AppliedMath

MDPI

Article

Genetic Algorithm Optimization of Sales Routes with Time and Workload Objectives

Filipa Costa [1,2], Margarida Brito [1], Pedro Louro [2] and Sílvio Gama [1,*]

[1] Mathematics Centre of the Porto University (CMUP), Department of Mathematics, Science Faculty, University of Porto, 4169-007 Porto, Portugal; filipamc19@gmail.com (F.C.); mabrito@fc.up.pt (M.B.)
[2] Nors, R. Manuel Pinto de Azevedo 711 1º, 4149-010 Porto, Portugal; plouro@nors.com
* Correspondence: smgama@fc.up.pt

Abstract

This work proposes a novel multi-objective genetic algorithm to solve the Periodic Vehicle Routing Problem with Time Windows (PVRPTWs) tailored for sales teams with diverse geographic scales and visit frequency requirements. Unlike existing models, our approach incorporates workload balancing and applies a clustering-based preprocessing step for long-distance routes using multidimensional scaling and fuzzy clustering, improving initial route grouping. When tested on three salesperson profiles (short-, mid-, and long-distance), the model achieved up to a 69% reduction in total travel time compared to a nearest neighbor baseline. These results demonstrate substantial improvements over existing methods and underscore the model's flexibility and potential for extension to dynamic or real-time sales routing applications.

Keywords: fuzzy clustering; genetic algorithm; multi-objective optimization; periodic vehicle routing problem with time windows; route optimization

check for updates

Received: 21 June 2025
Revised: 31 July 2025
Accepted: 4 August 2025
Published: 11 August 2025

Citation: Costa, F.; Brito, M.; Louro, P.; Gama, S. Genetic Algorithm Optimization of Sales Routes with Time and Workload Objectives. *AppliedMath* **2025**, *5*, 103. https://doi.org/10.3390/appliedmath5030103

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In Canada, a substantial portion of the market for the sales, rental, and after-sales services of heavy construction equipment is covered by operations that extend across more than half the national territory. Managing such a wide geographical area poses considerable logistical challenges, particularly for commercial teams that must travel frequently to serve clients dispersed across distant regions.

In this context, optimizing sales route planning becomes a critical opportunity to enhance operational efficiency. Effective planning can significantly reduce travel time and costs while also contributing to better service quality and stronger client relationships.

Efficient route planning has become increasingly important for companies whose operations rely on frequent client interactions. Sales teams, in particular, often struggle to organize their routes in a way that balances time and client satisfaction. Without a structured approach, scheduling is typically based on individual intuition, which can result in inefficient travel patterns, the under-servicing of valuable clients, and missed opportunities for engagement.

Recent advances in route optimization, particularly within the context of the Vehicle Routing Problem (VRP), have provided companies with tools to improve logistic efficiency and reduce operational costs [1,2]. Among these, the Periodic Vehicle Routing Problem with Time Windows (PVRPTWs) is a prominent model for structuring recurring visits across multiple days while respecting constraints such as time windows and visit frequency [3].

This work aims to develop an optimization model that considers both the varying importance of clients and the distinct profiles of salespeople. Each client is grouped according to their significance. Based on these groups, the required number of visits within a specific time interval is determined. In large sales teams, there are typically different types of salespeople, each with distinct client distribution patterns. Some salespeople work with clients located in urban areas, where clients are relatively close to each other, minimizing travel time and allowing more time for client visits. Others have clients in areas that lie between urban and rural regions, resulting in greater distances between clients and an increased need for travel time. Finally, some salespeople are assigned clients in very remote areas, where clients are far apart, and daily travel to and from home is not feasible. For this reason, a different optimization model is required for these long-distance salespeople. In this work, the salespeople are categorized as short-distance, mid-distance, and long-distance based on the spatial distribution of their clients.

All computational work was performed using the Python 3.13 programming language. A variety of libraries supported different aspects of the workflow, ranging from data manipulation and visualization to the formulation and solution of optimization models. Key packages include Pandas 2.2.3 [4], Matplotlib 3.10.0 [5], Scikit-learn 1.5.2 [6], and DEAP 1.4.1 [7]. All computation times reported in this work were obtained on a MacBook with an M1 chip and 8 GB RAM.

Section 2 reviews clustering-related ideas used to address the challenge of long-distance salespeople. Section 3 explains how the problem can be framed as a PVRPTW and presents relevant optimization methods to solve this type of problem, including a brief overview of genetic algorithms. Section 4 introduces a hybrid multi-objective genetic algorithm designed to solve a realistic PVRPTW by incorporating client importance, salesperson heterogeneity, and spatial clustering within a unified metaheuristic framework. The mathematical formulation of the problem is also provided. Section 5 presents the results for each salesperson profile, and Section 6 discusses these findings.

2. Clustering

Clustering is a technique that can provide valuable input for optimization methods, especially in scenarios where clients are widely dispersed. It is a technique from unsupervised machine learning used to group data points based on their similarity. Unlike supervised learning, it does not rely on predefined labels; instead, it identifies natural patterns or structures within the data. The goal is to assign observations to clusters such that those within the same cluster are more similar to each other than to those in different clusters.

This process typically depends on a similarity measure, most often a distance metric like Euclidean distance, to evaluate how close or related data points are. These similarity scores guide the algorithm in forming coherent and meaningful clusters [8].

2.1. Fuzzy C-Means

Fuzzy clustering, also known as soft clustering, is a method that allows each data point to belong to multiple clusters simultaneously, with varying degrees of membership. A commonly used method in soft clustering is fuzzy C-means, which was originally introduced by Jim Bezdek in 1981 [9]. Fuzzy C-means iteratively assigns to each data point a degree of membership to every cluster by optimizing the following objective function:

$$J(U, C) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \tag{1}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of point x_i in the cluster j, and c_j is the centre of this cluster. Firstly, the membership matrix $U = [u_{ij}]$

is initialized randomly, ensuring the memberships of each data point sum to 1 across all clusters. Afterwards, the cluster centers c_j are updated as weighted means of the data points:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}. \tag{2}$$

Then, the membership degrees u_{ij} are also updated using

$$u_{ij} = \left(\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}} \right)^{-1}. \tag{3}$$

These last two steps are repeated until the changes in the objective function J, (or in the membership values) become smaller than a predefined threshold [10].

2.2. Clustering Geographic Coordinates

Clustering methods like fuzzy C-means clustering require that each data point has numeric coordinates (features), and that a well-defined distance measure (such as Euclidean distance) can be used to compute both the positions of cluster centroids and the degree to which each data point is associated with each cluster. While it might seem reasonable to use a travel time matrix as input, this approach poses several challenges. A time matrix only provides pairwise dissimilarities without explicit coordinates, which prevents centroid computation. Additionally, the distances may violate essential properties such as the triangle inequality, making techniques like multidimensional scaling ineffective for converting them into usable Euclidean space [11,12].

For clustering geographic locations, accounting for Earth's curvature is essential. Instead of relying on simple Euclidean distances, which assume a flat surface, it uses the Haversine distance, which computes great-circle distances based on latitude and longitude [13].

To further adapt geographic data for clustering, multidimensional scaling (MDS) can be applied. Although geographic coordinates are two-dimensional, Haversine distances are not Euclidean. MDS transforms these into a 2D Euclidean space where distances approximate the original ones, making them suitable for fuzzy C-means [14].

In this work, the *Scikit-learn* implementation of MDS is used, which employs the SMACOF algorithm, an iterative optimization technique that minimizes a stress function:

$$\text{Stress}(X) = \sum_{i<j} \left(d_{ij}(X) - \delta_{ij} \right)^2, \tag{4}$$

where $d_{ij}(X)$ is the Euclidean distance between the embedded points i and j, and δ_{ij} is the original dissimilarity between the two points, calculated as their Haversine distance:

$$\delta_{ij} = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_j - \phi_i}{2} \right) + \cos(\phi_i)\cos(\phi_j)\sin^2\left(\frac{\lambda_j - \lambda_i}{2} \right)} \right). \tag{5}$$

Here, ϕ_i, λ_i and ϕ_j, λ_j represent the latitudes and longitudes (in radians) of points i and j, and $r = 6371$ km is the radius of the Earth [15]. The algorithm refines point positions iteratively to minimize stress, enacting multiple runs from different initial configurations and retaining the result with the lowest stress [16].

To help illustrate the role of δ_{ij} and d_{ij} in the stress function, and to clarify the transformation performed by MDS, Figure 1 provides a schematic comparison between the original geographic space and the resulting 2D embedded space. In the original space, the dissimilarities δ_{ij} correspond to geodesic (Haversine) distances over the Earth's surface. After applying MDS, these curved distances are approximated by straight-line Euclidean

distances $d_{ij}(X)$ in a flat 2D space. The "stress" measures how well these straight-line distances preserve the original geodesic structure.

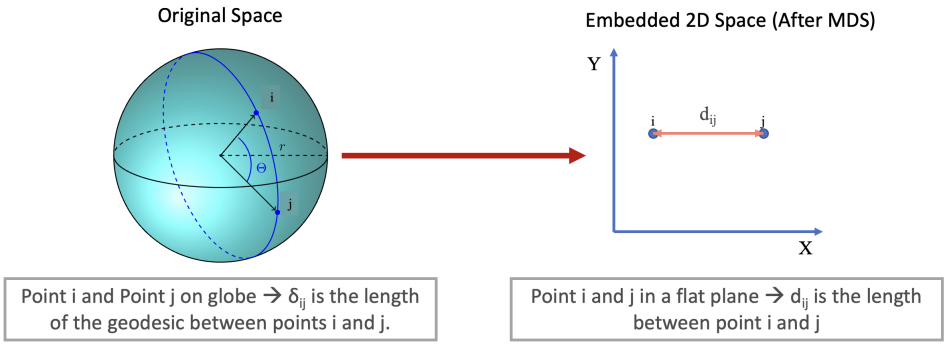


Figure 1. Illustration of the MDS transformation. **Left**: Original space with points A and B on the Earth's surface, where δ_{ij} represents their geodesic (Haversine) distance. **Right**: The same points embedded in a 2D Euclidean space, where d_{ij} is their straight-line distance. The "stress" minimized during MDS reflects the discrepancy between these two distances across all point pairs.

