Department of Management Science & Technology

MSc in Business Analytics

Investigating Optimization Techniques for the Consistent Vehicle Routing Problem with Heterogenous fleet

By:
Eleftherios Fthenos

Student ID Number: p2822227
Name of Supervisor: Emmanouil Zachariadis

December, 2024
Athens, Greece

Table of Contents

A. Introduction

A1. Vehicle Routing Problem Variants

Efficient logistics and transportation management are at the heart of numerous industries, playing a critical role in sectors ranging from retail distribution to healthcare services. One of the most fundamental and challenging problems in this context is the Vehicle Routing Problem (VRP), which focuses on determining optimal routes for a fleet of vehicles tasked with serving a set of customers. Since its inception by Dantzig and Ramser in 1959, the VRP has undergone extensive development, giving rise to various extensions that address the increasing complexity of real-world constraints, such as vehicle capacities, time windows, and customer-specific requirements.

Among the most practical and widely studied variants is the Consistent Vehicle Routing Problem (ConVRP), which adds a layer of complexity by incorporating service consistency requirements into the classic VRP framework. In today's business environment, many companies must not only aim for cost efficiency but also maintain a high level of service quality. Consistency in service delivery is one way to achieve this, where companies ensure that customers are visited by the same vehicle or driver and at roughly the same time during each service period. This is particularly crucial in sectors like healthcare, home delivery services, and grocery distribution, where reliable and predictable service can significantly enhance customer satisfaction and loyalty.

The challenge of maintaining consistency is compounded by the fact that, in real-world scenarios, the assumption of a homogeneous fleet — where all vehicles have the same capacities, costs, and operational characteristics — is rarely realistic. In practice, vehicle fleets are heterogeneous, consisting of vehicles with varying capacities, speeds, costs, and even environmental constraints. For instance, fleets may include electric vehicles, hybrids, and conventional vehicles, each with distinct operating costs and restrictions. Additionally, urban areas often impose regulatory limits, such as low-emission zones, which further constrain the types of vehicles that can be used for deliveries in specific locations .

The **Consistent Vehicle Routing Problem with Heterogeneous Fleet (HConVRP)**, introduced by Stavropoulou (2022), expands upon the traditional ConVRP by integrating the additional complexity of heterogeneous vehicle fleets . In this variant, the goal is not only to minimize the operational costs of servicing customers but also to ensure consistent service delivery. Consistency, in this context, can be defined along several dimensions, such as driver consistency (ensuring that the same driver visits a customer over multiple service periods) and time consistency (ensuring that customers are visited at approximately the same time across service periods). This is particularly relevant in industries where building a long-term relationship with customers is crucial, such as pharmaceutical deliveries, elderly care services, and home meal delivery.

A2. Problem Context

In modern logistics operations, a fleet of vehicles is often heterogeneous due to various operational, financial, and environmental factors. For example, fleet heterogeneity may arise because vehicles were procured at different times and therefore vary in terms of technological features and operational costs. Moreover, environmental regulations, especially in urban settings, can dictate the use of specific types of vehicles, such as electric or hybrid models, to meet emission standards. For instance, cities like London have introduced ultra-low emission zones (ULEZ) that restrict the use of high-emission vehicles within city centers, which forces logistics companies to diversify their fleet to comply with these regulations.

Fleet heterogeneity also includes operational differences, such as vehicle capacities, fuel efficiency, and speed. These factors can have a significant impact on the overall routing and scheduling of vehicles. In addition to these operational concerns, businesses must also account for customer-specific requirements, such as the need for refrigerated vehicles for perishable goods or vehicles with specific loading equipment for large or heavy items.

The inclusion of **service consistency** as a constraint introduces an additional layer of complexity to the already challenging task of vehicle routing. For industries where customer satisfaction and retention are paramount, consistent service is essential. This is particularly true in sectors like healthcare, where patients require regular, predictable visits from the same service provider, or in grocery delivery, where customers prefer deliveries to occur at the same time each day. As Groër et al. (2009) and other researchers have demonstrated, maintaining consistency improves operational efficiency by allowing service providers to become familiar with their customers' specific needs and preferences, thus reducing service times and enhancing overall satisfaction.

The **Heterogeneous Consistent Vehicle Routing Problem (HConVRP)** tackled in this thesis integrates these considerations by seeking to optimize the use of a heterogeneous fleet while simultaneously ensuring consistent service across multiple time periods. This requires the development of robust algorithms capable of handling both the diversity of the fleet and the consistency constraints. By addressing this gap in the current literature, this study builds on the foundational work of Stavropoulou (2022), who proposed a mathematical model and a hierarchical Tabu Search (HTS) framework for solving the HConVRP.

The key objectives of the HConVRP are twofold: first, to minimize the total transportation cost, which includes both the fixed and variable costs associated with operating a heterogeneous fleet, and second, to maintain consistent service in terms of both driver assignments and delivery times over a multi-period planning horizon. This dual objective is particularly challenging given the constraints imposed by fleet heterogeneity and customer demands for consistency .

In this thesis, a Variable Neighborhood Descent (VND) algorithm is proposed to solve the Consistent Vehicle Routing Problem with Heterogeneous Fleet (HConVRP). This algorithm explores various neighborhoods systematically, optimizing the solution by switching between different types of neighborhood structures, such as ChangeVehicleChain, SwapVehicle, ChangeVehicle, and 2-Opt operations. By applying this iterative improvement technique, the algorithm attempts to improve an initial solution generated through a constructive heuristic that takes into account the heterogeneity of the fleet and the consistency constraints imposed by the problem.

The primary focus of this work is to minimize operational costs while maintaining consistent service in terms of both vehicle assignments and customer delivery times over multiple periods. The VND algorithm provides a flexible and powerful framework to explore diverse solution spaces, ensuring that the algorithm can handle the problem's complexity, which arises from both the fleet's heterogeneity and the stringent consistency requirements. The detailed results of the optimization process and its efficacy will be discussed in the following sections, including a comparison of the initial and improved solutions, offering insights into the algorithm's performance in solving the HConVRP.

B. Literature and Related Works

The Vehicle Routing Problem (VRP) has been extensively studied in the field of operations research since its introduction by Dantzig and Ramser (1959). It focuses on determining optimal routes for a fleet of vehicles serving a set of customers. Variants of the VRP have emerged over time, addressing increasingly complex real-world constraints. Among these, the **Consistent Vehicle Routing Problem (ConVRP)** and its extensions are particularly relevant to this thesis. These problems incorporate service consistency requirements, which are essential in applications such as healthcare, home deliveries, and retail distribution, where customers expect reliable and predictable service.

B1. Consistent Vehicle Routing Problem (ConVRP)

The ConVRP was first introduced by Groër et al. (2009), who proposed consistency constraints that go beyond the typical capacity and distance limits in classical VRP formulations. The core concept of the ConVRP is that customers should be visited by the same vehicle or driver and at approximately the same time across multiple periods. This problem gained attention due to its practical importance in industries where consistent customer relationships and service quality are key for customer retention. Groër et al. (2009) developed a **Record-to-Record Travel (RRT)** heuristic to solve the ConVRP, employing templates to optimize routes that repeat across days while maintaining consistency. The method’s template-based approach has since become a common strategy in solving ConVRP problems. Later studies, such as those by Kovacs et al. (2014), enhanced this approach by developing advanced metaheuristics, including **Adaptive**

Large Neighborhood Search (ALNS) and **Cluster Column Generation (CCG)**. These methods provided better solutions for larger instances of the ConVRP, improving upon Groër's initial work .

