponds to the average value of the 10 solutions found for an instance, and S_{best} is the value of the best solution found among the 10 solutions.

The parameter configurations presented in Table 2 were used during the execution of the experiments. For instances from the libraries “Set A (Augerat, 1995)” and “Christofides, Mingozzi and Toth (1979),” 200 iterations were performed for each execution of the Free Ant algorithm, and 100 iterations were performed per group in the Restricted Ant algorithm. For instances from the “Golden et al. (1998)” library, the number of iterations were 300 and 200, respectively. The number of ants per iteration for the proposed algorithms was as follows: for Free Ant, it was determined by considering half of the total number of nodes in the problem, meaning that the number of ants was equal to 50% of the total number of nodes. In the case of Restricted Ant, the number of nodes varies depending on the group being solved. The number of ants was calculated by taking half of the node count of the current group, resulting in the number of ants being 50% of the number of nodes per group.

In Tables 3-5, we have six columns: Instance, Algorithm, Best Energy (Kj), Avg. Energy (Kj), Gap (%) and Time (Sec). The “Instance” column indicates the name of the instance being solved. The “Best Energy (Kj)” column provides the best energy value found for that instance in terms of kilojoules across the 10 executions. The “Avg. Energy (Kj)” column provides the average energy value across all executions for each algorithm. The “Gap (%)” column represents the averaged gap of all executions, as presented in (30). The “Time (Sec)” column indicates the time taken by the algorithm to find the best energy value in seconds. Finally, the “Algorithm” column indicates the algorithm applied to solve the instance and to which others columns values correspond. Four different algorithms are compared in the table: Free Ant + K-Means, Free Ant + K-Medoids, Restricted

Ant + K-Means, and Restricted Ant + K-Medoids. The best energy value found is the lowest value in terms of kilojoules among all the applied algorithms for a particular instance. The ant-based algorithms are divided into two types: “free” and “restricted.” The “free” algorithms use groups generated by the clustering-based algorithms to initially populate the pheromone matrix. In contrast, clustering-based algorithms are used to establish the search space in the “restricted” algorithms.

During the development of the experiments, we noticed that in some instances of CVRPLIB, the maximum capacity constraint per route was very tight compared to the total node demand. This caused the clustering algorithm (K-Means or K-Medoids) to be unable to find optimal solutions with each node assigned to a group. At the end of an iteration of this algorithm, some nodes remained unassigned. For this reason, the results tables do not show data for Restricted Ant in certain instances. However, in the case of Free Ant, this issue did not hinder the generation of the necessary pheromone matrix. The arcs related to the unassigned nodes still had an initial pheromone trail.

A. SET A (AUGERAT, 1995)

This was the first set of instances considered in this study. This set consists of 22 groups of instances ranging from 30 to 78 nodes (excluding the depot). A vehicle capacity of zero has been considered for this set. The results obtained for this set are presented in Table 3.

B. CHRISTOFIDES, MINGOZZI AND TOTH (1979)

This set consists of six groups of instances ranging from 50 to 199 nodes (excluding the depot). A vehicle tare equivalent to 15% of the maximum capacity was used for this set. For example, if the vehicle capacity Q is 200 units, the tare would be 30 units. 200 iterations were run per execution for the Free Ant algorithm, and 100 iterations were run for the Restricted Ant algorithm. The results obtained for this set are presented in Table 4.

C. GOLDEN ET AL. (1998)

This set consists of 15 groups of instances ranging from 200 to 483 nodes (excluding the depot). The same tare used for the Christofides, Mingozzi, and Toth set, i.e., 15% of the maximum capacity Q , was used for this set as well. The results obtained for this set are presented in Table 5.

VI. RESULTS ANALYSIS

In this section, we analyze the obtained results. To compare the results of the proposed algorithms, we referred to two different works. In the first one, Rathnakumara and Rupasinghe [19] presented the ML-TS algorithm, which is based on clustering and tabu search, to solve the EMVRP. In the second work, Kramer et al. [20] proposed the ILS-SP-SOA algorithm, which combines exact techniques and local searches to solve the Pollution-Routing Problem and other Green-VRP problems.

TABLE 3. Results of Set A (Augerat, 1995).

Instance	Algorithm	Best Energy (Kj)	Avg. Energy (Kj)	Gap (%)	Time (Sec)
A-n32-k5.vrp	Free Ant + K-Means	28132,57	28242,59	0,39	4,36
	Free Ant + K-Medoids	28123,46	28216,90	0,33	8,81
	Restricted Ant + K-Means	29994,14	30464,08	1,57	0,09
	Restricted Ant + K-Medoids	31735,30	32232,43	1,57	0,21
A-n33-k5.vrp	Free Ant + K-Means	22647,43	23086,39	1,94	3,68
	Free Ant + K-Medoids	22684,17	22924,10	1,06	7,87
	Restricted Ant + K-Means	28561,58	29502,73	3,30	0,11
	Restricted Ant + K-Medoids	26169,87	26793,90	1,57	0,22
A-n33-k6.vrp	Free Ant + K-Means	25721,09	25918,84	0,77	3,58
	Free Ant + K-Medoids	25456,98	25729,00	1,07	8,45
	Restricted Ant + K-Means	34216,95	35096,56	2,57	0,10
	Restricted Ant + K-Medoids	31361,09	31832,52	1,50	0,22
A-n34-k5.vrp	Free Ant + K-Means	28701,56	29002,60	1,05	3,74
	Free Ant + K-Medoids	28621,20	28894,00	0,95	8,08
	Restricted Ant + K-Means	34064,33	35543,80	4,34	0,09
	Restricted Ant + K-Medoids	34411,17	34965,58	1,61	0,24
A-n36-k5.vrp	Free Ant + K-Means	26033,68	26584,13	2,11	4,94
	Free Ant + K-Medoids	26150,40	26516,74	1,40	9,98
	Restricted Ant + K-Means	32082,88	34267,88	6,81	0,11
	Restricted Ant + K-Medoids	31797,32	32356,37	1,76	0,26
A-n37-k5.vrp	Free Ant + K-Means	20093,00	20521,10	2,13	5,80
	Free Ant + K-Medoids	20116,20	20351,12	1,17	11,44
	Restricted Ant + K-Means	27248,40	28912,44	6,11	0,19
	Restricted Ant + K-Medoids	27897,84	28815,07	3,29	0,28
A-n37-k6.vrp	Free Ant + K-Means	34845,03	35484,51	1,84	5,15
	Free Ant + K-Medoids	34698,42	35173,16	1,37	10,52
	Restricted Ant + K-Means	40741,55	42023,82	3,15	0,13
	Restricted Ant + K-Medoids	43367,11	43798,69	1,00	0,25
A-n38-k5.vrp	Free Ant + K-Means	26215,40	27287,51	4,09	8,95
	Free Ant + K-Medoids	33502,08	34163,30	1,97	0,27
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
A-n39-k5.vrp	Free Ant + K-Means	30502,15	31153,66	2,14	5,32
	Free Ant + K-Medoids	30552,76	30895,24	1,12	10,12
	Restricted Ant + K-Means	39530,56	41427,66	4,80	0,17
	Restricted Ant + K-Medoids	43274,74	44285,30	2,34	0,30
A-n39-k6.vrp	Free Ant + K-Means	27453,89	28006,44	2,01	6,57
	Free Ant + K-Medoids	27338,30	27794,55	1,67	13,51
	Restricted Ant + K-Means	29894,35	30630,77	2,46	0,17
	Restricted Ant + K-Medoids	33519,50	33982,09	1,38	0,27
A-n44-k5.vrp	Free Ant + K-Means	32364,93	34022,42	5,12	6,67
	Free Ant + K-Medoids	33005,19	33782,36	2,35	13,00
	Restricted Ant + K-Means	39239,11	39618,77	0,97	0,34
	Restricted Ant + K-Medoids	43739,96	45294,62	3,55	0,32
A-n45-k6.vrp	Free Ant + K-Means	37451,93	34022,42	5,12	5,08
	Free Ant + K-Medoids	45784,95	46596,37	1,77	0,48
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
A-n45-k7.vrp	Free Ant + K-Means	44079,64	44851,77	1,75	7,32
	Free Ant + K-Medoids	43662,78	44510,43	1,94	15,57
	Restricted Ant + K-Means	53187,20	54240,14	1,98	0,17
	Restricted Ant + K-Medoids	50754,82	52573,68	3,58	0,31
A-n46-k7.vrp	Free Ant + K-Means	32907,00	33280,96	1,14	8,14
	Free Ant + K-Medoids	32830,47	33004,38	0,53	19,19
	Restricted Ant + K-Means	40011,62	41981,55	4,92	0,20
	Restricted Ant + K-Medoids	40099,96	41338,12	3,09	0,32
A-n48-k7.vrp	Free Ant + K-Means	41313,67	41814,59	1,21	8,75
	Free Ant + K-Medoids	40737,93	41182,57	1,09	18,17
	Restricted Ant + K-Means	46093,22	46593,00	1,08	0,22
	Restricted Ant + K-Medoids	51339,67	52178,40	1,63	0,32
A-n53-k7.vrp	Free Ant + K-Means	39649,61	40717,87	2,69	11,04
	Free Ant + K-Medoids	38983,34	39988,27	2,58	22,23
	Restricted Ant + K-Means	47226,27	48007,44	1,65	0,68
	Restricted Ant + K-Medoids	47171,63	47950,15	1,65	0,40
A-n54-k7.vrp	Free Ant + K-Means	46489,04	47508,71	2,19	10,60
	Free Ant + K-Medoids	46144,38	46917,30	1,68	20,35
	Restricted Ant + K-Means	54976,37	56008,62	1,88	0,36
	Restricted Ant + K-Medoids	56334,78	57715,03	2,45	0,43