3. Periodic Vehicle Routing Problem with Time Windows

3.1. Definition of the Problem

The Periodic Vehicle Routing Problem with Time Windows (PVRPTWs) is a challenging logistics optimization problem. It requires designing routes for vehicles to serve a group of customers across multiple days while ensuring visits occur within specific time windows. This problem is characterized by several key features. First, the planning horizon spans multiple days, requiring the routes to be scheduled over several days. Second, each customer must be serviced within a designated time window. Third, customers have predetermined sets of valid combinations of visits and periods, known as patterns.

The main objective of the PVRPTWs is to minimize total transportation costs or time across the entire planning horizon while satisfying all customer needs and complying with operational constraints [17]. A simplified representation of the PVRPTWs is illustrated in Figure 2.

The PVRPTWs is represented using a complete directed graph $G = (V, A)$, where $V = \{0, 1, \ldots, n\}$ represents the set of vertices, and $A = \{(i, j) | i, j \in V, i \neq j\}$ represents the set of arcs. The planning horizon is defined as a set of t days, represented as $T = \{1, 2, \ldots, t\}$. Vertex 0 represents the depot, which has a time window $[a_0, b_0]$ and serves as the base for the salespeople. The vertices i correspond to the customers and the depot. Each customer i is characterized by a service duration $d_i \geq 0$, a time window $[a_i, b_i]$, a required service frequency f_i, and a set $C_i \subseteq T$ that specifies the allowed combinations of days for visits. Every arc $(i, j) \in A$ is associated with a travel time or cost $c_{ij} > 0$, representing the effort required to move between the respective vertices [18].

In this work, to solve this problem, a complete directed graph is constructed for each salesperson. It is considered that the depot (vertices 0 and $N_c + 1$) corresponds to the salespeople's homes, and it is considered to operate within a time window of [8 a.m., 7 p.m.]. It is also considered that each customer has a uniform time window of [9 a.m., 4 p.m.] and service duration of $d_i = 30$ min, reflecting typical visit times, as specified by the sales team manager. This value was used for illustration, but the model supports any service duration as appropriate for the application context. The service frequency f_i and the allowed visit-day combinations C_i for each customer are determined by their assigned group. This grouping enables the identification of higher-value clients, who are consequently allocated a greater number of visits. During this time interval, clients in frequency category 1 have monthly visits, with six being in-person visits; clients from frequency category 2 are visited

every 3 months, with two being in-person, and those from category 3 are visited twice a year, involving one in-person visit. This last constraint forces the optimization process to span the entire year since the optimization must account for which semester is more efficient for conducting the in-person visit. Moreover, short- and mid-distance salespeople conduct in-person visits on only 3 days of the week, reserving the remaining days for remote visits.

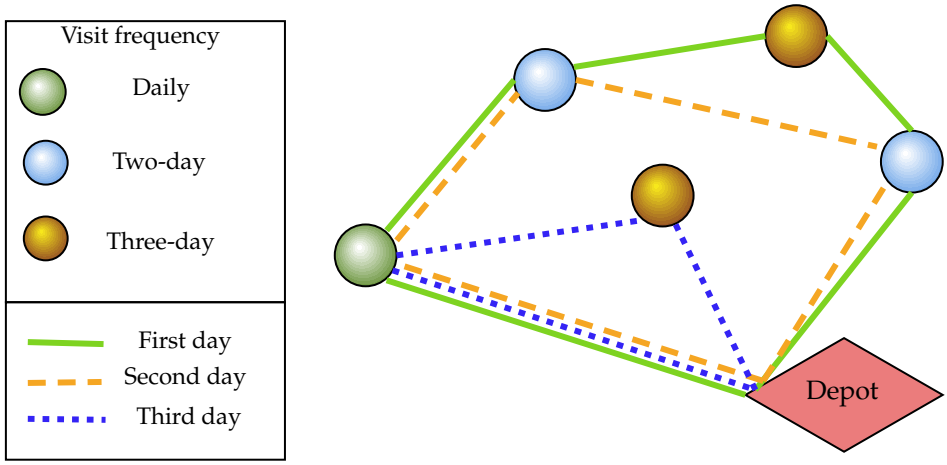


Figure 2. Illustrative representation of a general PVRPTWs solution over a three-day planning horizon. Clients are color-coded by visit frequency: green (daily), blue (twice per period), and brown (three times). Routes for each day are represented by different line styles—solid (first day), dashed (second), and dotted (third)—connecting clients to the depot (red diamond). This figure demonstrates how visits are distributed to meet frequency requirements while optimizing route efficiency.

3.2. Optimization Methods to Solve the PVRPTWs

Solving the PVRPTWs problem can be addressed using a range of optimization methods. These methods span from exact approaches [19], which guarantee finding the optimal solution by exploring the entire solution space, to metaheuristic techniques [20], which provide near-optimal solutions more efficiently by navigating the solution space strategically. The choice of the method depends on the size of the problem, the computational resources, and the quality of the desired solution.

The PVRPTWs is classified as an NP-complete problem with an estimated exponential time complexity [21], meaning that finding an optimal solution becomes increasingly challenging as the problem size grows. In particular, solving instances with 100 or more customers to optimality is highly complex [22]. Consequently, metaheuristic approaches are explored to address this challenge.

3.2.1. State of the Art

Metaheuristic techniques are particularly valuable for tackling complex optimization problems with large solution spaces and limited prior knowledge about optimal solutions. They iteratively explore and refine candidate solutions, making them effective in high-dimensional or nonlinear contexts where traditional methods struggle [23]. Various metaheuristics have been successfully applied to solve the PVRPTWs.

One of the earliest versions solves PVRPTWs using a unified tabu search heuristic that efficiently handles both the periodic scheduling of customer visits and the routing of vehicles within specific time windows [24]. The unified tabu search method applies specialized neighborhood moves to reassign customers between routes and days, uses adaptive memory to guide the search, and allows temporary constraint violations with penalties to explore more solutions.

Simulated annealing is a method inspired by the heating and cooling process in metallurgy. It starts with an initial solution and high temperature to allow exploration of the solution space, even accepting worse solutions to escape local optima. As the algorithm progresses, the temperature gradually decreases, reducing the probability of accepting worse solutions and focusing on improving the best solutions found [25]. This method has been applied to solve the Periodic Capacitated Vehicle Routing Problem in [26] and also to solve Vehicle Routing Problems with Time Windows in [27]. These studies demonstrate that simulated annealing can also be applied to solve the PVRPTWs.

In [28], the PVRPTWs is addressed using a hybrid genetic algorithm that integrates genetic operators with local search and repair strategies. This approach simultaneously optimizes customer scheduling and vehicle routing while effectively managing time window and capacity constraints. In [29], a hybrid generational genetic algorithm that combines specialized genetic operators with local search to optimize and repair solutions is also applied. Each individual encodes visit schedules and daily routes, with local search enhancing quality and correcting constraint violations. The fact that this method does not rely only on problem-specific knowledge or assumptions makes it applicable to a wide range of complex, nonlinear, or high-dimensional optimization problems, being the most unbiased and versatile of the three methods.

3.2.2. Genetic Algorithms

Genetic Algorithms (GAs) are inspired by Darwin's theory of evolution by means of natural selection, where survival and reproduction depend on natural selection and beneficial mutations. These principles are applied in GAs to optimize complex problems by mimicking biological evolution. GAs evolve solutions through selection, crossover, and mutation, enabling them to explore large, high-dimensional solution spaces effectively [30].

In GAs, potential solutions are represented as chromosomes, typically encoded as binary strings or real numbers. The fitness of each chromosome is evaluated using a fitness function, guiding selection for reproduction. Selection methods, like a roulette wheel or tournament selection, determine which chromosomes will produce offspring. The crossover operator combines two parent chromosomes to create new solutions, while mutation introduces small random changes to maintain diversity and avoid premature convergence. These operators work together to evolve better solutions through generations [31].

Designing a GA involves decisions on encoding solutions, population size, selection methods, crossover and mutation operators, and termination criteria. The process generally includes initialization, evaluation, selection, crossover, mutation, and replacement, continuing until the stopping criteria are met.

For problems with multiple objectives, multi-objective genetic algorithms (MOGAs) are used. These algorithms generate a set of solutions that offer trade-offs between objectives, represented by the Pareto front. In MOGAs, a Pareto solution dominates another if it is at least as good in all objectives and strictly better in at least one. The Pareto front consists of all Pareto-optimal solutions, representing the best trade-offs [32], as visualized in the combined schematic shown in Figure 3, which integrates the ranking of solutions by Pareto dominance and the selection of the best solution using the utopian point.

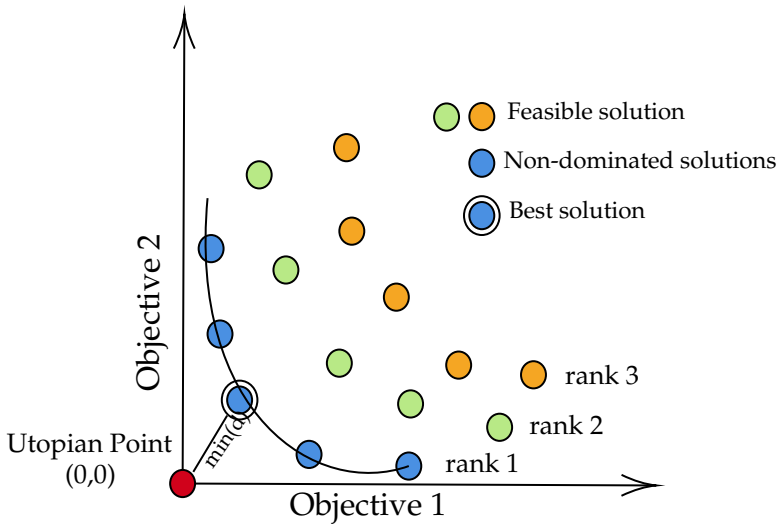


Figure 3. Illustration of multi-objective optimization using a genetic algorithm, showing the ranking of feasible solutions based on Pareto dominance (color-coded by ranks 1 to 3) and the identification of non-dominated solutions. The utopian point $(0, 0)$ is used as a reference to select the best solution—highlighted by a circle—as the one minimizing the Euclidean distance to this ideal point.