B2. Extensions to the ConVRP

Several extensions of the ConVRP have been proposed to address specific real-world applications. For example, Feillet et al. (2014) focused on time consistency, while Braekers and Kovacs (2016) introduced person consistency to better model services like home healthcare, where the same caregiver must visit a patient at regular intervals. Other researchers such as Smilowitz et al. (2013) expanded the problem into multi-objective frameworks, balancing consistency with other operational goals like cost minimization. In recent years, a notable trend has been the combination of VRP consistency constraints with other features, such as time windows and profits. Stavropoulou et al. (2019) introduced the **Consistent Vehicle Routing Problem with Profits (ConVRP-P)**, which blends customer service consistency with profit maximization objectives. This variant considers both frequent customers, who require regular visits, and non-frequent customers, whose inclusion in the route depends on their potential profitability.

B3. Heterogeneous Fleet VRP (HVRP)

In most VRP studies, fleets are assumed to be homogeneous, meaning all vehicles are identical in terms of capacity, speed, and operating costs. However, this assumption is rarely realistic in practice. Vehicle fleets often consist of different types of vehicles, such as electric, hybrid, and conventional vehicles, each with varying capacities and operating costs. This led to the development of the **Heterogeneous Fleet VRP (HVRP)**, introduced by Golden et al. (1984) and extended by Taillard (1999), who focused on limited and fixed fleet sizes. Koç et al. (2016) provided a comprehensive classification of HVRP variants, categorizing them based on whether the objective function includes fixed and variable costs or just variable costs. HVRPs have since been applied to numerous real-world contexts, such as waste collection, home delivery, and retail distribution. Despite the significant progress in HVRP research, the literature still lacks comprehensive solutions for problems that combine heterogeneity with consistency constraints .

B4. The Heterogeneous Consistent Vehicle Routing Problem (HConVRP)

Stavropoulou (2022) addressed this gap by introducing the **Heterogeneous Consistent Vehicle Routing Problem (HConVRP)**, which incorporates both consistency and fleet heterogeneity into the vehicle routing framework. In this model, the goal is to optimize routes while ensuring consistency in terms of driver assignments and delivery times, and simultaneously accounting for the diverse operational characteristics of the fleet. Stavropoulou's work uses a **Hierarchical Tabu Search (HTS)** combined with **Variable Neighborhood Descent (VND)** to solve the problem, demonstrating its effectiveness on benchmark instances.

B5. Path Consistency in VRP

Another recent development in the VRP literature is the concept of **path consistency**, introduced by Yao et al. (2021). This variant of the ConVRP focuses on ensuring that vehicles follow consistent paths in a road network, considering both customer and road-network constraints. The **Branch-Price-and-Cut** algorithm developed for this problem shows that maintaining consistent paths can reduce travel costs, particularly when frequent roads or arterial segments are repeatedly traversed .

B6. Related Applications

The ConVRP and its extensions have been applied to a wide range of industries. For example, pharmaceutical distribution often requires consistent and reliable deliveries to ensure patient safety. Similarly, home meal delivery services benefit from consistency to improve customer satisfaction and loyalty. In urban logistics, consistency is also essential for addressing environmental regulations, such as those imposed by low-emission zones, where fleet heterogeneity plays a significant role in compliance .

C. Problem Definition

The Heterogeneous Consistent Vehicle Routing Problem (HConVRP) extends the classical vehicle routing problem by integrating two critical real-world aspects: fleet heterogeneity and service consistency. The problem is modeled on a complete undirected graph $G = (N, E)$, where $N = \{0,1,2,...,n\}$ represents a set of nodes and $E = \{(i, j) \mid i, j \in N, i \neq j\}$ is the set of edges connecting these nodes. The depot is located at node 0, while the remaining nodes $N_c = N \setminus \{0\}$ represent the customers to be served. Each edge $(i, j) \in E$ has an associated non-negative distance d_{ij} and the distance matrix $[d_{ij}]$ is symmetric, satisfying the triangle inequality.

C1. Fleet Characteristics

The fleet consists of multiple types of vehicles, categorized into H distinct types, denoted as $K = \{1,...,H\}$. Each vehicle type $k \in K$ has a fleet of h_k identical vehicles available. The heterogeneity of the fleet is reflected in the varying attributes of each vehicle type, including carrying capacity Q_k, maximum operational duration T, average speed B_k, fixed costs F_k, and variable costs V_k. Fixed costs encompass rental or amortization and insurance expenses, while variable costs cover operational aspects such as fuel, maintenance, and driver costs. Due to different average speeds, the travel time $t_{ij}^k = \dfrac{d_{ij}}{B_k}$ is specific to each vehicle type k.

Each vehicle operates on a depot-returning route, starting and ending at the depot, with only one route allowed per period $p \in P$, where $P$is the set of planning periods $P = \{1,...,g\}$. Vehicle compatibility may also be a constraint, such that certain vehicles may not be able to traverse specific routes due to regulatory or physical restrictions. This compatibility is denoted by u_{ij}^k, where $u_{ij}^k = 1$ indicates that vehicle type k can traverse edge (i, j) and $u_{ij}^k = 0$ otherwise.

C2. Customer and Service Requirements

The set of customers N_c can be further divided into two subsets: frequent customers N_f and non-frequent customers N_{nf}. Frequent customers require consistent service across multiple periods, whereas non-frequent customers may have less stringent service requirements.

Each customer $i \in N_c$ has specific service demands q_{ip} for each period $p \in P$ in which they require service. Additionally, each customer has a predetermined service time s_{ip}, which must be accounted for during route planning. The customer-specific service needs for each period are indicated by binary variables w_{ip}, where $w_{ip} = 1$ if customer i needs to be serviced in period p, and $w_{ip} = 0$ otherwise.

C3. Objective and Constraints

The primary objective of the HConVRP is to design a set of routes that minimizes the total transportation cost, encompassing both the fixed costs associated with vehicle deployment and the variable costs of traversing the routes. The problem is subject to a series of constraints, which ensure that operational and customer-specific requirements are met. These constraints are detailed as follows:

1. **Customer Service Constraints:** Each customer $i \in N_c$ must be visited exactly once in any period p where $w_{ip} = 1$. This ensures that no customer is left unattended during a required service period.
2. **Fleet Constraints:** For each vehicle type $k \in K$, no more than h_k vehicles can be dispatched from the depot in any given period $p \in P$. This reflects the limited availability of each type of vehicle within the fleet.
3. **Route Connectivity**: The routes must be continuous and start and end at the depot. The design of the route ensures that no vehicle violates the connectivity requirements, allowing for efficient traversal between the depot and customer locations.
4. **Driver Consistency**: One of the distinguishing features of the HConVRP addressed in this thesis is the requirement for driver consistency. Frequent customers $i \in N_f$ must be serviced by the same vehicle m across all periods $p \in P$ where they require service. This builds a sense of reliability and familiarity, particularly important for sectors that rely on long-term customer relationships.
5. **Vehicle Capacity:** Each vehicle type k has a maximum load capacity Q_k that cannot be exceeded on any route. The accumulated demand from the customers served by a vehicle on a single route must not surpass this limit.