TABLE 3. (Continued.) Results of Set A (Augerat, 1995).

A-n55-k9.vrp	Free Ant + K-Means	41675,50	42274,94	1,44	10,16
	Free Ant + K-Medoids	41359,27	41900,31	1,31	22,40
	Restricted Ant + K-Means	46990,42	48391,49	2,98	0,25
	Restricted Ant + K-Medoids	54541,21	55218,90	1,24	0,37
A-n60-k9.vrp	Free Ant + K-Means	53028,40	53444,09	0,78	15,62
	Free Ant + K-Medoids	51974,10	53050,69	2,07	35,32
	Restricted Ant + K-Means	58181,62	58874,81	1,19	0,42
	Restricted Ant + K-Medoids	62720,76	64276,40	2,48	0,43
A-n62-k8.vrp	Free Ant + K-Means	48915,26	49447,71	1,09	15,42
	Free Ant + K-Medoids	48606,07	49050,73	0,91	34,78
	Restricted Ant + K-Means	59226,45	61071,98	3,12	0,32
	Restricted Ant + K-Medoids	59613,30	60888,54	2,14	0,49
A-n63-k9.vrp	Free Ant + K-Means	66038,45	67078,80	1,58	15,83
	Free Ant + K-Medoids	65274,17	66166,95	1,37	29,88
	Restricted Ant + K-Means	78246,73	79302,03	1,35	1,16
	Restricted Ant + K-Medoids	75045,56	76475,71	1,91	0,46
A-n63-k10.vrp	Free Ant + K-Means	49831,84	50981,14	2,31	17,56
	Free Ant + K-Medoids	49556,67	50500,02	1,90	35,33
	Restricted Ant + K-Means	60205,67	61871,58	2,77	0,47
	Restricted Ant + K-Medoids	61993,89	62901,61	1,46	0,44
A-n64-k9.vrp	Free Ant + K-Means	52538,51	53530,66	1,89	16,56
	Free Ant + K-Medoids	52344,74	53040,16	1,33	30,42
	Restricted Ant + K-Means	60581,58	71401,95	17,86	0,65
	Restricted Ant + K-Medoids	60316,68	61473,71	1,92	0,49
A-n65-k9.vrp	Free Ant + K-Means	47857,46	49851,79	4,17	13,60
	Free Ant + K-Medoids	48471,83	49860,35	2,86	27,61
	Restricted Ant + K-Means	57966,17	59555,15	2,74	1,21
	Restricted Ant + K-Medoids	63583,58	64162,12	0,91	0,50
A-n69-k9.vrp	Free Ant + K-Means	44158,77	45133,88	2,21	18,08
	Free Ant + K-Medoids	43163,27	45028,55	4,32	39,53
	Restricted Ant + K-Means	50941,42	51560,38	1,22	0,34
	Restricted Ant + K-Medoids	59307,66	60912,57	2,71	0,54
A-n80-k10.vrp	Free Ant + K-Means	68532,00	70152,18	2,36	25,15
	Free Ant + K-Medoids	67830,96	69354,39	2,25	58,19
	Restricted Ant + K-Means	82074,80	83977,46	2,32	0,56
	Restricted Ant + K-Medoids	90259,71	92640,93	2,64	0,67

Both authors achieved favorable results in their respective benchmarks using the CVRPLIB library.

A. SET A (AUGERAT, 1995)

Table 6 present the best results obtained and compare them with the best results of the ML-TS algorithm. Additionally, the difference (Gap) between the best energy obtained by Free Ant + K-Means, Free Ant + K-Medoids, Restricted Ant + K-Means, and Restricted Ant + K-Medoids is shown compared to the results provided by Rathnakumara and Rupasinghe. The calculation of this Gap is done using the next equation:

$$\text{Gap Best Energy} = 100 \cdot \frac{S_{ML-TS} - S_{best}}{S_{best}}. \quad (31)$$

Here, S_{ML-TS} corresponds to the best energy obtained by ML-TS for the instances in Set A (Augerat, 1995).

The data presented in Table 3 indicate that, in general, in terms of energy minimization, the Free Ant algorithms achieve better results than the Restricted Ant algorithms.

Furthermore, Table 3 reveals that Free Ant + K-Medoids achieves a higher solution quality, where lower values indicate better solutions. On the other hand, the Restricted Ant algorithms tend to generate lower-quality solutions compared to Free Ant algorithm. However, Restricted Ant algorithms are much more efficient, with Restricted Ant + K-Means being the fastest algorithm in completing its iterations

in most problems. This is evident in the lower values in the Time (Sec) column.

Compared to ML-TS, the Free Ant algorithms achieved better results for 20 out of 28 instances in Set A (Augerat, 1995), with an average Gap of -7.08% below the results reported by G. W. H. H. P. Rathnakumara and T. D. Rupasinghe. Restricted Ant yielded an average Gap of 12.21% above the results.

B. CHRISTOFIDES, MINGOZZI AND TOTH (1979)

Table 7 presents data corresponding to the ‘‘Christofides, Mingozi and Toth (1979)’’ dataset. In this case, the results obtained in the experimentation phase are compared to those presented by Kramer et al., and unlike Table 6, two additional columns are added: ‘‘Avg. Energy (Kj)’’ and ‘‘Gap Avg. Energy (%)’’. The first column shows the average energy obtained in all runs for each instance, whereas the second column shows a new Gap (33), which evaluates the average differences between each ACO algorithm and that of Kramer et al.

$$\text{Gap Best Energy} = 100 \cdot \frac{S_{ILS-SP-SOA_{best}} - S_{best}}{S_{best}}. \quad (32)$$

$$\text{Gap Avg. Energy} = 100 \cdot \frac{S_{ILS-SP-SOA_{avg}} - S_{best}}{S_{best}}. \quad (33)$$

TABLE 4. Results of Christofides, Mingozi and Toth (1979).