NSGA-II is a widely used MOGA that sorts solutions into Pareto fronts. Solutions are ranked based on dominance and crowding distance, with the aim of maintaining diversity. The algorithm ensures that solutions from less crowded regions of the Pareto front are preferred, preserving a variety of trade-off solutions [33].

4. Routing Optimization Model

Figure 4 illustrates the overall workflow of the proposed approach, highlighting the key stages from data preprocessing and clustering to optimization and evaluation of multi-objective genetic algorithms.

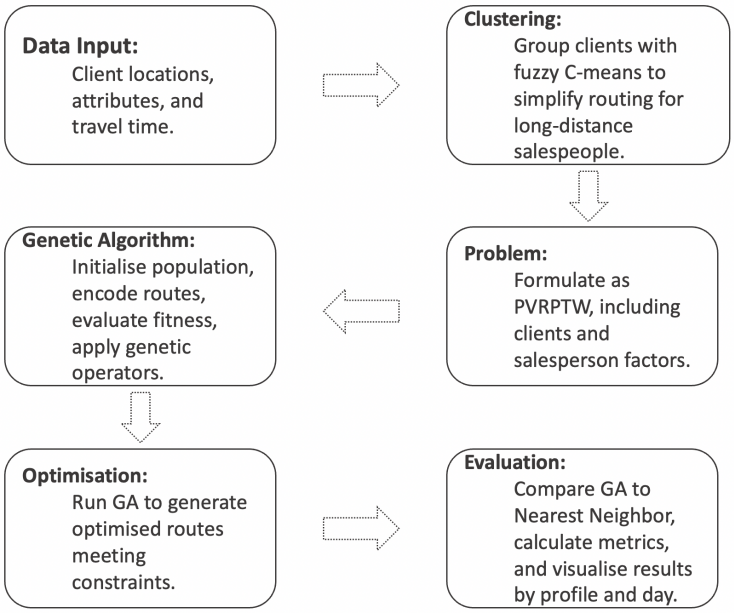


Figure 4. Workflow of the proposed hybrid multi-objective genetic algorithm approach, highlighting the integration of clustering, heterogeneous salesperson modeling, and multi-criteria optimization.

4.1. Hybrid Multi-Objective Genetic Algorithm for Diverse Salesperson and Client Profiles

This work develops a hybrid MOGA to address a realistic PVRPTWs, characterized by heterogeneous clients and diverse salesperson profiles. The aim of the problem is to

reduce the total travel time for improved efficiency while also minimizing daily travel time variability to ensure balanced workloads.

Clients are grouped by importance, which determines their required visit frequency over the planning horizon. This prioritization ensures that high-value customers receive an appropriate number of visits.

Salespeople are categorized into short-, mid-, and long-distance profiles based on their clients' spatial distribution. This categorization captures differences in travel constraints and scheduling flexibility. For long-distance salespeople, spatial clustering is applied to group geographically close clients, improving route coherence and reducing travel costs.

The GA evolves a population of candidate solutions encoding visit schedules and daily routes. It integrates specialized genetic operators with two local improvement methods: 2-opt local search, which optimizes route structure by swapping edges, and simulated annealing, which accepts worse solutions probabilistically to escape local optima.

This hybrid approach balances exploration and exploitation, effectively navigating the solution space and generating high-quality, feasible routes that satisfy time windows, visit frequencies, and capacity constraints.

By incorporating client importance, salesperson heterogeneity, and clustering within a hybrid metaheuristic framework, the proposed algorithm advances existing methods and is well-suited for complex, large-scale applications.

4.2. Mathematical Modeling of the Problem

The mathematical formulation of the problem is as follows. In this model, a single salesperson visits a group of customers over a planning horizon of N_k working days. To construct the model, the following sets are defined:

$Cc = \{1, 2, \cdots, Nc\}$—Set of customers;
$Cd = \{0, Nc + 1\}$—Set of depots (start and end);
$C = Cc \cup Cd$—Full set of nodes (customers and depot);
$C_1 \subset C_c$—Set of customers visited monthly, with six in-person visits;
$C_2 \subset C_c$—Set of customers visited four times a year, with two being in person;
$C_3 \subset C_c$—Set of customers visited two times a year, with one being in person;
$K = \{1, \cdots, N_k\}$—Set of planning days.

The following parameters are also defined:

t_{ij}—Travel time between nodes i and j;
$[a_i, b_i]$—Time window within which service at node i is permitted;
H—Salesperson's working hours per day;
H_{in}—Earliest allowable start time for a visit;
$s(i)$— Visit duration at customer i;
$M = 10^7$—A sufficiently large constant used for constraint linearization via the big-M method [34].

To address the problem, the decision variable is defined as follows:

x_{kij}—Binary variable: A value of 1 if, on day k, the salesperson visits customer j immediately after visiting customer i; otherwise, the value is 0.

Another variable is also introduced to ensure that the pre-established visit frequency is met:

y_{ki}—binary variable: A value of 1 if the salesperson visits customer i on day k; otherwise, the value is 0.

Finally, a time variable is defined:

w_{ik} , $\quad i \in C$, $\quad k \in K$—Specifies the start of service at customer i on day k .

This study approaches the problem as a multi-objective task: minimizing total travel time to enhance efficiency and minimizing the variance of daily travel time to ensure balanced workloads. Although these objectives may conflict, as reducing one can increase the other, accounting for the variance in daily travel time helps comply with time window restrictions. It prevents situations where one day is overloaded due to routing in a client-dense area, while the next day has very few visits. A weighted approach is employed, giving higher priority to total travel time (-4) over variance (-1). These values were chosen because, after testing various combinations, they proved most effective in satisfying all constraints. However, they can be adjusted based on specific priorities or preferences. An alternative using standard deviation was tested for interpretability, as both objectives would have the same unit measure, but it proved less effective at penalizing extremes. As illustrated in Appendix A, Figure A1, minimizing the variance of daily working durations leads to earlier and more consistent end-of-day times when compared to minimizing the standard deviation, thus better supporting compliance with time window constraints. For this reason, the objective function is expressed as

$$TTV = \sum_{k\in K}\sum_{i\in C}\sum_{j\in C} t_{ij}\, x_{kij} + \lambda\, \mathrm{Var}(t_k) \;\longrightarrow\; \min\,, \tag{6}$$

where λ is a weighting parameter balancing travel time and variance, and $\mathrm{Var}(t_k)$ is the variance of the daily working durations:

$$\mathrm{Var}(t_k) = \frac{1}{|K|}\sum_{k\in K}(t_k - \bar{d})^2, \quad \text{with } \bar{d} = \frac{1}{|K|}\sum_{k\in K} t_k\,, \quad \text{and } t_k = \sum_{i\in C}\sum_{j\in C} t_{ij}\, x_{kij}\,.$$

The constraints of the model are the following:

$$\sum_{j\in C_c} \mathbf{x}_{k0j} = 1\,, \quad \forall k \in K \tag{7}$$

$$\sum_{i\in C} \mathbf{x}_{kih} - \sum_{j\in C} \mathbf{x}_{khj} = 0\,, \quad \forall h \in C \quad \forall k \in K \tag{8}$$

$$\sum_{i\in C_c} x_{ki(N_c+1)} = 1\,, \quad \forall k \in K \tag{9}$$

$$\mathbf{x}_{kii} = 0\,, \quad \forall k \in K \quad \forall i \in C \tag{10}$$

$$w_{0,k} \le H_{in}\,, \quad \forall k \in K \tag{11}$$

$$w_{ik} \ge w_{jk} + s_j + t_{ji} - M\,(1 - \mathbf{x}_{kji})\,, \quad \forall k \in K \quad \forall i \in C \quad \forall j \in C \tag{12}$$

$$a_i \le w_{ik} \le b_i\,, \quad \forall i \in C \quad \forall k \in K \tag{13}$$

$$\sum_{k\in K}\sum_{j\in C} \mathbf{x}_{kji} \ge 1\,, \quad \forall i \in C_c \quad \forall k \in K \tag{14}$$

$$t_k = w_{(N_c+1)k} - w_{0k}\,, \quad \forall k \in K \tag{15}$$

$$t_k \le H\,, \quad \forall k \in K \tag{16}$$

$$\sum_{i\in C} \mathbf{x}_{kij} - y_{kj} = 0\,, \quad \forall k \in K \quad \forall j \in C_c \tag{17}$$

$$\sum_{k\in K} y_{ki} = 6\,, \quad \forall i \in C_1 \tag{18}$$

$$\sum_{k\in K} y_{ki} = 2\,, \quad \forall i \in C_2 \tag{19}$$

$$\sum_{k\in K} y_{ki} = 1\,, \quad \forall i \in C_3 \tag{20}$$

Constraints (7) and (9) ensure that the route begins and ends at the depot. Constraint (8) is intended to make sure that whenever the salesperson visits a customer, he

leaves for another customer or the depot. Constraint (10) does not allow the salesperson to visit the same customer consecutively. Constraint (11) ensures that the salesperson leaves the depot before H_{in} every day. The restraint imposed by (12) makes sure that if the salesperson on day k travels directly from customer i to customer j, then the start of service at customer i (w_{ik}) has to be after the initial service time at customer j (w_{jk}), plus its duration of the service (s_j) and plus the journey time from j to i (t_{ji}). Condition (13) guarantees that the visit is within the customer's visiting hours. In (14), the salesperson is obliged to make visits every day. Conditions (15) and (16) ensure that the salesperson's working hours are not exceeded. The constraint in (17) aims to ensure that the variables $\mathbf{x}_{kij}$ and y_{kj} are correctly aligned. Finally, Constraints (18) to (20) ensure that each customer's visit frequency is respected.

4.3. Model Implementation for Short- and Mid-Distance Salespeople

This subsection outlines the implementation of the multi-objective optimization model used for scheduling visits by short- and mid-distance salespeople. It describes how solutions are represented, how the GA operates, and how constraints are managed to ensure feasibility.

4.3.1. Solution Representation

Each solution is represented as a Python list, where each sublist corresponds to a day and contains the ordered client visits for that day, as illustrated in the Figure 5. This structure provides a clear and flexible way to model multi-day routes and simplifies manipulation during optimization.