6. **Route Duration:** The maximum duration T that any vehicle route can take is also constrained. This ensures that vehicles complete their routes within a feasible timeframe, accommodating both travel time and customer service time.
7. **Vehicle Compatibility:** Vehicles may face restrictions on the roads they can use due to physical constraints (e.g., low-emission zones) or legal regulations. Only compatible vehicles can be assigned to specific edges, as denoted by u_{ij}^k.
8. **Arrival Time and Synchronization**: While this thesis does not address time consistency constraints (ensuring arrival times fall within a specific window across periods), it ensures no vehicle idling occurs. Vehicles are scheduled to arrive, service, and depart without unnecessary delays, adhering to operational efficiency.

C4. Problem Complexity and Solution Approach

The HConVRP introduces significant complexity compared to traditional VRP models, primarily due to the dual challenge of handling fleet heterogeneity and enforcing driver consistency over multiple periods. The combination of these elements necessitates robust algorithmic solutions that can effectively navigate a complex solution space. Traditional VRP methods often fall short, necessitating more sophisticated approaches capable of handling multiple objective functions and constraints simultaneously.

The mathematical formulation used to model the HConVRP involves binary decision variables to indicate whether a vehicle traverses a specific edge and continuous variables to capture vehicle arrival times. Constraints are used to enforce vehicle capacities, service requirements, driver consistency, and route connectivity. The objective function aims to minimize both the fixed costs associated with deploying vehicles and the variable costs related to distance and operation, providing an optimal routing plan that satisfies all constraints.

The approach in this thesis builds upon existing formulations by simplifying and focusing specifically on the driver consistency constraint, enabling the development of a Variable Neighborhood Descent (VND) algorithm to address the problem effectively. This solution framework is detailed in the following section, where the heuristics and optimization strategies used to tackle the HConVRP are presented and analyzed. This problem definition sets the stage for exploring practical algorithmic solutions, considering real-world logistics scenarios where fleet heterogeneity and customer service consistency play crucial roles in achieving operational efficiency.

D. Solution Framework and Analysis

The solution framework developed in this thesis addresses the complexities of multi-period vehicle routing with a heterogeneous fleet, emphasizing consistency across multiple service periods. Unlike traditional approaches that can handle each period independently, the interdependencies created by service consistency requirements—especially for frequent customers—necessitate a more integrated solution method. These interdependencies make it essential to simultaneously consider customer assignments and route configurations over the entire planning horizon.

To manage these complexities, the framework employs a Variable Neighborhood Descent (VND) algorithm. This approach systematically explores different neighborhood structures, seeking to iteratively improve the initial solution. By alternating between various types of moves—ranging from complex vehicle reassignments to simpler route optimizations—the algorithm can adaptively navigate the problem's constraints. Moreover, the VND algorithm integrates a Tabu list for specific neighborhoods (Change Vehicle Chain, Swap, and Relocation) to prevent repetitive moves that could hinder progress. This mechanism helps maintain the diversity of the search process by avoiding cycles and encouraging exploration of new solutions. Overall, this iterative approach allows the framework to refine customer-to-vehicle assignments and optimize route sequences, effectively minimizing total transportation costs while upholding consistency requirements across the planning horizon.

D1. Initial Solution Construction

The development of an effective initial solution is a critical step in solving vehicle routing problems, particularly those with complex constraints like the HConVRP. A well-constructed starting point allows for more efficient exploration of the solution space during the subsequent optimization phases. The initial solution in this framework addresses the distinct requirements of frequent and non-frequent customers, taking into account both consistency demands and the diverse characteristics of the heterogeneous fleet. The algorithm follows a structured, multi-phase approach:

1. **Identification of Frequent Customers**: The first step involves identifying customers who require service in multiple periods within the planning horizon. These customers are classified as "frequent" because maintaining consistency in their service (e.g., ensuring they are visited by the same vehicle across all required periods) is crucial for operational efficiency and customer satisfaction. Identifying these customers early in the process allows the algorithm to prioritize their assignment, aligning with strategies used in previous studies on consistent VRPs, such as those by Groër et al. (2009) and Stavropoulou (2022).
2. **Assignment of Frequent Customers**: Once the frequent customers are obtained, the algorithm proceeds with assigning them to compatible vehicles. Each customer is considered for insertion into potential routes across all periods in which they have demand. The algorithm evaluates all feasible insertion positions, calculating the associated costs without immediately committing to a move. This allows the selection of the least-cost insertion across all relevant periods, ensuring that the solution remains feasible and cost-efficient while maintaining service consistency since the beginning. This approach is informed by principles seen in template-based routing heuristics, where recurring patterns are leveraged for better consistency management, although cost-efficiency is not a key aspect of template routes (Feillet et al., 2014; Groër et al., 2009).

   The decision-making process is guided by evaluating both fixed and variable costs, factoring in vehicle capacity, travel time, and operational characteristics (e.g., speed and fuel efficiency). By adopting a holistic view of the planning horizon, the algorithm ensures that the frequent customers receive consistent service, thus meeting one of the core constraints of the HConVRP.

3. **Randomized Assignment of Non-Frequent Customers**: After assigning frequent customers, the algorithm shifts focus to non-frequent customers who only need service in a single period across the planning horizon. To maintain flexibility and avoid deterministic outcomes, the algorithm employs a randomized insertion approach. It calculates the best insertion points with the lowest cost for each customer, for each vehicle during the period they require service. Then, for each each of customer, it selects a random vehicle to assign among the k best vehicles, based on a randomized cost ranking. This ensures that the initial solution can accommodate various customer demands without becoming rigid, thereby providing a better foundation for subsequent optimizations. The flexibility introduced at this stage mitigates the risk of local optima, a common issue in deterministic initial assignments.
4. **Solution Evaluation and Cost Calculation**: Following the assignment of both frequent and non-frequent customers, the algorithm evaluates the overall solution by calculating the total cost. The cost function integrates both fixed costs (e.g., vehicle deployment) and variable costs (e.g., distance traveled, fuel consumption). This comprehensive evaluation ensures that the initial solution not only meets operational constraints but also provides a cost-efficient starting point for further improvements.
5. **Updating and Refining the Initial Solution**: After constructing the initial routes, the solution is finalized by updating the routes for all vehicles across the planning horizon. This step consolidates the initial assignments into a structured

Algorithm 1 Initial Assignment Algorithm

Input: C_f, C_{nf}, V, P ▷ Frequent, Non-Frequent customers, vehicles and planning horizon

A. Frequent Customer Assignment with Randomization

```
Shuffle C_f                                  ▷ Random shuffle of frequent customers
for each c in C_f do
    cvp ← Dictionary to store costs and positions for each vehicle per period
    for each p in P do
        if c has demand in p then
            for each vehicle V compatible with the c do
                for each position in vehicle's route do
                    if customer insertion is valid at position then
                        mc ← Move cost without inserting customer
                        if mc < cvp[V]["cost"] then
                            Update cvp with the lower cost and position
                        end if
                    end if
                end for
            end for
        end if
    end for
    Calculate total insertion cost per vehicle for the customer
    Select best vehicle with the minimum total insertion cost
    Insert customer into best vehicle at the selected position per period
end for
```

B. Non-Frequent Customer Assignment with Randomization

```
Shuffle C_nf                                 ▷ Random shuffle of non-frequent customers
for each c in C_nf do
    for each p in the P do
        if c has demand in p then
            Calculate insertion costs for all compatible vehicles
            Choose randomly the best vehicle among the top-k candidates
            Insert c at the best insertion position in chosen vehicle
        end if
    end for
end for
```

Output: Initial solution with assigned routes and total cost solution, enabling a smooth transition to the optimization phase. The effectiveness of this initial solution directly impacts the performance of the Variable Neighborhood Descent (VND) algorithm in refining and enhancing route configurations.