Instance	Algorithm	Best Energy (Kj)	Avg. Energy (Kj)	Gap (%)	Time (Sec)
CMT1	Free Ant + K-Means	47285,57	49075,65	3,79	16,03
	Free Ant + K-Medoids	47520,35093	48750,70	2,59	15,58
	Restricted Ant + K-Means	52202,76668	52946,17002	1,42	0,69
	Restricted Ant + K-Medoids	68250,19697	69180,65327	1,36	0,48
CMT2	Free Ant + K-Means	65474,67	67032,21	2,38	40,01
	Free Ant + K-Medoids	64617,56	67432,8484	4,36	41,22
	Restricted Ant + K-Means	76620,42739	77477,23087	1,12	1,00
	Restricted Ant + K-Medoids	81621,71044	82594,08476	1,19	0,59
CMT3	Free Ant + K-Means	88701,93919	92377,56816	4,14	86,94
	Free Ant + K-Medoids	90008,64901	93870,56241	4,29	78,58
	Restricted Ant + K-Means	107567,5445	110256,5459	2,50	1,33
	Restricted Ant + K-Medoids	108605,0933	110095,3294	1,37	1,16
CMT4	Free Ant + K-Means	119587,6639	122032,5139	2,04	181,81
	Free Ant + K-Medoids	118934,4605	122811,2655	3,26	181,36
	Restricted Ant + K-Means	139418,1466	141368,3512	1,40	2,90
	Restricted Ant + K-Medoids	143408,3766	145602,8008	1,53	1,82
CMT5	Free Ant + K-Means	-	-	-	-
	Free Ant + K-Medoids	-	-	-	-
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
CMT6	Free Ant + K-Means	48338,20181	49198,03783	1,78	16,56
	Free Ant + K-Medoids	48862,81146	50355,77099	3,06	16,53
	Restricted Ant + K-Means	52879,4187	53613,06742	1,39	0,67
	Restricted Ant + K-Medoids	67655,89641	70303,42844	3,91	0,48
CM7	Free Ant + K-Means	65006,72245	66995,73992	3,06	42,48
	Free Ant + K-Medoids	65114,05557	66444,00333	2,04	41,58
	Restricted Ant + K-Means	76598,16731	77568,01755	1,27	1,00
	Restricted Ant + K-Medoids	81885,45067	82669,40139	0,96	0,59
CMT8	Free Ant + K-Means	90033,46874	93089,23784	3,39	84,85
	Free Ant + K-Medoids	88713,75843	91853,01172	3,54	84,20
	Restricted Ant + K-Means	107714,5209	109811,5258	1,95	1,33
	Restricted Ant + K-Medoids	107858,065	109395,5158	1,43	1,16
CMT9	Free Ant + K-Means	134857,0402	138457,3964	2,67	161,16
	Free Ant + K-Medoids	134365,7	137979,2573	2,69	170,39
	Restricted Ant + K-Means	145218,5761	146878,8732	1,14	2,84
	Restricted Ant + K-Medoids	155280,1537	156991,4437	1,10	1,78
CMT10	Free Ant + K-Means	189434,5228	194412,5106	2,63	185,65
	Free Ant + K-Medoids	191174,9854	195821,7924	2,43	187,79
	Restricted Ant + K-Means	198776,2682	201234,458	1,24	11,44
	Restricted Ant + K-Medoids	233234,3957	234469,4058	0,53	2,83
CMT11	Free Ant + K-Means	128324,9622	129849,5328	1,19	105,66
	Free Ant + K-Medoids	127210,3278	128866,4151	1,30	103,29
	Restricted Ant + K-Means	139784,4039	140317,8446	0,38	2,32
	Restricted Ant + K-Medoids	139462,7103	141535,2632	1,49	1,98
CMT12	Free Ant + K-Means	93933,72264	95050,10525	1,19	75,53
	Free Ant + K-Medoids	93875,06272	94863,47909	1,05	81,95
	Restricted Ant + K-Means	139784,4039	96190,43892	0,47	2,32
	Restricted Ant + K-Medoids	105450,1425	106282,562	0,79	1,00
CMT13	Free Ant + K-Means	127631,7756	129313,9953	1,32	106,00
	Free Ant + K-Medoids	128967,6665	129883,6682	0,71	105,42
	Restricted Ant + K-Means	139808,132	140369,1953	0,40	2,32
	Restricted Ant + K-Medoids	140042,42	141123,8484	0,77	1,98
CMT14	Free Ant + K-Means	94119,41519	94726,97567	0,65	78,61
	Free Ant + K-Medoids	94096,14034	95029,76001	0,99	80,84
	Restricted Ant + K-Means	95745,31951	96051,87834	0,32	1,40
	Restricted Ant + K-Medoids	105845,5458	106268,3895	0,40	1,01

Here, $S_{ILS-SP-SOA_{best}}$ corresponds to the best solution presented by Kramer et al. for each instance, and $S_{ILS-SP-SOA_{avg}}$ is the average of Kramer et al.'s solutions for each instance.

During the experimentation phase, we decided to test the effectiveness of using clusters with the ant colony optimization metaheuristic. As a result, we conducted tests with Free Ant, both with and without the use of K-Means or K-Medoids. We observed that when the algorithm was

assisted by clusters, it tended to converge towards a set of very similar local optima. This behavior did not manifest in the same way when using Free Ant without clustering; in this case, the convergence occurred in a much more staggered manner throughout the iterations. This discrepancy is attributed to the bias introduced by the information of cluster arcs in the pheromone matrix. The convergence graph without the use of clustering can be seen in Fig. 2, while

TABLE 5. Results of Golden et al. (1998).

Instance	Algorithm	Best Energy (Kj)	Avg. Energy (Kj)	Gap (%)	Time (Sec)
Golden_1	Free Ant + K-Means	1794485,33	1839994,09	2,54	313,18
	Free Ant + K-Medoids	1794241,83	1840565,07	2,58	332,84
	Restricted Ant + K-Means	1816835,04	1848875,53	1,76	9,86
	Restricted Ant + K-Medoids	1947157,52	1963021,21	0,81	5,24
Golden_2	Free Ant + K-Means	3392031,91	3543677,21	4,47	519,22
	Free Ant + K-Medoids	3428805,74	3505641,92	2,24	716,08
	Restricted Ant + K-Means	3599353,15	3673938,84	2,07	11,04
	Restricted Ant + K-Medoids	3437439,70	3490109,42	1,53	8,43
Golden_3	Free Ant + K-Means	6188152,51	6419673,13	3,74	784,14
	Free Ant + K-Medoids	6283131,21	6454124,05	2,72	1039,36
	Restricted Ant + K-Means	6557807,89	6857022,53	4,56	28,85
	Restricted Ant + K-Medoids	6562403,80	6661796,57	1,51	14,38
Golden_4	Free Ant + K-Means	8605180,34	9012942,71	4,74	1122,42
	Free Ant + K-Medoids	8765434,46	9035752,79	3,08	1475,99
	Restricted Ant + K-Means	9079222,64	9268418,84	2,08	31,92
	Restricted Ant + K-Medoids	8984686,99	9165456,57	2,01	18,89
Golden_5	Free Ant + K-Means	3197097,79	3546780,56	10,94	228,32
	Free Ant + K-Medoids	3413564,70	3549348,08	3,98	297,17
	Restricted Ant + K-Means	3240939,18	3289025,29	1,48	9,50
	Restricted Ant + K-Medoids	3884274,63	3957098,38	1,87	6,60
Golden_6	Free Ant + K-Means	4425352,82	4707693,42	6,38	416,09
	Free Ant + K-Medoids	4614756,65	4710028,07	2,06	551,27
	Restricted Ant + K-Means	4572758,83	4639142,62	1,45	11,89
	Restricted Ant + K-Medoids	4668692,11	4718943,21	1,08	9,14
Golden_7	Free Ant + K-Means	5807380,97	6021276,07	3,68	686,08
	Free Ant + K-Medoids	5781736,01	5922564,39	2,44	902,14
	Restricted Ant + K-Means	6332893,46	6483549,56	2,38	20,47
	Restricted Ant + K-Medoids	6172468,04	6273870,76	1,64	13,01
Golden_8	Free Ant + K-Means	6536564,26	6750035,09	3,27	972,11
	Free Ant + K-Medoids	6511290,55	6723972,49	3,27	1228,66
	Restricted Ant + K-Means	6725806,14	6849882,83	1,84	31,62
	Restricted Ant + K-Medoids	6772409,10	6899067,29	1,87	15,84
Golden_9	Free Ant + K-Means	354352,28	368537,73	4,00	454,30
	Free Ant + K-Medoids	395652,02	397140,06	0,38	5,20
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_10	Free Ant + K-Means	469330,80	476837,93	1,60	713,93
	Free Ant + K-Medoids	510679,39	512969,95	0,45	7,05
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_11	Free Ant + K-Means	589171,40	606887,14	3,01	973,63
	Free Ant + K-Medoids	636908,70	640218,25	0,52	9,86
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_12	Free Ant + K-Means	723836,53	737304,75	1,86	1461,72
	Free Ant + K-Medoids	774969,45	780308,23	0,69	13,44
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_13	Free Ant + K-Means	502177,89	510544,17	1,67	348,40
	Free Ant + K-Medoids	501018,13	511451,93	2,08	468,31
	Restricted Ant + K-Means	626314,59	645567,40	3,07	14,18
	Restricted Ant + K-Medoids	561491,28	562993,32	0,27	3,10
Golden_14	Free Ant + K-Means	-	-	-	-
	Free Ant + K-Medoids	-	-	-	-
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_15	Free Ant + K-Means	826239,20	840338,91	1,71	1031,22
	Free Ant + K-Medoids	902635,42	913921,38	1,25	6,80
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_16	Free Ant + K-Means	1089105,77	1121642,41	2,99	1297,08
	Free Ant + K-Medoids	1157229,24	1182172,87	2,16	9,41
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_17	Free Ant + K-Means	78001,30	80511,92	3,22	326,50
	Free Ant + K-Medoids	79682,76	81071,13	1,74	453,16
	Restricted Ant + K-Means	89158,72	89896,18	0,83	6,46
	Restricted Ant + K-Medoids	93314,59	94060,09	0,80	3,45

