INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. xx (2025) 1–31
DOI: xx.xxxx/itor.xxxxx

A Local Search metaheuristic for the last-mile Vehicle Routing Problem with Delivery Options

Ioannis Kitsos Kalyvianakis[a], Angelos Omirolis[a] Gerasimos Anastasios Athanasiadis[a] Eleftherios G. Manousakis[a, *] and Emmanouil E. Zachariadis[a]

[a]*Department of Management Science and Technology, School of Business, Athens University of Economics and Business, Athens, Greece*
E-mail: ikitsoskal@aueb.gr [I. Kitsos Kalyvianakis]; aomirolis@aueb.gr [A. Omirolis]; gathanasiadis@aueb.gr [G. Athanasiadis]; lmanousakis@aueb.gr [E. Manousakis]; ezach@aueb.gr [E. Zachariadis]

Abstract

This paper introduces a novel Local Search matheuristic for the Vehicle Routing Problem with Delivery Options (VRPDO). The VRPDO extends the Generalized Vehicle Routing Problem with Time-Windows (GVRPTW) by incorporating shared-location capacities and mandatory service-level constraints. The customer requests can be shipped to alternative locations with varying time windows, and the carrier must select one option to minimize cost while respecting customer service-level preferences and shared-location capacities. The problem requires the minimization of the number of vehicles first, and then the total routing cost. The proposed approach is a multi-restart local search metaheuristic featuring a randomized greedy insertion heuristic and a core algorithm guided by a promise-based tabu mechanism and a route utilization metric designed specifically to promote route consolidation. The matheuristic integrates five neighborhood operators, including moves for adjusting customer-option assignments. An extensive computational study, including detailed parameter tuning, is performed on two major VRPDO benchmark sets. The proposed approach manages to improve the best known solutions for 38/120 and 42/120 instances, respectively.

Keywords: transportation; routing; time-windows; delivery options; last-mile; local search

1. Introduction

The exponential growth of e-commerce has fundamentally transformed global retail landscapes and logistics operations. In 2024, global retail e-commerce sales reached an estimated six trillion US dollars, with projections indicating a 31% growth over the coming years, expecting to approach eight trillion dollars by 2028 (Statista, 2024). This unprecedented expansion represents a shift in consumer behav-

*Author to whom all correspondence should be addressed (e-mail: lmanousakis@aueb.gr).

ior, with e-commerce accounting for an increasingly significant portion of total retail transactions. The magnitude of this transformation is further evidenced by the fact that over 2.14 billion people worldwide engage in online shopping, demonstrating the widespread adoption of digital commerce platforms across diverse geographic regions and demographic segments.

The surge in e-commerce volumes has created substantial challenges for last-mile delivery operations, particularly concerning environmental sustainability and operational efficiency. Last-mile delivery, representing the final stretch of a parcel's journey from distribution centers to customers, contributes approximately 30-40% of all e-commerce-related carbon emissions, making it a critical focus area for sustainable logistics practices (Last Mile Experts, 2024). Traditional home delivery methods face significant operational challenges, including first-time delivery failure rates that can reach 25%, resulting in costly redelivery attempts and increased environmental impact. In response to these challenges, alternative delivery methods have gained considerable traction, with parcel lockers emerging as a particularly popular solution. By 2020, over one-third of online purchases were collected using parcel lockers worldwide, indicating strong consumer acceptance of out-of-home delivery options.

The Vehicle Routing Problem with Delivery Options (VRPDO) was introduced to address these complex last-mile delivery challenges by incorporating multiple delivery alternatives into traditional routing optimization models. The VRPDO extends the classic Vehicle Routing Problem with Time Windows (VRPTW) by allowing customers to specify multiple delivery options with different locations, time windows, and priority levels, enabling logistics providers to optimize routes while accommodating customer preferences and operational constraints. This optimization framework captures real-world delivery scenarios where customers can choose between home delivery, workplace delivery, parcel lockers, or other pickup points, each with distinct operational characteristics and customer satisfaction levels. The implementation of VRPDO solutions enables logistics companies to reduce delivery costs, minimize environmental impact through more efficient routing, and improve customer satisfaction by providing flexible delivery options that align with modern consumer preferences for convenience and sustainability.