D2. Variable Neighborhood Descent (VND) Algorithm

The core of the solution methodology is the Variable Neighborhood Descent (VND) algorithm, which iteratively explores different neighborhood structures to improve the initial solution. The VND framework is designed to start with the most complex neighborhoods, which are likely to yield significant cost reductions in the initial stages of the optimization process, and progress to simpler ones if no improvement is achieved. It is worth noting that the complexity of the neighborhoods explored can lead to larger computational times that increase exponentially with the size of the search space, i.e., as the number of customers requiring service grows, the set of possible solutions expands, resulting in a greater computational burden. This structured approach allows the VND to navigate the vast search space effectively, avoiding local optima and ensuring a thorough exploration of potential solutions.

The VND algorithm's strategic design is based on the premise that different types of neighborhood moves can yield varying levels of improvement. By starting with more comprehensive and computationally intensive neighborhoods, such as those that involve reassigning entire chains of customers between vehicles, the algorithm seeks to identify substantial cost savings early in the optimization process. If these complex moves do not lead to improvements, the algorithm progresses to less intricate neighborhoods, such as swapping or relocating individual customers, and finally to simple intra-route optimizations like 2-opt and Or-opt. This descending order of complexity ensures that larger structural adjustments are attempted before the more granular fine-tuning adjustments.

One of the key strengths of the VND framework is its adaptability. When a neighborhood successfully improves the solution, the VND resets and resumes from the most complex neighborhood. This resetting mechanism allows the algorithm to capitalize on significant cost reductions when they occur, effectively iterating through a feedback loop that adapts to the problem's dynamic nature. This design prevents the solution process from becoming prematurely fixed in a local minimum and ensures a comprehensive search for optimal or near-optimal solutions.

In order to enhance the effectiveness of the search process, the VND algorithm integrates a Tabu mechanism for certain neighborhood types, namely Change Vehicle Chain, Swap, and Relocation. The Tabu lists are employed to record moves that have been previously attempted without success, thereby discouraging the algorithm from revisiting these configurations and ensuring a broader exploration of the solution space. This combination of adaptive resets and the use of Tabu lists creates a robust search process capable of dealing with the inherent complexity of the HConVRP problem, where multiple competing factors such as customer demands, vehicle capacity, and consistency constraints must be balanced.

D3. Neighborhood Structures

The effectiveness of the VND algorithm depends heavily on the range and quality of neighborhood structures it employs. Each neighborhood structure represents a distinct type of move or adjustment to the current solution, targeting different aspects of the problem. The neighborhood operations can be broadly classified into two categories: inter-route adjustments, which involve reassigning customers between different routes or vehicles, and intra-route optimizations, which focus on refining the sequence of customer visits within a single route. The following sections provide a detailed examination of each neighborhood structure utilized in the VND algorithm, including their operational logic, rationale, and specific contributions to the overall optimization process.

1. Change Vehicle Chain Neighborhood Structure

The Change Vehicle Chain (Algorithm 3) is one of the most complex neighborhood structures, focusing on the reallocation of frequent customers across multiple vehicles. This neighborhood operates by forming a "chain" of moves, where a frequent customer is moved from its current vehicle to another, while another customer from the second vehicle may be transferred to a third vehicle, creating a cascading effect of reallocations (*Stavropoulou, F., Repoussis, P.P., Tarantilis, C.D., 2019*). Such chains allow the algorithm to explore substantial structural changes in the routing plan, potentially leading to large reductions in overall cost.

The primary rationale behind this neighborhood is to address the consistency requirements of frequent customers. These customers need to be served by the same vehicle across all periods in which they have demand. By using the Change Vehicle Chain, the algorithm can rearrange the assignment of frequent customers, ensuring that service consistency is maintained while also balancing loads across the fleet. This type of neighborhood is particularly useful when initial solutions are imbalanced, with certain vehicles being overburdened and others underutilized. However, the computational complexity of evaluating potential chains means that this neighborhood is computationally intensive, making it ideal for initial exploration phases when significant adjustments are still possible.

To enhance the efficiency of this neighborhood, a Tabu list is employed. The Tabu mechanism prevents the algorithm from repeating moves that have been attempted without success, encouraging it to explore new and varied configurations. This prevents the search from stagnating, especially in cases where simple moves may fail to disrupt the existing route structure sufficiently to escape local optima.

2. Swap Neighborhood Structure

The Swap neighborhood represents a less complex yet highly effective method of inter-route adjustment. This operation involves exchanging the positions of two customers, which can be done either within a single route (intra-route swap) or between different routes (inter-route swap). Unlike Change Vehicle Chain, which deals exclusively with frequent customers, the Swap neighborhood is applied to both frequent and non-frequent customers, allowing for a broader scope of optimization. It is, however, crucial to highlight at this point that any move(s) between customers only occur when both parties are categorised as either frequent or non-frequent customers.

The rationale for incorporating the Swap neighborhood into the VND framework is based on its capacity to rapidly balance demand across diverse routes. The process of swapping two customers can result in a reduction of travel times and an increased efficiency in the utilisation of vehicle capacities, achieved through the redistribution of loads. Furthermore, the resolution of inefficiencies resulting from the initial customer assignments, such as instances where two customers situated in close proximity are assigned to disparate vehicles, can also be achieved through the implementation of swaps. By facilitating both intra- and inter-route swaps, the algorithm is capable of adapting in a dynamic manner to accommodate fluctuations in customer demands and vehicle constraints, thus making this neighborhood a highly versatile tool within the VND's arsenal.

Algorithm 3 ChangeVehicleChain Optimization

Input: Vehicles, Frequent Customers, Planning Horizon

```
Select a random period p from the planning horizon
for each vehicle_from in vehicles do
    for each vehicle_middle in vehicles do
        for each vehicle_to in vehicles do
            // Find the best chain relocation:
            i) Evaluate moving a customer from vehicle_from to vehicle_middle
            ii) Evaluate moving a customer from vehicle_middle to vehicle_to
            iii) Calculate the total cost of both relocations
            iv) Validate if the chain relocation improves the objective function
            if the chain relocation improves the objective and is valid then
                Perform the relocation of customers between vehicles
                Update routes
            end if
        end for
    end for
end for
```

Output: Updated solution with new routes and total cost

Similarly to the Change Vehicle Chain, the Swap neighborhood also uses a Tabu list to prevent repetitive and redundant swaps that do not yield improvements. This helps to maintain diversity in the search process and drives the algorithm to explore new configurations that might lead to better solutions.

3. Relocation Neighborhood Structure

Relocation is another intra- and inter-route neighborhood that focuses on either the transfer of a single customer from one route to another, or the repositioning of that customer within the same route in order to reduce the overall transportation cost. Unlike Swap, which involves an exchange of customers, Relocation deals with a one-sided move, making it a simpler and more focused adjustment. This neighborhood is applicable to both frequent and non-frequent customers, providing a flexible means of reassigning demand across the fleet.