TABLE 5. (Continued.) Results of Golden et al. (1998).

Golden_18	Free Ant + K-Means	116825,42	118675,91	1,58	481,46
	Free Ant + K-Medoids	118110,08	119299,53	1,01	621,00
	Restricted Ant + K-Means	131109,93	132337,36	0,94	21,03
	Restricted Ant + K-Medoids	141625,62	144933,18	2,34	3,55
Golden_19	Free Ant + K-Means	157472,61	160434,10	1,88	677,29
	Free Ant + K-Medoids	156511,52	158588,24	1,33	902,57
	Restricted Ant + K-Means	184050,35	185386,21	0,73	21,99
	Restricted Ant + K-Medoids	194379,10	196016,43	0,84	4,42
Golden_20	Free Ant + K-Means	236472,51	238312,03	0,78	874,24
	Free Ant + K-Medoids	238814,66	241505,20	1,13	873,76
	Restricted Ant + K-Means	239991,48	242172,45	0,91	25,80
	Restricted Ant + K-Medoids	264870,88	268010,77	1,19	5,35

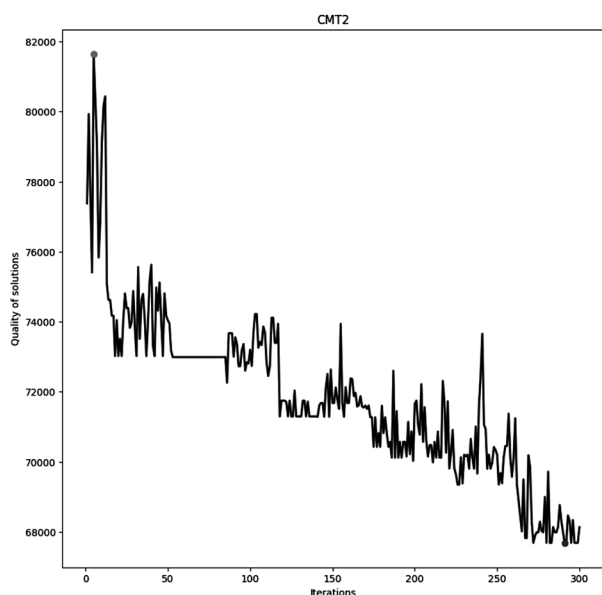


FIGURE 2. Convergence chart of the free ant algorithm without the use of a clustering algorithm in the CMT2 instance.

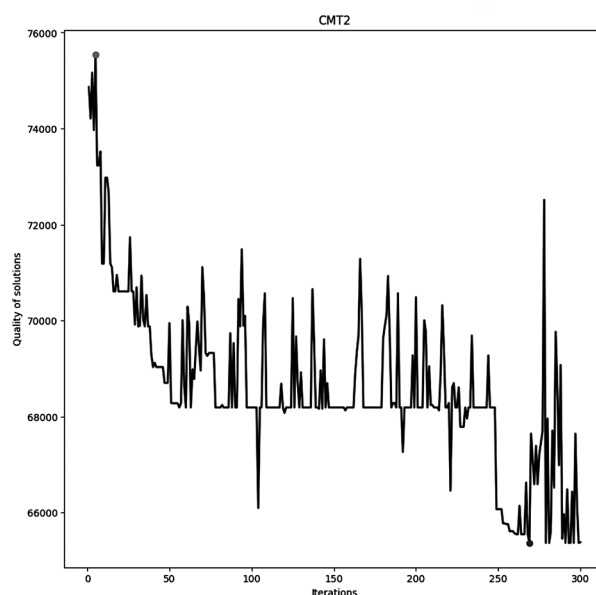


FIGURE 3. Convergence chart of the free ant algorithm with the use of K-Means in the CMT2 instance.

the convergence graph with the use of K-Means is depicted in Fig. 3.

Now, according to Table 4, we can see that once again, the Free Ant algorithms achieve better solutions in terms of energy consumption than the Restricted Ant algorithms. However, it is also evident that the Restricted Ant has significantly shorter execution times than the Free Ant. This suggests that allowing ants to explore the search space more freely and using clustering techniques such as K-Means and K-Medoids to guide pheromone formation produces superior results. However, the idea that restricting ant exploration helps reduce the time required to find solutions is also confirmed. It is important to note that this restriction limits the ability of the algorithm to find optimal solutions.

Table 7 indicate that the Free Ant algorithms achieved better results in 1 out of the 14 instances of Christofides, Mingozzi and Toth (1979), specifically CMT13 using Free Ant + K-Means. However, an average Gap of 10.33% above the results reported by Kramer et al. was observed in the “Gap Best Energy (%)” column for the Free Ant algorithms, and a

Gap of 30.20% for the Restricted Ant in the same column. In the “Gap Avg. Energy (%)” column, Free Ant had an average gap of 12.90% above ILS-SP-SOA, and 29.88% for Restricted Ant.

This time, there is an increase in the average Gap value for both Free Ant and Restricted Ant. This increase in the average Gap is approximately double that of the Set A (Augerat, 1995) dataset.

C. GOLDEN ET AL. (1998)

Table 5 presents the results obtained in Golden et al. (1998). In general, the Free Ant algorithms achieved the best results in terms of the energy consumption. The K-Means and K-Medoids algorithms showed similar results in terms of the number of optimal solutions found. On this occasion, the Restricted Ant algorithms proved to be much more efficient and faster in completing the executions, suggesting that for medium to large-sized problems, this type of restricted algorithm is a better choice than Free Ant when the runtime is crucial.

TABLE 6. Comparison results of Set A (Augerat, 1995).