Clients in category 3 must be visited twice a year, once in-person and once remotely, so the model spans the full year and selects the most efficient semester for the in-person visit. The number of in-person workdays, and thus the number of sublists per solution, is determined using Python's datetime module.

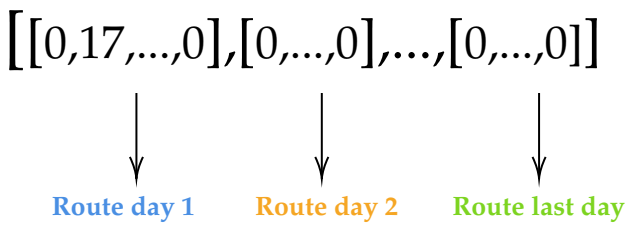


Figure 5. Representation of a route (individual/solution) in the GA model, showing the sequence of visits starting and ending at the salesperson's house (denoted by 0), with other numbers (e.g., 17) representing clients.

4.3.2. Genetic Algorithm Design

This part describes the structure of the genetic algorithm, including how the initial population is created and how selection, crossover, and mutation are applied. It also explains how constraints such as time windows and work hours are managed to ensure solutions are both optimized and feasible.

Initialization

The individuals are initialized by scheduling clients based on their visit frequencies for in-person visits per year, depending on their category. Clients are assigned to available days within their allowed time slots, prioritizing empty days to promote even distribution and reduce schedule imbalance. This strategy helps respect time windows and sets a strong foundation for feasible solutions. Once clients are assigned, each day's route is finalized by adding the depot (0) at the start and end. This process is repeated to generate the full population for the first generation.

Selection of Parents

Parent selection combines NSGA-II and tournament selection to ensure both quality and diversity. NSGA-II ranks individuals by Pareto dominance and crowding distance, prioritizing those that perform well across objectives and are more diverse. Then, tournament selection randomly compares two individuals, choosing the one with the better front or, if tied, the greater crowding distance. This process continues until the full parent pool is selected.

Crossover and Mutation

A two-point crossover is applied to exchange entire days (sublists) between two parent schedules, as represented in Figure 6, introducing variation while preserving overall structure. In this model, the mutation operators change the contents within each day. With a given probability, mutation occurs by selecting two distinct days within an individual and swapping one client from each, ensuring depots are not affected and that no duplicates are introduced in either day. This design allows the model to explore new valid configurations beyond simple day rearrangements.

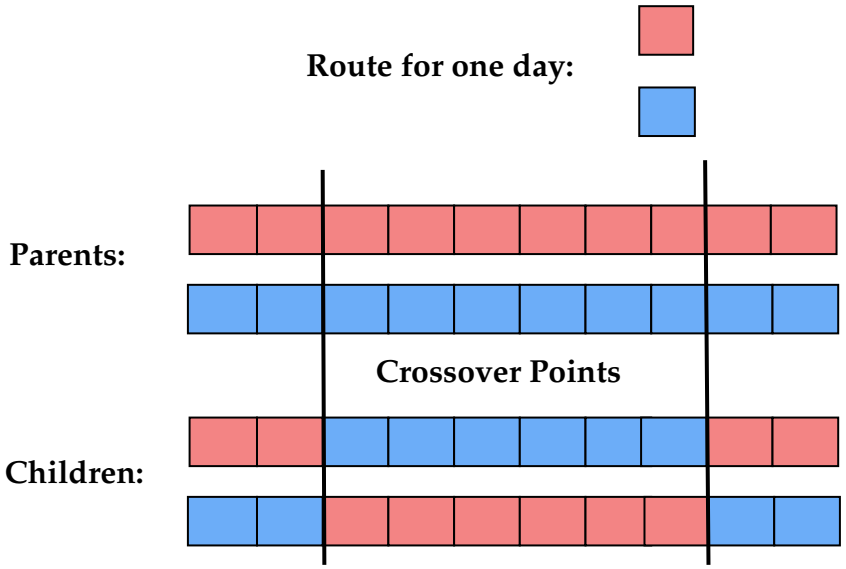


Figure 6. Representation of a two-point crossover.

Constraint Handling

Constraint violations are handled in two ways: through penalties in the evaluation function and a repair function after crossover and mutation. Penalties discourage infeasible solutions by adding weighted costs to the travel time, with higher penalties assigned to more critical constraints, such as visit frequency (500,000) and time windows (50,000). Lesser penalties are applied for exceeding daily work hours (1000) or not using all workdays (500).

Since crossover and mutation often generate infeasible solutions, a repair function is applied afterward to restore feasibility. If a client is scheduled for more visits than required, the algorithm removes visits from days with the longest working hours, prioritizing keeping visits on lighter days to maintain a balanced workload. If a client has too few visits, the algorithm adds visits on available days with the shortest working hours, ensuring constraints are met without overloading the schedule. Once client visit frequencies are corrected, a 2-opt heuristic is applied to each day's route to improve visit sequencing, which may have been degraded by the changes introduced during mutation, by swapping

pairs of visits and retaining changes that reduce total travel time, thereby refining overall route efficiency.

Island Model Implementation

To promote diversity and avoid premature convergence, the model uses an island model approach, where four independent populations evolve in parallel and periodically exchange solutions. This setup allows each island to explore different regions of the solution space while sharing strong individuals, improving overall solution quality. Migration is cyclic and occurs every 10 generations, helping to maintain diversity without causing excessive synchronization. Four islands were chosen to balance exploration and computational efficiency, ensuring manageable resource use while still capturing a broad range of solutions from the Pareto front.

Selection of the Best Solution (Individual)

After the model runs, each island holds 50 individuals. To select the best solution, the model identifies the individual on the first Pareto front of the four islands closest to the utopian point $(0, 0)$, which represents the ideal (though unattainable) minimum for both objectives. This is carried out by calculating the Euclidean distance of each Pareto-optimal solution to $(0, 0)$, and selecting the one with the smallest distance as the best individual. In Figure 3, a visual representation of this process is shown.

Handling Model Reruns

Throughout the year, changes such as client reclassification may alter visitation requirements, necessitating re-optimization of the routing plan. To address this, the model includes a rerun feature that considers all previous visits and focuses only on optimizing the remaining days. During reruns, individuals are generated with sublists corresponding to the missing days, and the algorithm proceeds with the same steps as previously explained. The resulting solutions are then merged with the original plan to complete the schedule up to the current date.

4.4. Model Implementation for Long-Distance Salespeople

Due to their extensive travel demands, the long-distance salesperson follows a distinct planning approach. Since this type of role involves significantly more time spent traveling, clients in groups 1 and 2 must now receive equal treatment, each visited four times per year, with two in-person and two remote visits. Unlike other salespeople, this individual typically travels for several consecutive days before returning home, rather than completing daily round-trips. Additionally, because the clients of these salespeople are more dispersed, some are located in very remote areas, where harsh winter weather can prevent travel for weeks at a time.

4.4.1. Geographical Clustering of Clients

Since some customers are located very far apart in the long-distance salespeople condition, which can make route optimization more complex and time-consuming, applying clustering to group nearby clients together provides the GA model with a useful starting point. This helps the algorithm better understand the spatial structure of the problem, guiding it towards more efficient solutions faster.

Due to hazardous winter conditions that might make it impossible to travel, only 40 weeks of the year were considered for optimization, avoiding the most travel-disrupted periods.

Clients were grouped into 10 geographic clusters using fuzzy C-means, ensuring spatial coherence while maintaining an even distribution of visits. This soft clustering

approach improves scheduling flexibility by allowing overlapping membership. The choice of 10 clusters using fuzzy C-means was made to balance regional grouping and scheduling flexibility across the 40-week planning period. Selecting too few clusters results in inefficient, dispersed routes, while too many creates small, inflexible groups. A total of 10 clusters achieved a practical compromise between these extremes and were supported by preliminary analysis. Varying the number of clusters can significantly affect route optimization outcomes. Preliminary tests using NSGA-II without any clustering or MDS preprocessing resulted in slower convergence and frequent violations of weekly time constraints due to incoherent client allocations across highly dispersed regions. The fuzzy clustering with MDS allowed the initial population to start from spatially meaningful groupings, which not only improved feasibility from the outset but also reduced the number of repair operations needed and accelerated the convergence process.

Clustering is based on clients' geographic coordinates. Since fuzzy C-means assumes Euclidean distances, the Haversine formula was used to compute real-world distances between points, followed by MDS to project them into Euclidean space for more accurate clustering. Each cluster is cyclically assigned to four fixed weeks; for example, cluster 1 corresponds to weeks 1, 11, 21, and 31, while cluster 10 corresponds to weeks 10, 20, 30, and 40.

4.4.2. Solution (or Individual) Representation

Each solution is a Python list of weekly sublists, where each week starts and ends at the depot, allowing overnight stays near clients. A 40-h weekly work limit ensures realistic, efficient long-distance planning within labor constraints.

4.4.3. Genetic Algorithm Design

Only the components of the genetic algorithm that differ from the previous model are described here; specifically, the initialization of individuals, the mutation process, and the repair function used for constraint handling. All other elements, such as crossover, selection, fitness evaluation, the island model, and the rerun logic, remain the same as before.

Initialization

Initialization in this model considers both visit frequency and spatial distribution using fuzzy C-means clustering. Each individual instance is a list of 40 weekly routes, starting and ending at the depot. Clients are assigned to weeks based on their required number of visits and their strongest cluster membership, e.g., once-visited clients are placed in a random week from their main cluster's four designated weeks, while twice-visited clients are assigned two distinct weeks. This ensures spatial coherence and avoids duplicate scheduling.

Mutation

In the current model, the mutation step has been adapted to reflect the fact that clients have already been assigned to weeks based on their highest fuzzy cluster membership, ensuring spatial coherence. Because of this, mutation focuses on reordering visits within each week rather than moving clients between weeks.

To optimize the order of visits within each route, simulated annealing is applied by reversing random segments of the route. The algorithm accepts changes based on improved travel time or, with decreasing probability, worse results, helping avoid local optima and refining the solution without disrupting the initial clustering.

Constraint Handling–Repair Function

Since the only constraint not directly handled during the initialization or mutation steps is the weekly maximum of 40 working hours, a dedicated repair function was implemented to enforce this limit.