The primary advantage of Relocation is its ability to address imbalances within the routing plan. For example, if a vehicle is overloaded or its route is too long, relocating one of its customers to a less busy vehicle can alleviate these issues. This process can help optimize the use of available vehicle capacity, reduce operational costs, and improve the overall efficiency of the fleet. The simplicity of the Relocation move makes it an ideal candidate for situations where only minor adjustments are needed, following more substantial changes made by earlier, more complex neighborhood operations.

To maximize the efficiency of this neighborhood, the algorithm integrates a Tabu mechanism to track and avoid previous relocations that have not resulted in improvements. This ensures that the search process remains dynamic and does not become fixated on specific solutions that do not contribute to further optimization.

4. 2-opt Local Search

The 2-opt local search is a classical intra-route optimization technique that focuses on improving the sequence of customers within a single route. This operation works by removing two connections (edges) in the route and reconnecting them in a way that reduces the total distance traveled. By systematically testing different ways to reconnect the route, the 2-opt technique can significantly shorten the travel path, leading to lower transportation costs.

The rationale for using 2-opt in the VND framework is its effectiveness in fine-tuning routes after more substantial inter-route changes have been made. While it does not involve moving customers between different routes, 2-opt plays a critical role in ensuring that each individual route is as efficient as possible. It is relatively simple compared to the other neighborhoods, making it suitable for the later stages of the optimization process, where the primary focus is on refining rather than restructuring the solution.

5. Or-opt Local Search

Similar to 2-opt, Or-opt is an intra-route optimization that deals with rearranging small segments of the route. However, instead of just swapping two edges, Or-opt allows for the relocation of segments consisting of 1, 2, or 3 consecutive customers within the same route. This flexibility enables the algorithm to make more nuanced adjustments to the sequence of visits, allowing it to resolve specific inefficiencies without disrupting the entire route structure.

Or-opt's main advantage is its ability to address more subtle inefficiencies that may not be apparent with broader optimization techniques. By moving small clusters of customers, the algorithm can smooth out irregularities and ensure that each route operates at peak efficiency. As with 2-opt, Or-opt is generally applied in the later stages of the VND process, where the focus has shifted from making major structural changes to refining the solution for optimal performance.

D4. Neighborhood Exploration Strategy

The exploration of neighborhoods within the VND framework represents a fundamental aspect of the solution methodology for addressing the HConVRP with consistency constraints. The strategy adopted is designed to sequentially apply various neighborhood structures in a way that enhances the efficiency of the solution process, leveraging different types of operations to minimize total transportation costs. The flow and logic of this process are depicted in **Algorithm 2** of the report, which visually outlines the iterative and hierarchical nature of the VND algorithm.

The rationale behind the neighborhood hierarchy and exploration order is based on the objective of achieving maximum cost reductions at an early stage of the optimization process. This is achieved by initiating the process with more complex and computationally demanding moves that have the potential to yield significant improvements, followed by the application of simpler local search methods that serve to refine the solution.

Algorithm 2 VND Optimization Algorithm

```
Input: s                                                    ▷ Initial solution
s* ← s                            ▷ Save the initial solution as the best solution
k ← 0                                          ▷ Initialize neighborhood counter
improved ← True
while improved or k ≤ 5 do
    improved ← False
    if k = 0 then
        Perform Change Vehicle Chain optimization
        if Solution improves then
            Update best solution and cost (s*)
            k ← 0                                   ▷ Reset neighborhood counter
            improved ← True
        else
            k ← k + 1                   ▷ Explore the next neighborhood structure
        end if
    else if k = 1 then
        Perform Swap optimization
        if Solution improves then
            Update best solution and cost (s*)
            k ← 0
            improved ← True
        else
            k ← k + 1
        end if
    else if k = 2 then
        Perform Relocation optimization
        if Solution improves then
            Update best solution and cost (s*)
            k ← 0
            improved ← True
        else
            k ← k + 1
        end if
    else if k = 3 then
        Perform 2-opt local-search optimization
        if Solution improves then
            Update best solution and cost (s*)
            k ← 0
            improved ← True
        else
            k ← k + 1
        end if
    else if k = 4 or k = 5 then
        Perform Or-opt optimization
        if Solution improves then
            Update best solution and cost (s*)
            k ← 0
            improved ← True
        else
            k ← k + 1
        end if
    end if
Output: s*                                                ▷ Optimized solution
```

The **Change Vehicle Chain** is a priority because it allows the algorithm to address large-scale inefficiencies in vehicle assignments. Its ability to reorganize customer-vehicle relationships holistically often results in the most considerable reductions in operational costs, thus justifying its position at the start of the optimization sequence. Should the solution improve after exploring this neighborhood, the VND resets, returning to this neighborhood to capitalize on any further potential improvements.

To effectively utilize the Change Vehicle Chain exploration move without computationally straining the entire framework, the algorithm selects a random period for each run (instead of exploring all periods) and identifies the optimal frequent customer assignments and exchanges within that period, thereby reducing the overall cost of the solution. This approach enables the exploration of all periods eventually, in significantly reduced computational times.

If no improvements are made using the Change Vehicle Chain, the algorithm proceeds to the **Swap** and **Relocation** neighborhoods, respectively. These moves are less complex and computationally demanding than the initial neighborhood but are still effective in refining the solution. The decision to include both **Swap** and **Relocation** at this stage is strategic, as these operations can address both inter-vehicle and intra-route inefficiencies. They offer flexibility in exploring potential improvements without altering the fundamental structure established by the Change Vehicle Chain, thus preserving consistency for frequent customers.

As the algorithm progresses through the neighborhoods, it moves on to **2-opt** and **Or-opt** local search techniques to further optimize the given improved in-route solutions.

The hierarchy of the neighborhood structures is designed based on the potential impact each neighborhood can have on the overall solution and the computational expense associated with exploring it. The **Change Vehicle Chain** is the most comprehensive, able to reorganize the entire routing framework, which is why it is given priority. Starting with complex moves helps in addressing larger-scale inefficiencies first, and subsequent simpler moves help in local improvements.

The use of a **tabu list** in the Change Vehicle Chain, Swap, and Relocation neighborhoods further enhances the robustness of the VND algorithm by preventing the algorithm from revisiting solutions that have been previously explored, thereby reducing the risk of becoming trapped in local optima. This mechanism ensures a diversified search path, which is crucial in exploring a large solution space effectively.

E. Computational Framework and Results

A number of experiments on various dataset instances with different characteristics, were carried out to evaluate the validity and computational capabilities of the VND Optimization Algorithm presented in *Algorithm 2.* For this particular optimization methodology to work, each dataset was required to provide explicit values for the parameters presented in *Table 1* (described also in detail in paragraph **C3**).

Variable Name	**Description**
Q_k	Capacity of vehicle k.
T	Maximum Route Duration.
N_c	Set of customer nodes.
N_f	Set of frequent customer nodes, where $N_f \subseteq N_c$.
M	Set of vehicle types.
K	Set of vehicles.
P	Set of periods in the planning horizon.
z_{mip}	Binary variable indicating if vehicle m visits customer i in period p.
w_{ip}	Binary parameter indicating if customer i requires service in period p (demand of customer i in period p).
s_{ip}	Service time required for customer i in period p.
u_{kij}	Binary parameter indicating edge compatibility for vehicle k between nodes i and j.