Instance	Algorithm	Best Energy (Kj)	Gap Best Energy (%)
A-n32-k5.vrp	ML-TS [19]	28560,00	0,00
	Free Ant + K-Means	28132,57	-1,50
	Free Ant + K-Medoids	28123,46	-1,53
	Restricted Ant + K-Means	29994,14	5,02
	Restricted Ant + K-Medoids	31735,30	11,12
A-n33-k5.vrp	ML-TS [19]	29789,00	0,00
	Free Ant + K-Means	22647,43	-23,97
	Free Ant + K-Medoids	22684,17	-23,85
	Restricted Ant + K-Means	28561,58	-4,12
	Restricted Ant + K-Medoids	26169,87	-12,15
A-n33-k6.vrp	ML-TS [19]	29045,00	0,00
	Free Ant + K-Means	25721,09	-11,44
	Free Ant + K-Medoids	25456,98	-12,35
	Restricted Ant + K-Means	34216,95	17,81
	Restricted Ant + K-Medoids	31361,09	7,97
A-n34-k5.vrp	ML-TS [19]	31284,00	0,00
	Free Ant + K-Means	28701,56	-8,25
	Free Ant + K-Medoids	28621,20	-8,51
	Restricted Ant + K-Means	34064,33	8,89
	Restricted Ant + K-Medoids	34411,17	10,00
A-n36-k5.vrp	ML-TS [19]	32149,00	0,00
	Free Ant + K-Means	26033,68	-19,02
	Free Ant + K-Medoids	26150,40	-18,66
	Restricted Ant + K-Means	32082,88	-0,21
	Restricted Ant + K-Medoids	31797,32	-1,09
A-n37-k5.vrp	ML-TS [19]	32798,00	0,00
	Free Ant + K-Means	20093,00	-38,74
	Free Ant + K-Medoids	20116,20	-38,67
	Restricted Ant + K-Means	27248,40	-16,92
	Restricted Ant + K-Medoids	27897,84	-14,94
A-n37-k6.vrp	ML-TS [19]	33892,00	0,00
	Free Ant + K-Means	34845,03	2,81
	Free Ant + K-Medoids	34698,42	2,38
	Restricted Ant + K-Means	40741,55	20,21
	Restricted Ant + K-Medoids	43367,11	27,96
A-n38-k5.vrp	ML-TS [19]	36782,00	0,00
	Free Ant + K-Means	26215,40	-28,73
	Free Ant + K-Medoids	33502,08	-8,92
	Restricted Ant + K-Means	-	-
	Restricted Ant + K-Medoids	-	-
A-n39-k5.vrp	ML-TS [19]	35123,00	0,00
	Free Ant + K-Means	30502,15	-13,16
	Free Ant + K-Medoids	30552,76	-13,01
	Restricted Ant + K-Means	39530,56	12,55
	Restricted Ant + K-Medoids	43274,74	23,21
A-n39-k6.vrp	ML-TS [19]	33290,00	0,00
	Free Ant + K-Means	27453,89	-17,53
	Free Ant + K-Medoids	27338,30	-17,88
	Restricted Ant + K-Means	29894,35	-10,20
	Restricted Ant + K-Medoids	33519,50	0,69
A-n44-k5.vrp	ML-TS [19]	34295,00	0,00
	Free Ant + K-Means	32364,93	-5,63
	Free Ant + K-Medoids	33005,19	-3,76
	Restricted Ant + K-Means	39239,11	14,42
	Restricted Ant + K-Medoids	43739,96	27,54
A-n45-k6.vrp	ML-TS [19]	34890,00	0,00
	Free Ant + K-Means	37451,93	7,34
	Free Ant + K-Medoids	45784,95	31,23
	Restricted Ant + K-Means	-	-
	Restricted Ant + K-Medoids	-	-
A-n45-k7.vrp	ML-TS [19]	37821,00	0,00
	Free Ant + K-Means	44079,64	16,55
	Free Ant + K-Medoids	43662,78	15,45
	Restricted Ant + K-Means	53187,20	40,63
	Restricted Ant + K-Medoids	50754,82	34,20
A-n46-k7.vrp	ML-TS [19]	41294,00	0,00
	Free Ant + K-Means	32907,00	-20,31
	Free Ant + K-Medoids	32830,47	-20,50
	Restricted Ant + K-Means	40011,62	-3,11
	Restricted Ant + K-Medoids	40099,96	-2,89

TABLE 6. (Continued.) Comparison results of Set A (Augerat, 1995).

A-n48-k7.vrp	ML-TS [19]	43193,00	0,00
	Free Ant + K-Means	41313,67	-4,35
	Free Ant + K-Medoids	40737,93	-5,68
	Restricted Ant + K-Means	46093,22	6,71
	Restricted Ant + K-Medoids	51339,67	18,86
A-n53-k7.vrp	ML-TS [19]	42159,00	0,00
	Free Ant + K-Means	39649,61	-5,95
	Free Ant + K-Medoids	38983,34	-7,53
	Restricted Ant + K-Means	47226,27	12,02
	Restricted Ant + K-Medoids	47171,63	11,89
A-n54-k7.vrp	ML-TS [19]	45248,00	0,00
	Free Ant + K-Means	46489,04	2,74
	Free Ant + K-Medoids	46144,38	1,98
	Restricted Ant + K-Means	54976,37	21,50
	Restricted Ant + K-Medoids	56334,78	24,50
A-n55-k9.vrp	ML-TS [19]	46193,00	0,00
	Free Ant + K-Means	41675,50	-9,78
	Free Ant + K-Medoids	41359,27	-10,46
	Restricted Ant + K-Means	46990,42	1,73
	Restricted Ant + K-Medoids	54541,21	18,07
A-n60-k9.vrp	ML-TS [19]	46890,00	0,00
	Free Ant + K-Means	53028,40	13,09
	Free Ant + K-Medoids	51974,10	10,84
	Restricted Ant + K-Means	58181,62	24,08
	Restricted Ant + K-Medoids	62720,76	33,76
A-n62-k8.vrp	ML-TS [19]	52134,00	0,00
	Free Ant + K-Means	48915,26	-6,17
	Free Ant + K-Medoids	48606,07	-6,77
	Restricted Ant + K-Means	59226,45	13,60
	Restricted Ant + K-Medoids	59613,30	14,35
A-n63-k9.vrp	ML-TS [19]	54389,00	0,00
	Free Ant + K-Means	66038,45	21,42
	Free Ant + K-Medoids	65274,17	20,01
	Restricted Ant + K-Means	78246,73	43,86
	Restricted Ant + K-Medoids	75045,56	37,98
A-n63-k10.vrp	ML-TS [19]	51239,00	0,00
	Free Ant + K-Means	49831,84	-2,75
	Free Ant + K-Medoids	49556,67	-3,28
	Restricted Ant + K-Means	60205,67	17,50
	Restricted Ant + K-Medoids	61993,89	20,99
A-n64-k9.vrp	ML-TS [19]	52489,00	0,00
	Free Ant + K-Means	52538,51	0,09
	Free Ant + K-Medoids	52344,74	-0,27
	Restricted Ant + K-Means	60581,58	15,42
	Restricted Ant + K-Medoids	60316,68	14,91
A-n65-k9.vrp	ML-TS [19]	54256,00	0,00
	Free Ant + K-Means	47857,46	-11,79
	Free Ant + K-Medoids	48471,83	-10,66
	Restricted Ant + K-Means	57966,17	6,84
	Restricted Ant + K-Medoids	63583,58	17,19
A-n69-k9.vrp	ML-TS [19]	66390,00	0,00
	Free Ant + K-Means	44158,77	-33,49
	Free Ant + K-Medoids	43163,27	-34,99
	Restricted Ant + K-Means	50941,42	-23,27
	Restricted Ant + K-Medoids	59307,66	-10,67
A-n80-k10.vrp	ML-TS [19]	69578,00	0,00
	Free Ant + K-Means	68532,00	-1,50
	Free Ant + K-Medoids	67830,96	-2,51
	Restricted Ant + K-Means	82074,80	17,96
	Restricted Ant + K-Medoids	90259,71	29,72

Furthermore, the results presented in Table 5 were compared once again with those obtained by Kramer et al. A comparison with the results of Kramer et al. is presented in Table 8. The Free Ant algorithms achieved better results in one out of the 20 instances of Golden et al. (1998), specifically in Golden_17 using the Free Ant + K-Means algorithm. On the other hand, an average Gap of

24.28% above the results reported by Kramer et al. was observed for the Free Ant algorithms in the “Gap Best Energy (%)” column, and a 32.87% Gap for the Restricted Ant algorithms in the same column. Regarding the “Gap Avg. Energy (%)” column, Free Ant had an average of 27.45% above ILS-SP-SOA, and a 35.00% difference for the Restricted Ant.