The function begins by calculating the total travel time for each weekly route and identifies those that exceed the 40 h threshold. For each overloaded week, clients are ranked based on how weakly they are associated with that week's cluster; those with the lowest membership values are prioritized for reassignment. The function then relocates these clients to alternative weeks, where their cluster membership is stronger, and the resulting route still remains within the time constraint. This targeted reassignment helps reduce weekly overloads while respecting the fuzzy clustering structure that guides client–week associations.

4.5. Hyperparameter Tuning

To optimize the genetic algorithm's performance, key hyperparameters, including population size, crossover and mutation rates, and the number of generations, were tuned to balance solution quality and runtime. Contrary to typical GA practices, always applying both crossover and mutation yielded better results, especially given the individual's structure (lists of sublists representing routes). Crossover enabled effective exploration by exchanging entire routes, while mutation, enhanced with a 2-opt local or simulated annealing optimizer, improved refinement and diversity, which is especially valuable in small populations. It is important to note that, in our algorithm, the mutation operator is not purely random; instead, it includes a local optimization procedure: 2-opt for short- and mid-distance cases, or simulated annealing for long-distance instances. This hybrid mutation operator serves both to enhance population diversity and provide solution refinement. This approach differs from conventional genetic algorithms, which typically use lower, purely random mutation rates. In our setting, especially with small populations, this "intelligent mutation" did not hinder convergence; on the contrary, it consistently led to better solution quality by promoting both exploration and effective local improvement. The empirical results showed that our combined mutation/local search strategy outperformed standard random mutation in preliminary experiments.

A population size of 50 individuals per island was selected based on performance and efficiency trade-offs. Mutation applied at 100% is particularly beneficial in early generations (< 100) to avoid local optima, though this can be disruptive in longer runs. For the short-distance case, convergence was reached by generation 50; for mid- and long-distance cases, 100 generations were optimal, as increasing to 150 brought no substantial improvements.

The final run times were 16.56 min (short distance), 17.31 min (mid distance), and 59.29 min (long distance), the latter being slower due to the use of simulated annealing during route mutation. To improve clarity and support reproducibility, all key model components and algorithm parameters are summarized in Table 1.

Table 1. Summary of model notation and genetic algorithm parameters.

Symbol	Description
$C_c = \{1, 2, \ldots, N_c\}$	Set of customers
$C_d = \{0, N_c + 1\}$	Set of depots (start and end)
$C = C_c \cup C_d$	Set of all nodes (customers and depots)
$C_1 \subset C_c$	Customers with monthly visits (six in person/year)
$C_2 \subset C_c$	Customers with 4 yearly visits (two in person)
$C_3 \subset C_c$	Customers with 2 yearly visits (one in person)
$K = \{1, \ldots, N_k\}$	Set of planning days
t_{ij}	Travel time between nodes i and j
$[a_i, b_i]$ = [09:00, 16:00] h	Time window, during which service at node i is allowed
$H = 11$ h	Daily working hours of the salesperson
$H_{\text{in}} = 9$ h	Earliest allowable start time
$s(i) = 30$ min	Service duration at customer i
$M = 10^7$	Large constant used for constraint linearization
x_{kij}	Binary: A value of 1 if, on day k, j is visited after i; otherwise, the value was 0
bluey_{ki}	Binary: A value of 1 if customer i is visited on day k; otherwise, the value was 0
w_{ik}	Start time of service at customer i on day k
Genetic Algorithm Parameters	
Population size	50 individuals per island
Crossover rate	100% (always applied)
Mutation rate	100% (especially useful in early generations)
Local refinement	2-opt (short/mid-distance) or simulated annealing (long-distance) used in mutation
Number of generations	50 (short-distance), 100 (mid/long-distance)
Convergence behavior	No significant gain beyond 100 generations

5. Results

5.1. Analyses of the Time Matrices

In this work, it is considered that the short-distance salesperson serves a total of 588 customers, the mid-distance salesperson manages 342, and the long-distance salesperson is responsible for 313. For the short- and long-distance salespeople, most clients belong to category 2, whereas for the mid-distance salesperson, category 3 is the most common. However, in the three cases, the number of clients in categories 2 and 3 is fairly similar, while the number of clients in category 1 is very low, as shown in Figure 7.

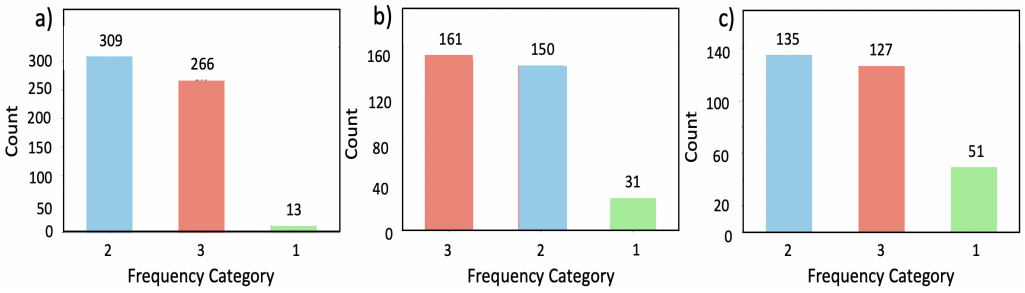


Figure 7. Panels (**a**), (**b**) and (**c**) show the distribution of client frequency categories for the short-, mid-, and long-distance salespeople, respectively. Clients in group 1 are shown in green, group 2 in blue, and group 3 in coral. While the number of clients in categories 2 and 3 remains relatively consistent, there is a notable decrease in category 1 clients across all three salespeople.

The time matrices used as input for the models for each of the three types of salespeople are based on proprietary data provided by Nors, Group, S.A., and cannot be shared publicly

due to confidentiality restrictions. As a result, illustrative examples of these matrices are not included as supplementary material. However, their structure and distribution in the modeling process are described in detail within this work. Figure 8 shows the distribution of each time matrix. The short-distance salesperson has a relatively compact, slightly right-skewed distribution, with most travel times between 10 and 40 min (mean: 27.04 min; median: 26.00 min; variance: 146.4 min²). The mid-distance salesperson exhibits a broader and multimodal distribution for travel times, extending beyond 200 min. This pattern is characterized by a higher average travel time of 83.45 min and a substantially larger variance of 2450.3 min squared, indicating that the clients served by this salesperson are geographically more widely distributed. The distribution of long-distance salespeople is highly influenced on the right, with most travel times between 100 and 500 min, a mean of 330.41, and extreme outliers beyond 1000 min, indicating a large service area.

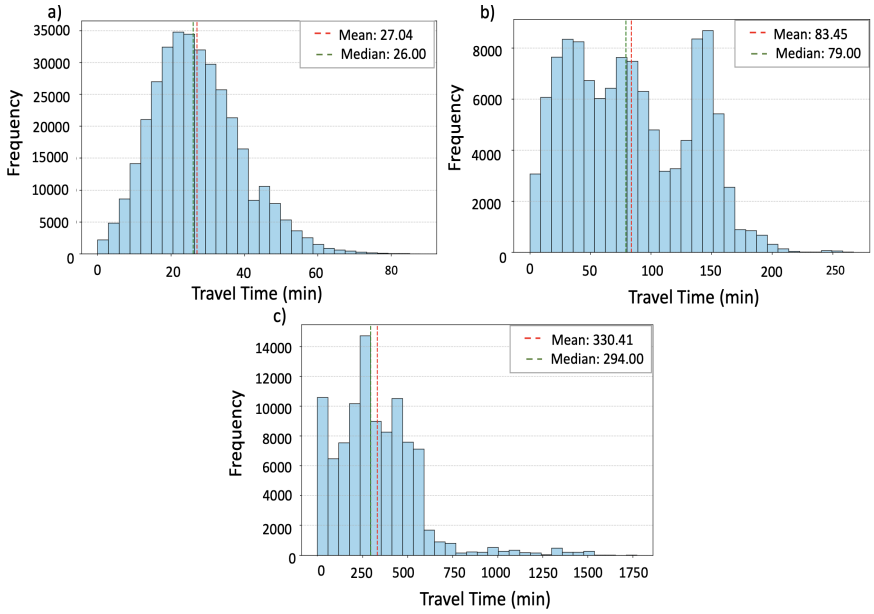


Figure 8. Distribution of travel times for (**a**) the short-distance, (**b**) mid-distance, and (**c**) long-distance salespeople.

Figure 9 further illustrates these differences through boxplots and violin plots. The short-distance salesperson shows a tight interquartile range (15–35 min) with a clear unimodal peak around 25 min and some longer outliers. The mid-distance has a wider interquartile range (50–130 min) and a multimodal distribution, suggesting substantial variability that complicates route optimization. For the long-distance salesperson, the interquartile range spans 150–400 min, with a long upper tail and multiple peaks, highlighting the need for careful planning due to significant variation in travel times.

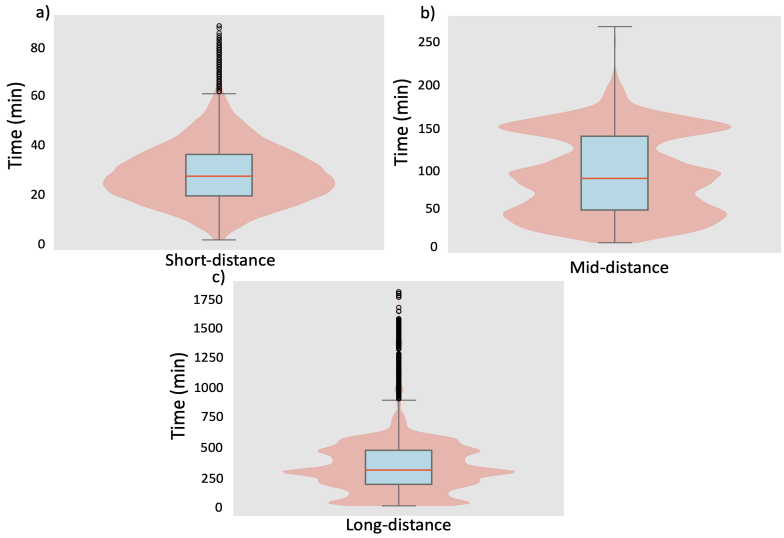


Figure 9. Distribution of pairwise travel times between locations for (**a**) short-distance, (**b**) mid-distance, and (**c**) long-distance salespersons. Each subplot combines a violin plot (showing the kernel density estimate of the time distribution) and a boxplot (showing the median, interquartile range, and outliers). The x-axis indicates the salesperson profile; the y-axis measures travel time (minutes).