Table 1: Dataset Input and Constraint Requirements

In her paper, 'The Consistent Vehicle Routing Problem with Heterogeneous Fleet', F. Stavropoulou undertook a diligent and praiseworthy task in creating datasets tailored to address the specific requirements of this study. Recognizing the rigor and relevance of her work, we utilized these datasets in our research with her explicit permission. Thus, in the following section we provide a brief overview of the key properties and characteristics of these datasets to illustrate their suitability for evaluating the VND Optimization Algorithm.

In contrast to the proposed HTS (Hierarchical Tabu Search) algorithmic framework by F. Stavropoulou that uses a predefined number of iterations and predefined values for the neighborhood and heuristic algorithms' shift, for each solution to be extracted; this study employs an alternative approach whereby the neighborhood and heuristic methods are subject to dynamic alteration contingent on whether the preceding iteration of the algorithm yielded an improvement or deterioration in the optimal solution that has been attained thus far.

However, the absence of a clearly defined number of iterations carries the risk of the algorithm becoming trapped in local optima when exploring potential solutions to the problem. Consequently, the incorporation of a tabu list of size $\omega_t = 10$ was considered essential in the present implementation (as in F. Stavropoulou's implementation). The size of the tabu list was determined based on empirical observations to optimize the algorithm's utility, essentially holding a short-term memory to avoid revisiting past moves.

E1. Dataset Instances

The datasets used are practically divided into 3 principal subgroups based on the proportion of frequent and non-frequent customers contained in each dataset. In the first category (*Set A*) the ratio of frequent - non-frequent customers is 85:15, which essentially means that out of all customers 85% are frequent customers who require service in more than one period of the planning horizon, while the remaining 15% are non-frequent customers who require service in only one period of the horizon.

The number of customers in this set (as well as in the rest of the datasets) varies from 50 up to 199. Furthermore, all datasets are constrained by a predefined and fixed planning horizon of five periods. This approach is designed to ensure that the solution to the problem being considered reflects realistic conditions as accurately as possible. The key differentiating factor between the datasets within a set, which share a common customer number – comprising both frequent and non-frequent customers – is the variation in the maximum duration T of each route, as well as the fixed F_k and variable costs V_k of the vehicles required to serve them.

In the second category (*Set B*), the ratio of frequent to non-frequent customers is 75:25, reflecting a balanced proportion that includes a larger presence of non-frequent customers compared to Set A. Specifically, 75% of customers in these datasets require service in more than one period of the planning horizon, while the remaining 25% require service in only one period. Similar to Set A, the datasets feature a variable number of customers, ranging from 50 to 199, ensuring the exploration of scenarios across a wide spectrum of customer loads.

The third category (*Set C*) is defined by an equal distribution of frequent and non-frequent customers, with a ratio of 50:50. This balance presents a more complex scenario, whereby half of the customers require recurring services across the planning horizon, while the other half necessitate service in only one period. This structure models a market scenario in which the priorities for customer retention and one-time engagements are equally weighted. The combination of these datasets allows for a comprehensive examination of the impact of varying customer proportions on market strategies, operational decisions, and service design.

E2. Computational Resources and Obtained Solutions

A robust computational framework was essential for implementing and testing the proposed algorithms. This section summarizes the programming tools, computational resources, and methods used to ensure accurate and reliable results. Additionally, it highlights the computational challenges encountered during the process, particularly with larger datasets, and explains how these were managed through optimization techniques and iterative testing to ensure the efficiency and scalability of the solutions.

The primary programming language employed for executing the algorithms and extracting the solutions was Python version 3.11, with the incorporation of numerous supplementary libraries, of which the most significant were pandas and numpy for data analysis. All algorithm runs and the solutions extracted were performed with the computing power of an Apple M1 processor with 8 GB RAM memory, which represented a limitation in the computational capabilities available to us. However, with the exception of datasets that contained more than 150 customers and required excessive computational time until the solution was generated, the computation time of the solutions was proportional to their complexity (which increased with the volume of customers).

The results for the initial dataset (Set A) are presented in *Table 2* below. It should be noted at this point, that the subsequent tables present the average results obtained following four runs of the algorithm for each dataset. This approach enables the calculation of both the average cost and the average time required to achieve a solution. Moreover, each table presents the optimal solution, defined as the lowest solution cost, as well as the fastest run.

From the results of Set A, it is evident that the computational time is directly proportional to the number of nodes, in this algorithmic framework. This may prove to be a disadvantage for larger instances with a greater number of nodes. Nevertheless, this shortcoming is attributable to the considerable number of calculations that are necessary to determine the optimal vehicle for each period with respect to frequent customers. As previously reported, each time a superior solution for a frequent customer is identified within a given period, the algorithm must ascertain whether the service provided by the optimal vehicle remains optimal for the remainder of the customer's required service periods. This is necessary to eventually assign the customer to that vehicle. Consequently, the computational time in this framework is increased in proportion to the number of frequent customers in the dataset relative to that of non-frequent customers.

Table 2: Summary of Computational Results for Set A (15%) instances

Instance	Nodes	Lowest Cost	Lowest Time (s)	Average Cost	Average Time (s)
15% b1	50	**11661.27**	16.11	12 863.35	261.01
15% b2	75	**13660.27**	544.70	14 695.96	803.72
15% b3	100	**11098.59**	2014.13	11 493.12	3846.60
15% b4	150	**16735.01**	9964.49	17 182.50	20 276.29
15% b6	50	**20162.29**	67.97	21 137.55	503.34
15% b7	75	**22085.21**	627.32	24 221.90	758.31
15% b8	100	**17708.53**	1165.90	18 124.31	2145.01
15% b9	150	**23911.73**	9892.60	24 218.33	13 940.08
15% b11	120	**14202.90**	5596.67	15 023.05	5879.81
15% b12	100	**14394.83**	1424.02	15 368.15	1859.03

This is explicitly evident also when observing the computational results of Set B and Set C presented in *Table 3* and *Table 4* respectively, where the execution times significantly reduce following the reduction of the proportion of frequent customers in the datasets.

Table 3: Summary of Computational Results for Set B (25%) Instances

Instance	Nodes	Lowest Cost	Lowest Time (s)	Average Cost	Average Time (s)
25% b1	50	**9836.61**	64.28	10 376.88	111.14
25% b2	75	**12761.83**	520.92	12 885.39	755.38
25% b3	100	**10373.06**	3410.89	10 657.21	5684.82
25% b4	150	**15795.01**	3768.04	16 205.12	7487.79
25% b6	50	**15781.12**	60.17	17 156.21	99.53
25% b7	75	**18187.36**	479.98	19 226.96	625.67
25% b8	100	**16796.08**	894.40	17 447.65	1233.52
25% b9	150	**22209.21**	7206.34	22 423.64	8693.07
25% b11	120	**13962.43**	1995.95	14 833.52	3568.36
25% b12	100	**14394.83**	1424.02	15 167.43	1814.45

Table 4: Summary of Computational Results for Set C (50%) Instances

Instance	Nodes	Lowest Cost	Lowest Time (s)	Average Cost	Average Time (s)
50% b1	50	**9385.83**	41.63	9708.90	24.63
50% b2	75	**11503.69**	109.27	11 809.93	147.66
50% b3	100	**9015.75**	341.95	9377.21	446.86
50% b4	150	**13218.31**	954.24	13 712.85	3060.05
50% b6	50	**14660.76**	43.98	15 682.13	34.28
50% b7	75	**14604.77**	171.20	15 769.74	273.06
50% b8	100	**14263.19**	390.06	14 590.25	436.52
50% b9	150	**20339.01**	2080.52	20 488.44	2579.13
50% b11	120	**11123.65**	385.19	12 089.38	569.80
50% b12	100	**13000.66**	187.04	13 589.24	234.86