TABLE 7. Comparison results of Christofides, Mingozzi and Toth (1979).

Instance	Algorithm	Best Energy (Kj)	Gap Best Energy (%)	Avg. Energy (Kj)	Gap Avg. Energy (%)
CMT1	ILS-SP-SOA [20]	46210,35	0,00	46383,05	0,00
	Free Ant + K-Means	47285,57	2,33	49075,65	5,81
	Free Ant + K-Medoids	47520,35	2,83	48750,70	5,10
	Restricted Ant + K-Means	52202,77	12,97	52946,17	14,15
	Restricted Ant + K-Medoids	68250,20	47,69	69180,65	49,15
CMT2	ILS-SP-SOA [20]	60564,99	0,00	60575,76	0,00
	Free Ant + K-Means	65474,67	8,11	67032,21	10,66
	Free Ant + K-Medoids	64617,55	6,69	67432,85	11,32
	Restricted Ant + K-Means	76620,43	26,51	77477,23	27,90
	Restricted Ant + K-Medoids	81621,71	34,77	82594,08	36,35
CMT3	ILS-SP-SOA [20]	83490,41	0,00	83490,41	0,00
	Free Ant + K-Means	88701,94	6,24	92377,57	10,64
	Free Ant + K-Medoids	90008,65	7,81	93870,56	12,43
	Restricted Ant + K-Means	107567,54	28,84	110256,55	32,06
	Restricted Ant + K-Medoids	108605,09	30,08	110095,33	31,87
CMT4	ILS-SP-SOA [20]	105842,20	0,00	105869,55	0,00
	Free Ant + K-Means	119587,66	12,99	122032,51	15,27
	Free Ant + K-Medoids	118934,46	12,37	122811,27	16,00
	Restricted Ant + K-Means	139418,15	31,72	141368,35	33,53
	Restricted Ant + K-Medoids	143408,38	35,49	145602,80	37,53
CMT5	ILS-SP-SOA [20]	135123,96	0,00	135142,92	0,00
	Free Ant + K-Means	-	-	-	-
	Free Ant + K-Medoids	-	-	-	-
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
CMT6	ILS-SP-SOA [20]	42403,25	0,00	42403,25	0,00
	Free Ant + K-Means	48338,20	14,00	49198,04	16,02
	Free Ant + K-Medoids	48862,81	15,23	50355,77	18,75
	Restricted Ant + K-Means	52879,42	24,71	53613,07	26,44
	Restricted Ant + K-Medoids	67655,90	59,55	70303,43	65,80
CM7	ILS-SP-SOA [20]	61164,47	0,00	61164,47	0,00
	Free Ant + K-Means	65006,72	6,28	66995,74	9,53
	Free Ant + K-Medoids	65114,06	6,46	66444,00	8,63
	Restricted Ant + K-Means	76598,17	25,23	77568,02	26,82
	Restricted Ant + K-Medoids	81885,45	33,88	82669,40	35,16
CMT8	ILS-SP-SOA [20]	80247,16	0,00	80247,16	0,00
	Free Ant + K-Means	90033,47	12,20	93089,24	16,00
	Free Ant + K-Medoids	88713,76	10,55	91853,01	14,46
	Restricted Ant + K-Means	107714,52	34,23	109811,53	36,84
	Restricted Ant + K-Medoids	107858,07	34,41	109395,52	36,32
CMT9	ILS-SP-SOA [20]	105281,84	0,00	105618,73	0,00
	Free Ant + K-Means	134857,04	28,09	138457,40	31,09
	Free Ant + K-Medoids	134365,70	27,62	137979,26	30,64
	Restricted Ant + K-Means	145218,58	37,93	146878,87	39,07
	Restricted Ant + K-Medoids	155280,15	47,49	156991,44	48,64
CMT10	ILS-SP-SOA [20]	135819,48	0,00	136390,40	0,00
	Free Ant + K-Means	189434,52	39,48	194412,51	42,54
	Free Ant + K-Medoids	191174,99	40,76	195821,79	43,57
	Restricted Ant + K-Means	198776,27	46,35	201234,46	47,54
	Restricted Ant + K-Medoids	233234,40	71,72	234469,41	71,91
CMT11	ILS-SP-SOA [20]	123348,14	0,00	123430,83	0,00
	Free Ant + K-Means	128324,96	4,03	129849,53	5,20
	Free Ant + K-Medoids	127210,33	3,13	128866,42	4,40
	Restricted Ant + K-Means	139784,40	13,33	140317,84	13,68
	Restricted Ant + K-Medoids	139462,71	13,06	141535,26	14,67
CMT12	ILS-SP-SOA [20]	93521,30	0,00	93521,30	0,00
	Free Ant + K-Means	93933,72	0,44	95050,11	1,63
	Free Ant + K-Medoids	93875,06	0,38	94863,48	1,44
	Restricted Ant + K-Means	139784,40	49,47	96190,44	2,85
	Restricted Ant + K-Medoids	105450,14	12,76	106282,56	13,65
CMT13	ILS-SP-SOA [20]	128345,69	0,00	128356,05	0,00
	Free Ant + K-Means	127631,78	-0,56	129314,00	0,75
	Free Ant + K-Medoids	128967,67	0,48	129883,67	1,19
	Restricted Ant + K-Means	139808,13	8,93	140369,20	9,36
	Restricted Ant + K-Medoids	140042,42	9,11	141123,85	9,95
CMT14	ILS-SP-SOA [20]	93827,15	0,00	93827,15	0,00
	Free Ant + K-Means	94119,42	0,31	94726,98	0,96
	Free Ant + K-Medoids	94096,14	0,29	95029,76	1,28
	Restricted Ant + K-Means	95745,32	2,04	96051,88	2,37
	Restricted Ant + K-Medoids	105845,55	12,81	106268,39	13,26

TABLE 8. Comparison results of Golden et al. (1998).