5.2. Results for the Short-Distance Salesperson

For the short-distance salesperson, the algorithm generated balanced schedules that effectively grouped nearby clients, minimizing travel time (Figure 10a). The model ran in 16.56 min, producing a best solution with a total yearly travel time of 15,286 min (21.2 hpm → travelling hours per month) and a low daily work duration variance of 183.8 min^2, indicating consistent workload distribution. Daily client visits ranged from three to seven clients (average: 5.2), and all constraints were satisfied without penalties. These travel times were calculated by analyzing each day's route and extracting inter-location travel durations from the time matrix shown in Figure 8.

As shown in Figure 10b, daily routes ended between 12:51 and 14:40 (mean: 13:57), assuming a 9:00 start. This leaves ample time each day for non-travel tasks, supporting both efficiency and flexibility.

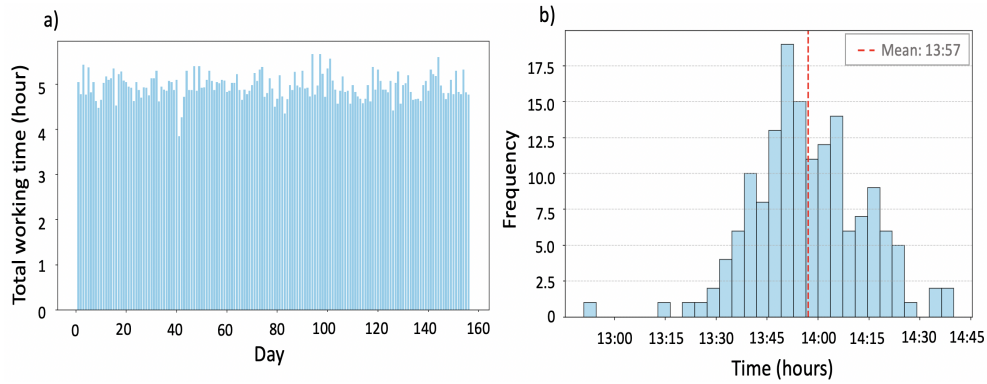


Figure 10. Workload and visit timing distribution for the short-distance salesperson. (**a**) Daily working hours across in-person visit days, accounting for travel time to and from each client, as well as 30 min visit durations. (**b**) Distribution of the ending times of the last client visit, assuming a departure from home at 9:00 a.m.

Figure 11 shows six solutions from the Pareto front, illustrating the trade-offs between the objectives. Observe that the Pareto front for the short-distance case exhibits a well-distributed set of non-dominated solutions, which is quantitatively supported by the low spread (Δ) [33] value of 0.0703, indicating good diversity among the solutions. Moreover,

this low value suggests that the population has settled and is not producing new, widely separated solutions in the final generations, a typical signal that the search process has stabilized. This visual evidence aligns with the convergence analysis in Section 4.5, where no significant improvements were observed beyond 50 generations, confirming that the front had stabilized. Still, the plot shows a clear spread of solutions, ranging from schedules with lower travel time but higher variance, to those with more balanced workloads but longer total travel times. The best solution, highlighted with a distinct marker, corresponds to the one with the smallest Euclidean distance to the origin (0,0), representing the most balanced trade-off.

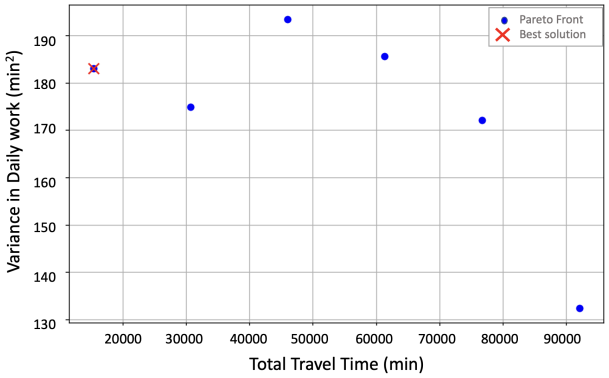


Figure 11. Pareto front plot showing six selected solutions for the short-distance case. The blue points represent local Pareto-optimal solutions obtained independently from different islands in the multi-island genetic algorithm. Although some solutions may appear dominated from a global perspective, they are included to illustrate the diversity of trade-offs identified across islands.

5.3. Results for the Mid-Distance Salesperson

For the mid-distance salesperson, the model ran in 17.31 min. As shown in Figure 12a), the resulting schedules were less balanced than for the short-distance case due to greater variability in travel times (Figures 8 and 9). Nonetheless, all constraints were satisfied. The total travel time was 35,637 min (49.5 hpm), with a high daily work duration variance of 1504.10 min^2.

As seen in Figure 12b), visit end times ranged from 9:39 to 16:16 (mean: 14:14), assuming a 9:00 departure. Many first clients are over an hour away, so earlier departures (e.g., 8:00) could shift schedules earlier while still respecting time windows, highlighting the importance of considering travel dynamics in interpreting these results.

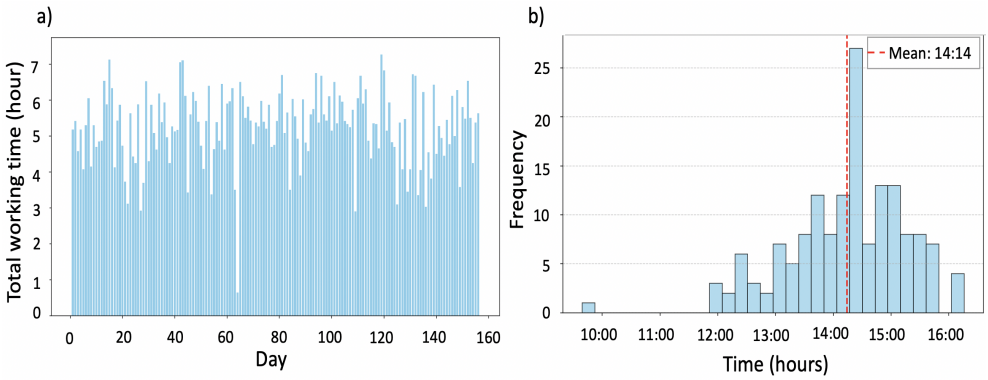


Figure 12. Workload and visit timing distribution for the mid-distance salesperson. (**a**) Daily working hours across in-person visit days, accounting for travel time to and from each client, as well as 30 min visit durations. (**b**) Distribution of the ending times of the last client visit, assuming a departure from home at 9:00 a.m.

Figure 13 illustrates the trade-off between total travel time and the variance in daily work duration for six selected solutions in the Pareto front. The chosen solution, marked with a red X and with the lowest Euclidean distance to the point (0,0), lies near the lower-left corner of the Pareto front, indicating a good balance between objectives. This solution achieves a relatively low total travel time while keeping daily work variance at a reasonable level, showing that the model successfully identifies efficient and balanced routing schedules. The overall set of non-dominated solutions displays a satisfactory distribution, as supported by the calculated spread (Δ) value of 0.28.

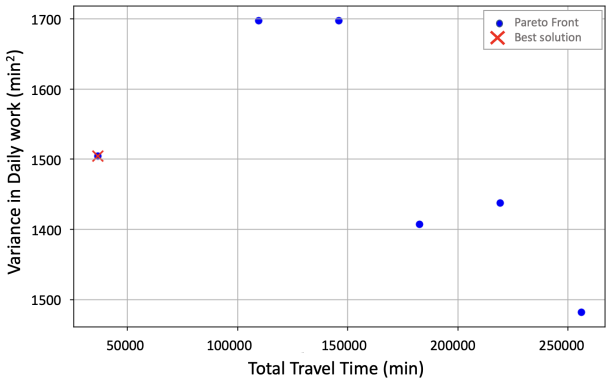


Figure 13. Pareto front plot showing six selected solutions for the mid-distance case. As in Figure 11, the blue points represent local Pareto-optimal solutions from different islands in the multi-island genetic algorithm, included to show the diversity of trade-offs.

5.4. Results for the Long-Distance Salesperson

Figure 14 shows the total working time per week for the long-distance salesperson. The running time of the model was 59.29 min. The best trade-off solution resulted in a total travel time of 29,204 min (40.6 hpm), with a variance in weekly work duration of 179,980.43 min^2. The results indicate some variability in weekly workloads, with most days falling between 10 and 26 h, though a few weeks reach even higher, peaking at over 34 h. These fluctuations may reflect the uneven geographic distribution of clients in some clusters, which exists. Some clients in a cluster are very far away from the others. Despite the variability, the results show that the 40 h working weekly limit is respected. The solution also satisfies all the other requirements.

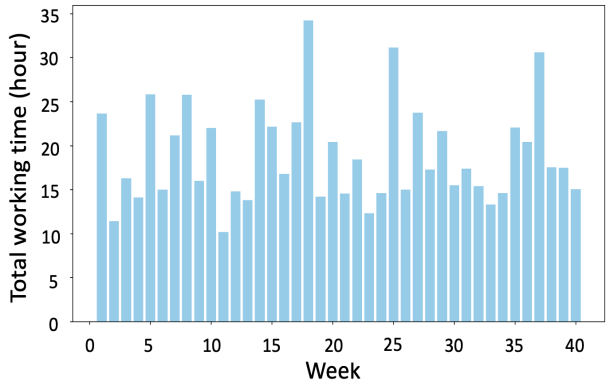


Figure 14. Weekly working hours across in-person visits, accounting for travel time to and from each client, as well as 30 min visit durations.

The Pareto front shown in Figure 15 illustrates the trade-off between total travel time and the variance in weekly working time for the long-distance salesperson. The best solution, marked in red, reflects a compromise between the two conflicting objectives, achieving relatively low total travel time while maintaining a moderate level of variance

since it is the solution with the smallest Euclidean distance to (0,0). This balance suggests that the model successfully identified a practical route plan that avoids excessive fluctuation in weekly workload without incurring prohibitively high travel costs, aligning well with real-world considerations for long-distance planning. The calculated spread (Δ) for the long-distance Pareto front is approximately 0.0655, indicating a well-distributed set of solutions along the front.