The computational results presented allow several key conclusions to be drawn regarding the performance and effectiveness of the proposed Variable Neighborhood Descent (VND)

Table 5: Summary of Computational Results for HTS, HTS-noVND, and VND Comparisons

Instance	HTS	HTS-noVND	VND	HTS vs VND (%)	HTS-noVND vs VND (%)
15% b1	11 417.35	11 841.84	**11661.27**	−2.09	1.55
15% b2	11 338.74	12 659.15	**13660.27**	−16.99	−7.33
15% b3	10 722.74	11 358.74	**11098.59**	−3.39	2.34
15% b4	15 840.05	16 646.72	**16735.01**	−5.35	−0.53
15% b6	12 096.82	12 675.80	**20162.29**	−40.00	−37.13
15% b7	12 735.75	13 067.09	**22085.21**	−42.33	−40.83
15% b8	10 712.07	11 356.33	**17708.53**	−39.51	−35.87
15% b9	14 665.99	15 749.04	**23911.73**	−38.67	−34.14
15% b11	13 787.72	15 081.97	**14202.90**	−2.92	6.19
15% b12	13 462.23	14 430.29	**14394.83**	−6.48	0.25
25% b1	9628.27	10 219.27	**9836.61**	−2.12	3.89
25% b2	10 995.86	11 813.62	**12761.83**	−13.84	−7.43
25% b3	9873.55	10 403.28	**10373.06**	−4.82	0.29
25% b4	14 518.13	15 571.73	**15795.01**	−8.08	−1.41
25% b6	10 035.62	11 041.55	**15781.12**	−36.41	−30.03
25% b7	11 078.79	13 006.85	**18187.36**	−39.09	−28.48
25% b8	10 572.45	11 678.05	**16796.08**	−37.05	−30.47
25% b9	14 204.24	15 382.26	**22209.21**	−36.04	−30.74
25% b11	11 625.48	13 665.89	**13962.43**	−16.74	−2.12
25% b12	12 973.11	14 034.27	**14394.83**	−9.88	−2.50
50% b1	9094.71	10 178.14	**9385.83**	−3.10	8.44
50% b2	9810.23	11 236.92	**11503.69**	−14.72	−2.32
50% b3	7933.24	9311.37	**9015.75**	−12.01	3.28
50% b4	11 794.33	13 473.98	**13218.31**	−10.77	1.93
50% b6	8086.79	11 206.00	**14660.76**	−44.84	−23.56
50% b7	9324.83	11 931.87	**14604.77**	−36.15	−18.30
50% b8	8546.57	10 078.29	**14263.19**	−40.08	−29.34
50% b9	11 660.87	14 758.12	**20339.01**	−42.67	−27.44
50% b11	10 374.90	12 601.46	**11123.65**	−6.73	13.29
50% b12	10 684.56	12 787.91	**13000.66**	−17.82	−1.64

framework in comparison to the alternative frameworks (HTS and HTS-noVND) by F. Stavropoulou. These conclusions not only reflect the results but also highlight the potential strengths of the VND approach, situating it within the context of algorithmic advancements for the HConVRP.

The results demonstrate that the VND framework provides competitive and, in certain instances, superior solutions compared to the HTS-noVND framework (*Table 5*). To illustrate, in specific instances (e.g., "15% b11" and "50% b11"), the VND framework demonstrated superior performance compared to HTS-noVND, achieving up to 6.19% and 13.29% better solutions, respectively. These outcomes highlight the effectiveness of the VND framework in effectively exploring the solution space, particularly in smaller-scale or less complex instances. Also, results indicate that the proposed VND framework provides an enhanced ability to handle local optima effectively compared to the simpler Tabu-based mechanisms in HTS-noVND.

Moreover, the computational results demonstrate that the VND approach was consistently competitive across varying fleet and problem complexities (15%, 25%, and 50% fleet compositions). While it may not have consistently outperformed the HTS framework, it delivered high-quality solutions with a significant reduction in computational complexity compared to hierarchical metaheuristics. This makes VND a viable and efficient choice for practitioners handling problems with constrained resources.

While the VND approach provided feasible solutions and maintained consistency constraints, it was outclassed by HTS in both solution quality and computational times. The results indicate that VND's localized search strategy, while efficient in exploring smaller neighborhoods, lacks the hierarchical structure and global search capabilities of HTS. This limitation was particularly evident in larger problem instances, where VND struggled to match the performance and computational efficiency of HTS.

However, although HTS dominates in performance, the proposed VND framework can serve as an important baseline and provide valuable insights into the role of simpler heuristic methods. Its straightforward design and modularity make it a useful approach for exploring neighborhood structures and understanding the nuances of fleet heterogeneity and consistency constraints. Future research could build upon the foundational elements of VND to create hybrid algorithms that leverage the computational efficiency of VND and the global optimization capabilities of HTS.

E3. Custom Web Application for Solution Management and Evaluation

To enhance the usability and accessibility of the HConVRP solver and its associated functionalities, a custom web application was developed. This application serves as an interactive platform for users to generate, evaluate, and compare solutions effectively. The application consists of four core functionalities: solution generation, solution evaluation, solution comparison, and dataset management, each of which is described in detail below.

Solution Generation

The solution generation module allows users to execute the HConVRP solver algorithm with minimal effort. The user interface provides a straightforward mechanism for selecting a dataset and initiating the solver with a single click. Upon initiation, the application automatically runs the algorithm, which is underpinned by the Variable Neighborhood Descent (VND) framework.

A *detailed logging system* is integrated into this module to enhance transparency and understanding of the solution progression. This log captures essential real-time information and presents them in a visually informative way for the user. Those information include:

1. **Optimization Moves:** The specific algorithmic operations performed, including ChangeVehicleChain, Swap, Relocation, 2-opt and or-opt moves. Each operation is annotated with the affected solution components, providing users with a granular view of the optimization process. Furthermore, the number of moves applied per neighborhood exploration is also included.
2. **VND Iterations:** The number of iterations completed, offering insight into the iterative improvement methodology.
3. **Cost Progression:** Detailed records of cost improvements achieved at each iteration and period, highlighting how the algorithm converges toward an optimal or near-optimal solution.

The system outputs all generated solutions as YAML files. These files are structured according to predefined schemas, ensuring consistency and seamless interoperability with other tools for further analysis or visualization. This design choice facilitates the reproducibility of results, which is essential for academic and industrial applications. The reader of this report will find a relevant screenshot of an example HConVRP Solver run on the following page (Screenshot 1).