Instance	Algorithm	Best Energy (Kj)	Gap Best Energy (%)	Avg. Energy (Kj)	Gap Avg. Energy (%)
Golden_1	ILS-SP-SOA [20]	1451439,72	0,00	1451439,74	0,00
	Free Ant + K-Means	1794485,33	23,63	1839994,09	26,77
	Free Ant + K-Medoids	1794241,83	23,62	1840565,07	26,81
	Restricted Ant + K-Means	1816835,04	25,17	1848875,53	27,38
	Restricted Ant + K-Medoids	1947157,52	34,15	1963021,21	35,25
Golden_2	ILS-SP-SOA [20]	2736086,81	0,00	2736274,33	0,00
	Free Ant + K-Means	3392031,91	23,97	3543677,21	29,51
	Free Ant + K-Medoids	3428805,74	25,32	3505641,92	28,12
	Restricted Ant + K-Means	3599353,15	31,55	3673938,84	34,27
	Restricted Ant + K-Medoids	3437439,70	25,63	3490109,42	27,55
Golden_3	ILS-SP-SOA [20]	4527654,42	0,00	4527654,48	0,00
	Free Ant + K-Means	6188152,51	36,67	6419673,13	41,79
	Free Ant + K-Medoids	6283131,21	38,77	6454124,05	42,55
	Restricted Ant + K-Means	6557807,89	44,84	6857022,53	51,45
	Restricted Ant + K-Medoids	6562403,80	44,94	6661796,57	47,14
Golden_4	ILS-SP-SOA [20]	6146280,48	0,00	6150983,46	0,00
	Free Ant + K-Means	8605180,34	40,01	9012942,71	46,53
	Free Ant + K-Medoids	8765434,46	42,61	9035752,79	46,90
	Restricted Ant + K-Means	9079222,64	47,72	9268418,84	50,68
	Restricted Ant + K-Medoids	8984686,99	46,18	9165456,57	49,01
Golden_5	ILS-SP-SOA [20]	2742484,44	0,00	2742484,44	0,00
	Free Ant + K-Means	3197097,79	16,58	3546780,56	29,33
	Free Ant + K-Medoids	3413564,70	24,47	3549348,08	29,42
	Restricted Ant + K-Means	3240939,18	18,18	3289025,29	19,93
	Restricted Ant + K-Medoids	3884274,63	41,63	3957098,38	44,29
Golden_6	ILS-SP-SOA [20]	3352282,80	0,00	3352707,87	0,00
	Free Ant + K-Means	4425352,82	32,01	4707693,42	40,41
	Free Ant + K-Medoids	4614756,65	37,66	4710028,07	40,48
	Restricted Ant + K-Means	4572758,83	36,41	4639142,62	38,37
	Restricted Ant + K-Medoids	4668692,11	39,27	4718943,21	40,75
Golden_7	ILS-SP-SOA [20]	4205444,71	0,00	4205444,72	0,00
	Free Ant + K-Means	5807380,97	38,09	6021276,07	43,18
	Free Ant + K-Medoids	5781736,01	37,48	5922564,39	40,83
	Restricted Ant + K-Means	6332893,46	50,59	6483549,56	54,17
	Restricted Ant + K-Medoids	6172468,04	46,77	6273870,76	49,18
Golden_8	ILS-SP-SOA [20]	4828825,54	0,00	4830980,29	0,00
	Free Ant + K-Means	6536564,26	35,37	6750035,09	39,72
	Free Ant + K-Medoids	6511290,55	34,84	6723972,49	39,18
	Restricted Ant + K-Means	6725806,14	39,28	6849882,83	41,79
	Restricted Ant + K-Medoids	6772409,10	40,25	6899067,29	42,81
Golden_9	ILS-SP-SOA [20]	302127,64	0,00	302320,91	0,00
	Free Ant + K-Means	354352,28	17,29	368537,73	21,90
	Free Ant + K-Medoids	395652,02	30,96	397140,06	31,36
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_10	ILS-SP-SOA [20]	384069,09	0,00	384217,14	0,00
	Free Ant + K-Means	469330,80	22,20	476837,93	24,11
	Free Ant + K-Medoids	510679,39	32,97	512969,95	33,51
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_11	ILS-SP-SOA [20]	476743,98	0,00	476933,1	0,00
	Free Ant + K-Means	589171,40	23,58	606887,14	27,25
	Free Ant + K-Medoids	636908,70	33,60	640218,25	34,24
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_12	ILS-SP-SOA [20]	576582,81	0,00	579327,34	0,00
	Free Ant + K-Means	723836,53	25,54	737304,75	27,27
	Free Ant + K-Medoids	774969,45	34,41	780308,23	34,69
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_13	ILS-SP-SOA [20]	447899,37	0,00	448193,11	0,00
	Free Ant + K-Means	502177,89	12,12	510544,17	13,91
	Free Ant + K-Medoids	501018,13	11,86	511451,93	14,11
	Restricted Ant + K-Means	626314,59	39,83	645567,40	44,04
	Restricted Ant + K-Medoids	561491,28	25,36	562993,32	25,61
Golden_14	ILS-SP-SOA [20]	566803,22	0,00	567382,89	0,00
	Free Ant + K-Means	-	-	-	-
	Free Ant + K-Medoids	-	-	-	-
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_15	ILS-SP-SOA [20]	699609,07	0,00	700357,75	0,00
	Free Ant + K-Means	826239,20	18,10	840338,91	19,99
	Free Ant + K-Medoids	902635,42	29,02	913921,38	30,49
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-

TABLE 8. (Continued.) Comparison results of Golden et al. (1998).

Golden_16	ILS-SP-SOA [20]	845985,34	0,00	846679,46	0,00
	Free Ant + K-Means	1089105,77	28,74	1121642,41	32,48
	Free Ant + K-Medoids	1157229,24	36,79	1182172,87	39,62
	Restricted Ant + K-Means	-	-	-	-
	Restricted Ant + K-Medoids	-	-	-	-
Golden_17	ILS-SP-SOA [20]	80319,35	0,00	80393,13	0,00
	Free Ant + K-Means	78001,30	-2,89	80511,92	0,15
	Free Ant + K-Medoids	79682,76	-0,79	81071,13	0,84
	Restricted Ant + K-Means	89158,72	11,01	89896,18	11,82
	Restricted Ant + K-Medoids	93314,59	16,18	94060,09	17,00
Golden_18	ILS-SP-SOA [20]	111317,98	0,00	111393,13	0,00
	Free Ant + K-Means	116825,42	4,95	118675,91	6,54
	Free Ant + K-Medoids	118110,08	6,10	119299,53	7,10
	Restricted Ant + K-Means	131109,93	17,78	132337,36	18,80
	Restricted Ant + K-Medoids	141625,62	27,23	144933,18	30,11
Golden_19	ILS-SP-SOA [20]	150845,06	0,00	151155,2	0,00
	Free Ant + K-Means	157472,61	4,39	160434,10	6,14
	Free Ant + K-Medoids	156511,52	3,76	158588,24	4,92
	Restricted Ant + K-Means	184050,35	22,01	185386,21	22,65
	Restricted Ant + K-Medoids	194379,10	28,86	196016,43	29,68
Golden_20	ILS-SP-SOA [20]	198976,65	0,00	199052,22	0,00
	Free Ant + K-Means	236472,51	18,84	238312,03	19,72
	Free Ant + K-Medoids	238814,66	20,02	241505,20	21,33
	Restricted Ant + K-Means	239991,48	20,61	242172,45	21,66
	Restricted Ant + K-Medoids	264870,88	33,12	268010,77	34,64

Again, we observed an increase in the average Gap value for both algorithms. However, it is important to note that the increase in Restricted Ant is considerably smaller than the increase experienced by Free Ant in relation to the previous set of instances. This difference suggests that the Restricted Ant can operate more efficiently with a smaller number of iterations, while Free Ant requires a higher number of iterations as the problem size increases.

Finally, we encountered a limitation regarding the scalability of Free Ant in comparison to Restricted Ant. In the case of the former, both the Gap and the total algorithm execution time, begin to increase as we attempt to solve problems with a greater number of inputs (i.e., a higher number of nodes). These issues are not reflected in the same manner in the Restricted Ant algorithms, which prove to be more efficient with larger-scale instances.

Another limitation of the proposed algorithms (Free Ant and Restricted Ant) is that, in order to operate efficiently, a proper parameter configuration must be established. This entails a preliminary experimentation process to determine the configuration that yields the best results for a specific instance.

VII. CONCLUSION

Combinatorial optimization is a complex and challenging field in which significant advances have been achieved through operations research, applied mathematics, and computer science research. In this work, two hybrid ant colony-based algorithms with clustering techniques, namely Free Ant and Restricted Ant, were presented for solving the Energy Minimizing Vehicle Routing Problem (EMVRP), a “green”-oriented variant of the Vehicle Routing Problem (VRP). These algorithms has applications in green logistics and distribution operations.

The results obtained in an extensive experimentation phase indicate that the algorithms proposed in this work are capable of consistently generating high-quality solutions. In particular, the results show that the Free Ant algorithm outperformed its counterpart, Restricted Ant, in all scenarios, while the latter demonstrated greater robustness and balance in medium and large-scale instances, primarily due to its superior runtime efficiency. This may be attributed to the advantages of the clustering algorithms used, which allowed Restricted Ant to find higher-quality routes in a much more confined search space. It is important to note that the performance of the algorithms should be tested on a larger number of instances and iterations to draw more conclusive conclusions.

While the results are promising, it is important to highlight that there is still room for improvement and further research is needed to achieve more efficient solutions. Green VRP problems are still relatively unexplored, leaving an ample scope for further investigations.