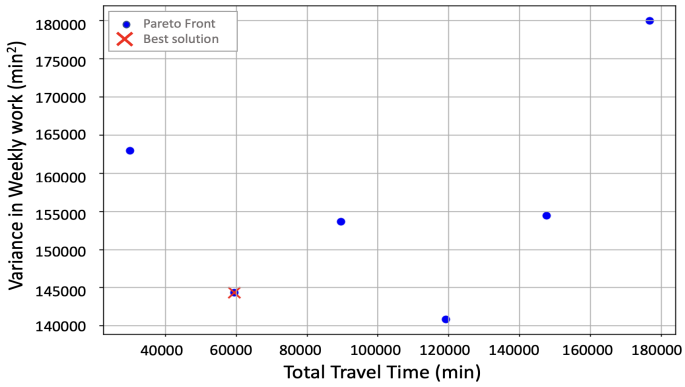


Figure 15. Pareto front plot showing six selected solutions for the long-distance case. As in Figure 11, the blue points show local Pareto-optimal solutions from different islands, highlighting the diversity of trade-offs.

6. Discussion

The results obtained from the optimization model were critically examined. The aim is to interpret how well the model performed in generating feasible and efficient schedules. This discussion also highlights the advantages of the model, identifies its limitations, and considers its practical implications for route planning.

6.1. Solution Evaluation and Performance

Since there are no available datasets that closely match the specific conditions addressed by this model that could serve as a benchmark test, it is not possible to precisely quantify how much the proposed solution improved client reach or reduced travel time relative to existing benchmarks. In the absence of appropriate benchmark datasets and the ability to directly calculate the proportional decrease in travel time relative to this expanded reach, alternative methods were used to evaluate the model's performance.

One way to evaluate the model's performance is by comparing the average home-to-client distance with the actual total travel time per client after optimization. For the short-distance salesperson, the average travel time from home to a client is 50.68 min, yet the total travel time per client across all routes is only 26.00 min. This indicates that the model effectively clusters clients and builds efficient multi-client routes, significantly reducing redundant travel. Similarly, for the mid-distance salesperson, while the average home-to-client distance is 123.45 min, the total travel time per client is reduced to 104.20 min. Although the reduction is less dramatic due to longer base distances, it still reflects meaningful route optimization and consolidation. These differences suggest that the model leverages spatial efficiencies by grouping geographically close clients and planning visit sequences that minimize back-and-forth travel. The same reasoning applies to the long-distance salesperson. In this case, the average home-to-client distance is 432.24 min, while the total travel time per client after optimization drops significantly to 92.89 min. This drastic reduction highlights the ability of the model to form highly efficient weekly routes by grouping clients in distant regions in a way that substantially minimizes total travel, even when operating on a broader geographic scale.

Since the model successfully met all predefined constraints, it was important to evaluate the quality of route ordering within each day. To do this, the Nearest-Neighbor (NN) algorithm was used as a benchmark. This algorithm starts at the depot (the salesperson's home) and iteratively visits the closest unvisited client, mimicking the intuitive decision-making process a salesperson could follow in the field. For each daily route generated by the GA model, the same set of clients was reordered using the NN algorithm, and the corresponding total travel time was computed. These values were then compared with the routes generated by the original GA model to assess whether the model was producing equally or more efficient route sequences.

We summarize the total travel time results obtained from the NN and GA approaches for each type of salesperson in Table 2. As shown, the GA model consistently outperforms the NN approach in all scenarios. The corresponding percentage reductions further illustrate the efficiency gains of the GA method. The Figure 16 illustrates the daily and weekly travel time differences between the NN and GA models for the three salespeople, highlighting the efficiency improvements with the GA model.

Table 2. Comparison of NN and GA travel time results by salesperson type.

Salesperson Type	Method	Total Travel Time (min)	Reduction (%)
Short-Distance	NN GA	49,977 15,286	69%
Mid-Distance	NN GA	64,270 35,637	45%
Long-Distance	NN GA	65,369 29,105	55%

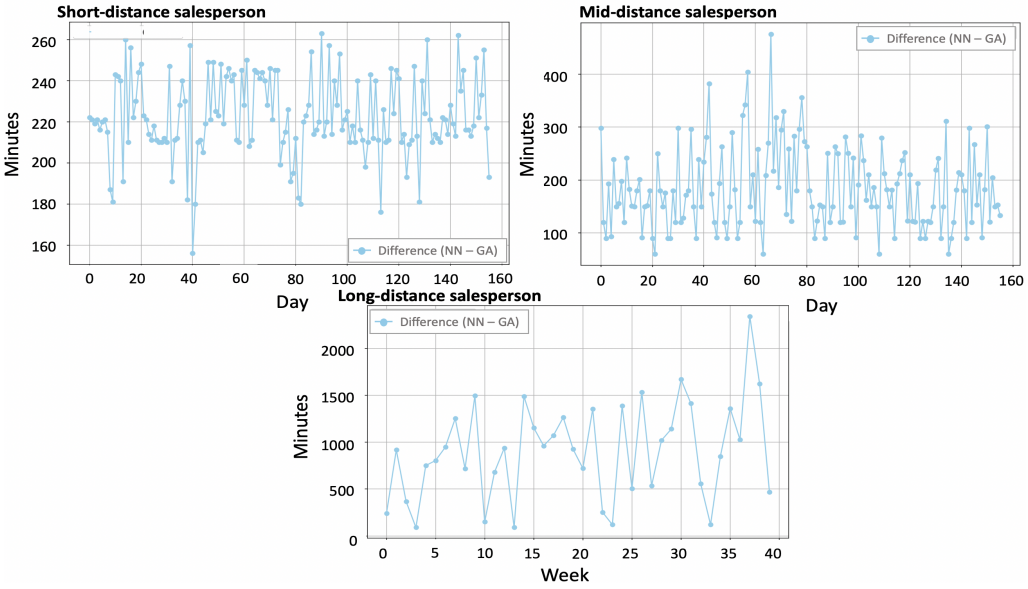


Figure 16. Differences in daily travel time between the NN and GA models for each in-person day, highlighting how the GA consistently outperforms the NN algorithm, which mimics salesperson intuition, by producing better-ordered routes.

The comparison with the NN algorithm reinforces the effectiveness of the GA model in optimizing route sequencing. NN reflects a common and intuitive approach that a salesperson might take, prioritizing proximity at each step. These findings highlight how efficient the GA model is in ordering client visits within each route.

6.2. Model Advantages and Limitations

The proposed model offers several advantages in addressing this adaptation of the PVRPTWs. Its key strength lies in its flexibility to handle diverse constraints such as client visit frequencies, time windows, and daily work hour limits. By using a multi-objective genetic algorithm that minimizes both travel time and workload variance, it generates balanced schedules and allows decision-makers to prioritize based on operational needs.

The integration of a 2-opt repair function further improves route quality by ensuring geographically logical sequencing. The model performs well across various geographic contexts, including long-distance routes, and can be easily updated when client data or constraints change, demonstrating strong adaptability and scalability.

However, the model has limitations. The lack of benchmark data prevents the full quantification of efficiency gains and does not yet account for real-world uncertainties, such as traffic or delays. Its reliance on accurate travel-time matrices also means that input errors could affect output quality. Since we are only accounting for the total travel time by car and not the costs associated with it, a major limitation of our approach is the exclusion of actual travel expenses and alternative modes of transportation, which, in some cases, could reduce both cost and time.

Another limitation is the absence of a direct performance comparison with other metaheuristic approaches, such as Tabu Search [24], Simulated Annealing [25], or Ant Colony Optimization [35], which are commonly applied to VRPTW variants. Although the Nearest-Neighbor heuristic was used as a baseline to evaluate route-ordering efficiency, future work should consider benchmarking the proposed model against these established methods on synthetic instances of a similar size to enable a more systematic evaluation of solution quality and runtime. Nonetheless, the choice of a multi-objective genetic algorithm in this study is supported by prior work showing the strong performance of GA-based methods in high-dimensional VRP contexts [36,37]. Our approach builds on these foundations by incorporating local search mechanisms, specifically 2-opt and simulated annealing, within the mutation step, enhancing both solution refinement and route sequencing quality.

Another factor to consider is that the current model does not explicitly account for uncertainties such as travel time variability or unexpected changes in client availability. Incorporating robustness assessment, as suggested in the recent literature [38] on multi-objective optimization under uncertainty, would improve the reliability of the proposed approach and represent a promising direction for future work.

7. Conclusions

We present a scheduling solution for large sales teams using a customized PVRPTWs model that incorporates critical constraints such as client visit frequencies, time windows, and daily work limits. In summary, the key findings of this study are the following:

- A MOGA was developed to balance total travel time with weekly workload distribution, offering flexibility for various business priorities;
- Scalability was demonstrated by applying the model to three representative salesperson profiles: short-, medium-, and long-distance;
- For long-distance scenarios, the combination of MDS clustering and fuzzy logic effectively grouped clients, leading to improved route quality;
- The model consistently produced geographically efficient routes and outperformed the NN heuristic baseline, reducing total travel time by up to 69%, by globally optimizing visit sequences rather than relying on step-by-step proximity, which often leads to suboptimal detours;

- The approach is practical for long-term use and easily adapts to updated client data or operational constraints;
- Although the model assumes static conditions and lacks benchmarking against other metaheuristics, performance proxies validate its effectiveness.

Beyond sales planning, the proposed methodology can also be adapted to other real-world scenarios involving geographically dispersed operations, such as maintenance scheduling, inspection tours, pharmaceutical logistics, or agricultural monitoring. Future developments will focus on extending the model to multi-agent coordination, incorporating uncertainty in travel and service times, and benchmarking its performance against alternative multi-objective optimization techniques.

Author Contributions: F.C.: conceptualization, methodology, software, validation, formal model, and writing—original draft; M.B.: resources, supervision, formal model, and writing—review and editing; P.L.: conceptualization, resources, and supervision; S.G.: resources, supervision, formal model, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: S.G. and M.B. were supported by CMUP, a member of LASI, which is financed by national funds through FCT–Fundação para a Ciência e a Tecnologia, I.P., under the project with reference UIDB/00144/2020. F.C. and P.L. were partially supported by Nors Group, S.A.