1.	01-12-2024 20:33:11	INITIATION	**HConVRP Solver has initiated successfully.**	Time: 0.01"	VND Iteration: 0
2.	01-12-2024 20:33:11	INITIATION	**Number of frequent customers: 25**	Time: 0.01"	VND Iteration: 0
3.	01-12-2024 20:33:11	INITIATION	**Initial Solution constructed**	Time: 0.03"	VND Iteration: 0

Step	Period 0	Period 1	Period 2	Period 3	Period 4	Total Cost
Initial	2753.39	2578.90	3380.91	2901.20	2172.11	13786.51

4.	01-12-2024 20:33:11	INFO	**Performing ChangeVehicleChain Optimization on solution with cost: 13786.51 - (Number of vehicle combinations: 135)**	Time: 0.03"	VND Iteration: 0
5.	01-12-2024 20:33:12	INFO	**Total chain relocations: 1**	Time: 0.48"	VND Iteration: 0
6.	01-12-2024 20:33:12	INFO	**ChangeVehicleChain optimization improved the solution by 326.34.**	Time: 0.48"	VND Iteration: 1
7.	01-12-2024 20:33:12	INFO	**VND Optimization - Iteration 1**	Time: 0.48"	VND Iteration: 1

Step	Period 0	Period 1	Period 2	Period 3	Period 4	Total Cost
Initial	2753.39	2578.90	3380.91	2901.20	2172.11	13786.51
ChangeVehicleChain	2709.82	2573.67	3351.33	2883.15	1942.20	13460.17

8.	01-12-2024 20:33:12	INFO	**Performing ChangeVehicleChain Optimization on solution with cost: 13460.17 - (Number of vehicle combinations: 135)**	Time: 0.48"	VND Iteration: 1
9.	01-12-2024 20:33:12	INFO	**Total chain relocations: 0**	Time: 0.75"	VND Iteration: 1
10.	01-12-2024 20:33:12	INFO	**Performing Swap Optimization with Tabu List on solution with cost: 13460.17**	Time: 0.75"	VND Iteration: 1
11.	01-12-2024 20:33:14	INFO	**Total intra-swaps: 16 | Total inter-swaps: 1 | Total frequent swaps: 3**	Time: 2.71"	VND Iteration: 1
12.	01-12-2024 20:33:14	INFO	**Swap optimization improved the solution by 1037.53.**	Time: 2.71"	VND Iteration: 2
13.	01-12-2024 20:33:14	INFO	**VND Optimization - Iteration 2**	Time: 2.71"	VND Iteration: 2

Step	Period 0	Period 1	Period 2	Period 3	Period 4	Total Cost
Initial	2753.39	2578.90	3380.91	2901.20	2172.11	13786.51
ChangeVehicleChain	2709.82	2573.67	3351.33	2883.15	1942.20	13460.17
Swap	2343.31	2425.25	3032.97	2766.49	1854.61	12422.64

Screenshot 1: Example Solver Logging

Solution Evaluation

The solution evaluation module provides users with an in-depth analysis of the results generated by the solver. Each solution is presented with comprehensive metadata and performance metrics that are critical for assessing its quality and feasibility. The evaluation page is divided into the following key sections:

1. **Validation Criteria:** The application systematically checks and reports whether the solution satisfies all problem constraints, including:
   - Vehicle capacity and duration adherence.
   - Ensuring each customer is visited once and serviced within their requirements.
   - Consistency for frequent customers, including adherence to the same vehicle and driver across periods.
   - Depot start and return compliance.
2. **Quantitative Metrics:** Detailed numerical outputs are provided, such as:
   - **Total Cost of the Solution:** The cumulative cost of the routes across all periods.
   - **Evaluated Cost:** A recalculated total cost after applying all constraints, ensuring accuracy in the final result.
   - **Computational Time:** The runtime required to generate the solution, measured in seconds, providing an empirical benchmark for performance evaluation.

3. **Optimization Process Visualization:** A stepwise breakdown of cost improvements is presented in tabular format, illustrating the manner in which each move reduced the overall cost across different periods. For example, users may observe how a swap operation reduced the cost in a specific period, or how a ChangeVehicleChain adjustment optimized the route network.

4. **Graphical Solution Representation:** The solution can be illustrated as a network graph in which customers or depots are represented by nodes and routes between them are represented by edges. Nodes are geospatially plotted using their coordinates, thus enabling users to understand the geographical distribution of the solution. This visual representation is particularly beneficial in the case of large datasets, where textual or

tabular formats are insufficient for grasping the underlying structure of the solution. All directed solution graphs are generated with the help of Python's NetworkX library for graph visualization.

Solution Comparison

The solution comparison module allows users to assess multiple solutions simultaneously, offering insights into their relative performance. This feature is particularly useful for benchmarking or selecting the most suitable solution for a specific operational context. The comparison results are presented in two primary dimensions:

1. **Cost Comparison:** Solutions are ranked based on their total costs. A tabular summary displays:
   - The total cost of each solution.
   - Statistical measures such as the average and median costs, which provide additional context for decision-making.
2. **Computational Time Analysis:** Similar to cost, solutions are ranked by computational time. This allows users to assess trade-offs between solution quality and computational efficiency, which is critical in time-sensitive or resource-constrained environments.

The integration of statistical summaries enhances the interpretability of the results, enabling users to identify trends or anomalies in solution performance.

Dataset Viewer

The dataset viewer module is designed to provide users with an interactive and intuitive means of exploring the datasets used for solution generation. This module incorporates:

1. **Interactive JSON Viewer:** Users can access and inspect the full properties of any dataset, presented in a hierarchical and interactive format. This feature ensures that users can examine critical dataset attributes, such as:
   - Customer locations and demand levels.
   - Vehicle types, capacities, and constraints.
   - Temporal requirements for service consistency.

2. **Data Validation:** The application performs automated checks to ensure dataset integrity, such as verifying that all required attributes are present and correctly formatted.

The custom web application represents a significant advance in the operationalization of the HConVRP solver. The application's modular design and comprehensive functionality render it a potent instrument for both academic research and practical applications in logistics and transportation management. By integrating solution generation, evaluation, comparison and dataset management into a unified platform, the application enhances the user experience while upholding the rigour and transparency essential for solving intricate optimization problems. Potential future enhancements may encompass the incorporation of supplementary visualization tools and the expansion of the algorithm library to address evolving vehicle routing challenges.

F. Conclusions

In summary, this research addresses the complexities and intricacies of the Heterogeneous Consistent Vehicle Routing Problem (HConVRP), a critical variant of the traditional Vehicle Routing Problem (VRP) that integrates the dual challenges of fleet heterogeneity and service consistency over multiple time periods. Addressing the need for optimized routing solutions in logistics and transportation, particularly in industries where customer satisfaction and operational efficiency are paramount, this study contributes to the existing body of knowledge with a robust Variable Neighborhood Descent (VND) algorithm. The algorithm not only seeks to minimize the transport costs associated with a diverse fleet, but also ensures that service delivery remains consistent, thereby enhancing customer relationships and operational reliability.

The results underscore the importance of considering both fleet characteristics and customer service requirements in the routing process, highlighting the interplay between cost efficiency and service quality. Through thorough computational testing on different datasets, the VND framework demonstrates competitive performance against established methodologies, demonstrating its potential as a practical tool for practitioners in the field.

Furthermore, the development of a custom web application facilitates the accessibility and usability of the HConVRP solver, encouraging further exploration and application of the proposed solutions in real-world scenarios.

Future research directions may include the exploration of hybrid algorithms that combine the strengths of the VND approach with other optimization techniques to improve solution quality and computational efficiency. Overall, this study not only fills a critical gap in the vehicle routing literature, but also provides actionable insights for logistics operations seeking to balance cost and service consistency in increasingly complex environments.