Nonetheless, both algorithms are innovative and utilize techniques that have not been extensively explored together in the resolution of combinatorial optimization problems. The proposed algorithms are novel in how they use clustering techniques to guide the search of the ACO metaheuristic. One of the novelties is the combined use of two clustering algorithms (K-Means with capacity-constrained and K-Medoids with capacity-constrained) with the ant colony system. Additionally, the Free Ant algorithm is a completely new proposal to the best of the authors’ knowledge. Future work includes considering the reuse of clustering algorithms in later stages after initialization, using active re-clustering techniques to modify the ants’ search space. It is believed that this new approach will help the Restricted Ant algorithm improve its results, as it tends to

converge to worse solutions than Free Ant due to the cluster shapes.

Furthermore, it would be interesting to investigate the combined use of these algorithms with other machine learning techniques to improve the self-regulation of the ACO system's hyperparameters, which are highly variable-dependent. Additionally, the implementation of a parallelization system for deploying ants in the iterations of Free Ant and Restricted Ant should be considered.

One of the main challenges researchers in this field face is the lack of documented instances for benchmarking. Currently, there are only a limited number of standardized and widely used instance sets, such as those available in CVRPLIB. However, these solutions do not always expose the optimal solution for the EMVRP model, where the goal is to minimize the total energy consumed. Therefore, more research and collaboration are needed to develop additional instances and optimal solutions for such problems.

In summary, the proposed algorithms were able to solve EMVRP problems and achieved good results in the instances from CVRPLIB where they were tested. However, although the algorithms did not reach the best results in all instances, this work lays the foundation for future research and improvements in the use of these metaheuristic techniques. The proposed algorithms are innovative in the efficient integration of clustering techniques and the ACO metaheuristic.

REFERENCES

- [1] I. Kara, B. Y. Kara, and M. K. Yetis, "Energy minimizing vehicle routing problem," in *Combinatorial Optimization and Applications* (Lecture Notes in Computer Science), vol. 4616, A. Dress, Y. Xu, and B. Zhu, Eds. Berlin, Germany: Springer, 2007, doi: [10.1007/978-3-540-73556-4_9](https://doi.org/10.1007/978-3-540-73556-4_9).
- [2] O. Cordon, I. F. D. Viana, F. Herrera, and L. Moreno, "A new ACO model integrating evolutionary computation concepts: The best-worst ant system," in *Proc. Ants*, 2000, pp. 22–29.
- [3] S. Geetha, G. Poonthali, and P. T. Vanathi, "Improved k-means algorithm for capacitated clustering problem," *INFOCOMP J. Comput. Sci.*, vol. 8, no. 4, pp. 52–59, 2009. [Online]. Available: <https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/282>
- [4] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: [10.1109/3477.484436](https://doi.org/10.1109/3477.484436).
- [5] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997, doi: [10.1109/4235.585892](https://doi.org/10.1109/4235.585892).
- [6] B. Bullnheimer, R. Hartl, and C. Strauß, "Working papers SFB adaptive information systems and modelling in economics and management science," WU Vienna Univ. Economics Bus., Working Paper, Apr. 1997.
- [7] T. Stützle and H. Hoos, "Improvements on the ant-system: Introducing the MAX-MIN ant system," in *Artificial Neural Nets and Genetic Algorithms*. Vienna, Austria: Springer, 1998, doi: [10.1007/978-3-7091-6492-1_54](https://doi.org/10.1007/978-3-7091-6492-1_54).
- [8] F.-Y. Wang and D. Liu, *Advances in Computational Intelligence*. World Scientific, 2006. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/6072>
- [9] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982, doi: [10.1109/TIT.1982.1056489](https://doi.org/10.1109/TIT.1982.1056489).
- [10] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab., Statist.*, vol. 1. Berkeley, CA, USA: University of California Press, Berkeley, 1967, pp. 281–297. [Online]. Available: <http://projecteuclid.org/euclid.bsmsp/1200512992>
- [11] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program PAM)," in *Finding Groups in Data*, L. Kaufman and P. J. Rousseeuw, Eds. Wiley, 1990, ch. 2, pp. 68–125. [Online]. Available: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470316801.ch2>, doi: [10.1002/9780470316801.ch2](https://doi.org/10.1002/9780470316801.ch2).
- [12] P. Hansen and N. Mladenovic, "Variable neighborhood search," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds. Boston, MA, USA: Springer, 2005, doi: [10.1007/0-387-28356-0_8](https://doi.org/10.1007/0-387-28356-0_8).
- [13] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, Aug. 1964. [Online]. Available: <http://www.jstor.org/stable/167703>
- [14] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, vol. 8, Philadelphia, PA, USA. New Orleans, LA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: <https://www.bibsonomy.org/bibtex/2553bbfa74b13c47b4e9c7c0034a8406e/sdo>, doi: [10.1145/1283383.1283494](https://doi.org/10.1145/1283383.1283494).
- [15] B. Bullnheimer, R. F. Hartl, and C. Strauss, "Applying the ANT system to the vehicle routing problem," in *Meta-Heuristics*, S. Voß, S. Martello, I. H. Osman, and C. Roucairol, Eds. Boston, MA, USA: Springer, 1999, doi: [10.1007/978-1-4615-5775-3_20](https://doi.org/10.1007/978-1-4615-5775-3_20).
- [16] A. E. Rizzoli, R. Montemanni, E. Lucibello, and L. M. Gambardella, "Ant colony optimization for real-world vehicle routing problems," *Swarm Intell.*, vol. 1, pp. 135–151, Sep. 2007, doi: [10.1007/s11721-007-0005-x](https://doi.org/10.1007/s11721-007-0005-x).
- [17] S. S. El-Khatib, Y. A. Skobtsov, and S. I. Rodzin, "Theoretical and experimental evaluation of hybrid ACO-k-means image segmentation algorithm for MRI images using drift-analysis," *Proc. Comput. Sci.*, vol. 150, pp. 324–332, Jan. 2019, doi: [10.1016/j.procs.2019.02.059](https://doi.org/10.1016/j.procs.2019.02.059).
- [18] N. Kusumahardhini, G. F. Hertono, and B. D. Handari, "Implementation of K-means and crossover ant colony optimization algorithm on multiple traveling salesman problem," *J. Phys., Conf. Ser.*, vol. 1442, no. 1, Jan. 2020, Art. no. 012035, doi: [10.1088/1742-6596/1442/1/012035](https://doi.org/10.1088/1742-6596/1442/1/012035).
- [19] G. W. H. H. P. Rathnakumara and T. D. Rupasinghe, "Clustering-based augmented Tabu search for energy minimizing vehicle routing problem (EMVRP)," in *Proc. 12th Annu. Sessions*. Nugegoda, Sri Lanka: Sri Lanka Association for Artificial Intelligence, 2017, p. 77.
- [20] R. Kramer, A. Subramanian, T. Vidal, and L. D. A. F. Cabral, "A matheuristic approach for the pollution-routing problem," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 523–539, Jun. 2015, doi: [10.1016/j.ejor.2014.12.009](https://doi.org/10.1016/j.ejor.2014.12.009).



NICOLÁS FRÍAS received the Graduate degree from Universidad de Playa Ancha, Valparaíso, Chile. He is currently a computer engineer in Chile. With a specialization in information management, he has built a successful career as a software developer. He stands out for his ability to merge theoretical concepts with practical applications.



FRANKLIN JOHNSON received the B.E. degree, the master's degree in informatics, and the Ph.D. degree from Pontificia Universidad Católica de Valparaíso, in 2006, 2010, and 2017, respectively. He is currently the Head of the Department of Computer Sciences, Universidad de Playa Ancha, Valparaíso, Chile. His research interests include metaheuristics, autonomous search, and combinatorial optimization. His research interests include metaheuristics, autonomous search, and combinatorial optimization.



CARLOS VALLE received the Ph.D. degree in computer science from Universidad Técnica Federico Santa María, Chile, in 2014. He is currently with the Department of Computer Science and Informatics, Universidad de Playa Ancha, Valparaíso, Chile. From 2014 to 2018, he was a Researcher with the Computer Science Department, Universidad Técnica Federico Santa María. Since 2018, he has been an Associate Professor with the Department of Computer Science and Informatics, Universidad de Playa Ancha, Chile. His research interests include machine learning, ensemble learning, deep neural networks, and pattern recognition with applications to time series.

...