Data Availability Statement: The origin–destination time matrices used in this study were computed from proprietary operational records provided by Nors, Group, S.A. These data form part of Nors' internal logistics information system and are subject to strict confidentiality and commercial-sensitivity agreements. Consequently, the raw matrices and underlying records cannot be made publicly available.

Acknowledgments: The authors are thankful to FCUP, CMUP, and NORS for their indispensable support. We also sincerely thank the anonymous reviewers for their valuable suggestions, which helped improve the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GA	Genetic Algorithm
MDS	Multidimensional Scaling
MOGA	Multi-Objective Genetic Algorithm
NN	Nearest Neighbor
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PVRPTWs	Periodic Vehicle Routing Problem with Time Windows
VRP	Vehicle Routing Problem
SCAMOF	Scaling by MAjorizing a COmplicated Function

Appendix A

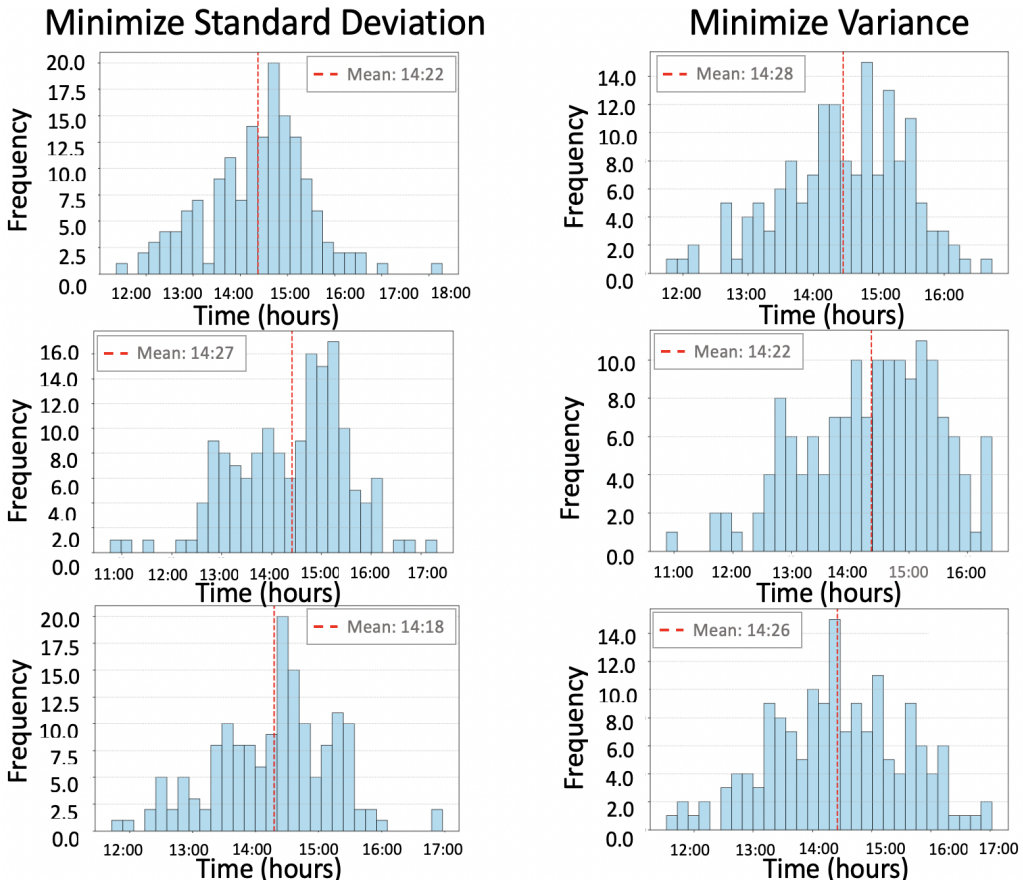


Figure A1. Distribution of the times of the latest stops every day for the mid-distance salesperson by minimizing the standard deviation and the variance of the daily travel times.

References

1. Santos, M.J.; Jorge, D.; Bonomi, V.; Ramos, T.; Póvoa, A. Enhancing logistics through a vehicle routing problem with deliveries, pickups, and backhauls. *Int. Trans. Oper. Res.* **2025**, *33*, 13577. [CrossRef]
2. Konstantakopoulos, G.D.; Gayialis, S.P.; Kechagias, E.P. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Oper. Res.* **2022**, *22*, 2033–2062. [CrossRef]
3. Yu, B.; Yang, Z.Z. An ant colony optimization model: The period vehicle routing problem with time windows. *Transp. Res. Part E Logist. Transp. Rev.* **2011**, *47*, 166–181. [CrossRef]
4. McKinney, W. Data structures for statistical computing in Python. *SciPy* **2010**, *445*, 51–56.
5. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
6. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
7. Fortin, F.A.; De Rainville, F.M.; Gardner, M.A.G.; Parizeau, M.; Gagné, C. DEAP: Evolutionary algorithms made easy. *J. Mach. Learn. Res.* **2012**, *13*, 2171–2175.
8. Mirzaei, K.; Arashpour, M.; Asadi, E.; Masoumi, H.; Bai, Y.; Behnood, A. 3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. *Adv. Eng. Inform.* **2022**, *51*, 101501. [CrossRef]
9. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
10. Bora, D.J.; Gupta, D.A.K. A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *arXiv* **2014**, arXiv:1404.6059. [CrossRef]
11. Deng, C.; Gao, J.; Lu, K.; Luo, F.; Sun, H.; Xin, C. Neuc-MDS: Non-Euclidean multidimensional scaling through bilinear forms. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 121539.
12. Marín Díaz, G.; Gómez Medina, R.; Aijón Jiménez, J.A. Integrating Fuzzy C-Means Clustering and Explainable AI for Robust Galaxy Classification. *Mathematics* **2024**, *12*, 2797. [CrossRef]

13. Sharmila, S.; Sabarish, B.A. Analysis of distance measures in spatial trajectory data clustering. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1085*, 012021. [CrossRef]
14. Mead, A. Review of the development of multidimensional scaling methods. *J. R. Stat. Soc. Ser. D (Stat.)* **1992**, *41*, 27. [CrossRef]
15. Chopde, N.R.; Nichat, M. Landmark based shortest path detection by using A* and Haversine formula. *Int. J. Innov. Res. Comput. Commun. Eng.* **2013**, *1*, 298–302.
16. Kruskal, J.B. Nonmetric multidimensional scaling: A numerical method. *Psychometrika* **1964**, *29*, 115–129. [CrossRef]
17. Baldoquin, M.G.; Martinez, J.A.; Díaz-Ramírez, J. A unified model framework for the multi-attribute consistent periodic vehicle routing problem. *PLoS ONE* **2020**, *15*, e0237014. [CrossRef]
18. Pirkwieser, S.; Raidl, G.R. Multiple variable neighborhood search enriched with ILP techniques for the periodic vehicle routing problem with time windows. In *International Workshop on Hybrid Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2009.
19. Rothenbächer, A.K. Branch-and-price-and-cut for the periodic vehicle routing problem with flexible schedule structures. *Transp. Sci.* **2019**, *53*, 850–866. [CrossRef]
20. Gendreau, M.; Iori, M.; Laporte, G.; Martello, S. A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks: Int. J.* **2008**, *51*, 4–18. [CrossRef]
21. Cattani, S. Time Agitation Heuristic A New Constructive Heuristic for the VRPTW. Available online: https://bdta.abcd.usp.br/directbitstream/75d4c958-9fb0-4df8-926a-2998a867f0e4/SimoneCattani.pdf (accessed on 3 October 2024).
22. Ombuki, B.; Ross, B.J.; Hanshar, F. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Appl. Intell.* **2006**, *24*, 17–30. [CrossRef]
23. Luke, S. *Essentials of Metaheuristics*; Publisher Lulu: Research Triangle, NC, USA, 2013. Available online: http://cs.gmu.edu/~sean/book/metaheuristics/(accessed on 4 October 2024).
24. Cordeau, J.F.; Laporte, G.; Mercier, A. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* **2001**, *52*, 928–936. [CrossRef]
25. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
26. Aydemir, E.; Karagul, K. Solving a periodic capacitated vehicle routing problem using simulated annealing algorithm for a manufacturing company. *Braz. J. Oper. Prod. Manag.* **2020**, *17*, 1–13. [CrossRef]
27. Lin, S.W.; Ying, K.C.; Lee, Z.J.; Chen, H.S. Vehicle routing problems with time windows using simulated annealing. In Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, 8–11 October 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 1, pp. 645–650.
28. Berger, J.; Salois, M.; Begin, R. A hybrid genetic algorithm for the vehicle routing problem with time windows. In *Advances in Artificial Intelligence: 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI'98 Vancouver, BC, Canada, June 18–20, 1998*; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 1998; pp. 114–127.
29. Nguyen, P.K.; Crainic, T.G.; Toulouse, M. A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. *J. Heuristics* **2014**, *20*, 383–416. [CrossRef]
30. Back, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*; Oxford University Press: Oxford, UK, 1996.
31. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
32. Castro, C.F.; António, C.C.; Sousa, L.C. Multi-Objective Optimisation of Hot Forging Processes using a Genetic Algorithm. In Proceedings of the Tenth International Conference on Computational Structures Technology, Valencia, Spain, 14–17 September 2010.
33. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
34. Conforti, M.; Cornuéjols, G.; Zambelli, G. Integer programming models. In *Integer Programming*; Springer International Publishing: Cham, Switzerland, 2014; pp. 45–84.
35. Blum, C. Ant colony optimization: Introduction and recent trends. *Phys. Life Rev.* **2005**, *2*, 353–373. [CrossRef]
36. Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2004**, *31*, 1985–2002. [CrossRef]
37. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **2013**, *40*, 475–489. [CrossRef]
38. D'Agostino, D.; Minelli, F.; Minichiello, F. New genetic algorithm-based workflow for multi-objective optimization of Net Zero Energy Buildings integrating robustness assessment. *Energy Build.* **2023**, *284*, 112841. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.