The main contributions of this paper are summarized as follows. First, we propose a new algorithm for solving the VRPDO. The algorithm is simple yet effective, delivering competitive results with minimal parameter tuning. To account for the specific characteristics of the problem, we design two tailored neighborhood operators that enable flexible transitions between different customer options. To handle the bi-objective nature of the problem, we introduce a penalized objective function incorporating a route utilization metric that encourages load consolidation and more efficient use of vehicle capacity. Finally, the proposed method matches the best-known solutions for 54 out of 120 small and medium instances and 5 out of 120 medium-large instances, while improving 38 out of 120 and 59 out of 120 best-known solutions in the two benchmark sets, respectively.

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature on the problem under study. Section 3 presents the mathematical formulation of the Vehicle Routing Problem with Delivery Options. Section 4 presents the proposed optimization framework and details its algorithmic components. Section 5 reports the results of extensive computational experiments, including parameter tuning and performance evaluation. The obtained solutions are compared against state-of-the-art results from the literature on two benchmark instances. Finally, Section 6 provides concluding remarks and outlines potential directions for future research.

2. Literature

The Vehicle Routing Problem with Delivery Options (VRPDO) is an advanced variant of the classic Vehicle Routing Problem (VRP) that addresses the complexities of modern last-mile logistics. It includes time windows and extends traditional routing problems by incorporating multiple delivery options for customers, resource synchronization, and service-level constraints. This flexibility makes VRPDO particularly suitable for modeling real-world last-mile delivery scenarios such as urban delivery and healthcare logistics.

2.1. Origins of VRPDO: VRPTW and GVRPTW

The VRPDO is rooted in the Vehicle Routing Problem with Time Windows (VRPTW), introduced by Savelsbergh (1985). The VRPTW extends the classical VRP by adding time window constraints for customer deliveries, making it more applicable to real-world logistics. Over the years, the VRPTW has been extensively studied with numerous heuristics and exact methods being proposed, as reviewed by Bräysy and Gendreau (2005) and Vidal et al. (2013). The Generalized Vehicle Routing Problem with Time Windows (GVRPTW) (Moccia et al., 2012), is a direct precursor to the VRPDO. The GVRPTW allows customers to be served at multiple locations, but does not consider shared delivery locations or resource synchronization. It can be seen as a special case of VRPDO without shared location and with a single preference per customer. Even close to the concept of the VRPDO, lies the Vehicle Routing Problem with Multiple Time Windows (VRPMTW) (Favaretto et al., 2007). The model supports multiple time windows for the same location and can be seen as a special case of the VRPDO and of the GVRPTW, where all options of a customer take place at the same location. The objective is to minimize the total travel distance or travel duration (total travel, service, and waiting time) and a fixed cost incurred per vehicle used.

2.2. Delivery options

One of the defining features of the VRPDO is the inclusion of multiple delivery options for customers. These options can include home delivery, parcel lockers, shared delivery locations, and even roaming delivery locations (e.g., delivery to a customer's car trunk). Shared delivery locations, such as parcel lockers and shops, have gained significant attention in recent years due to their potential to reduce last-mile delivery costs and improve efficiency (Savelsbergh and Van Woensel, 2016; Zhang and Lee, 2016).

Customers increasingly prefer alternative delivery locations such as parcel lockers, shared pickup points, and even trunk delivery due to the added convenience and flexibility they offer (Yuen et al., 2019). Research shows that convenience, along with reliability and security, is a primary driver of preference for these options (Tsai et al., 2024), as they let shoppers collect packages on their own schedule without having to wait at home. Such self-service solutions also add value by involving customers directly in the delivery process, enriching the overall e-Commerce experience (Vakulenko et al., 2018). Reflecting their popularity, pickup point networks used to handle roughly 20% of household parcels in France during 2012 (Morganti et al., 2014). Beyond convenience, these out-of-home delivery methods help reduce failed home deliveries (and costly re-delivery) through consolidated drop-offs, which in turn lowers delivery costs and environmental impact by cutting unnecessary trips (Rai et al., 2020).

With respect to modeling approaches, Mancini and Gansterer (2021) introduced a model where customers can choose between home delivery and shared delivery locations, with monetary compensation for using the latter. This model was further expanded by Grabenschweiger et al. (2021), who took into account parcel sizes and locker slot capacities. Other studies have explored innovative delivery concepts, such as reception boxes, controlled access systems, and trunk delivery (Felch et al., 2019). Most recently, Janinhoff et al. (2024) present a data-driven framework for evaluating various business models on a multitrip vehicle routing problem with delivery options and location-dependent costs and provide a heuristic solution using adaptive large neighborhood search. The study examines real-life instances from a major European parcel service, showing that delivery costs can be mitigated by consolidating orders in pickup stations.

Of course, the concept of alternative service delivery choices with respect to location has been studied in other models. For instance, Baldacci et al. (2010) study applications of the Generalized Vehicle Routing Problem (GVRP). The GVRP is an extension of the classical Vehicle Routing Problem (VRP) in which the vertex set is partitioned into clusters and vehicles must visit exactly one (or at least one) vertex per cluster. In general the objective for such problems is to select locations to visit and identify routes between these locations, such that requirements are met at minimum cost or such that profit is maximized. The concept of locations choice to maximize profit with respect to operational constraints can be found in the Team Orienteering Problem (TOP) (Vansteenwegen et al., 2011b), the Set Orienteering Problem (Archetti et al., 2018) and many others. More recently, location choice has also been investigated in time-critical humanitarian settings. Kılıç et al. (2025) propose a novel post-disaster covering tour formulation has been proposed that jointly determines which locations to visit, assigns vehicles to selected relief centers, and satisfies unvisited-node demand through time-dependent transfers from nearby visited nodes. This integration of routing, location selection, and demand-transfer mechanisms highlights the growing relevance of flexible service-delivery structures, particularly under resource-constrained and time-sensitive operational conditions.

2.3. Resource synchronization and service levels

Resource synchronization is a critical aspect of the VRPDO, ensuring that delivery resources (e.g., parcel lockers, vehicles, and personnel) are used efficiently. Unlike traditional VRPs, where resources are used temporarily, the VRPDO requires permanent resource allocation throughout the delivery process. For example, parcel lockers must be managed to avoid exceeding their capacity, and delivery options must be synchronized with customer preferences and time windows.

Drexl (2012) highlighted the importance of resource synchronization in vehicle routing problems, noting that few studies had addressed this issue prior to Hempsch and Irnich (2008). Recent work by Grangier et al. (2021) and Froger et al. (2017) has explored synchronized resource constraints in the context of loading and unloading operations at satellite facilities. In VRPDO, resource synchronization is modeled through constraints on locker capacities and customer preferences, ensuring that delivery options are feasible and efficient.

Service levels, which are intimately tied to customer preferences, further complicate the model. For example, while home deliveries may be subject to strict time windows, alternative locations might operate under capacity constraints or provide flexibility with different cost structures. The integration of such constraints not only improves the realism of the model but also offers insights into trade-offs between operational efficiency and customer satisfaction.

In related contexts, Souffriau et al. (2013) extend these ideas to a tourism planning problem, the Multiconstraint Team Orienteering Problem with Multiple Time Windows (MC-TOP-MTW), where resource synchronization (in terms of time budgets and visit limits) plays a central role in route planning. Such methodologies provide a useful perspective on how synchronized resources can be managed within VRPDO frameworks.

2.4. *Roaming delivery and beyond*

The flexibility of the VRPDO framework enables it to encompass several other variants of the vehicle routing problem. For instance, The VRP with Roaming Delivery Locations (VRPRDL), introduced by Reyes et al. (2017) and further developed by Ozbaygin et al. (2017) and Lombard et al. (2018), is a specialized variant of the VRPDO. Unlike traditional vehicle routing problems, where deliveries are made to fixed locations, VRPRDL considers deliveries to moving vehicles, such as cars or autonomous vehicles, based on their known schedules and planned routes. This dynamic approach aims to enhance flexibility and convenience, particularly for recipients who are constantly on the move. However, despite its advantages, VRPRDL presents significant challenges, particularly in terms of customer acceptance. One of the key concerns, as highlighted by Felch et al. (2019), is privacy. Customers may be hesitant to share real-time location data or travel schedules due to potential risks related to surveillance, data security, and unauthorized tracking.

Additional variants include the Capacitated VRP with Pick-up and Alternative Delivery (CVRPPAD) by Sitek and Wikarek (2019) and the Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options (MD-2EVRP-DO) by Zhou et al. (2018). These studies extend the concept of multiple delivery options by introducing capacity constraints or multi-depot configurations, and they model customer preferences via penalties or other cost adjustments. In each case, the core challenge remains the effective integration of location choices and service-level constraints within a unified routing framework. Studying a variant of the Clustered Travelling Salesman Problem with Time Windows, (Nguyê n et al., 2019) focus on last-mile delivery operations in London. They propose a two-level parcel distribution model, combining driving and walking while optimizing parking decisions that yields over 20% savings in total operation time compared to conventional methods.

2.5. *VRPDO*

The concept of VRPDO was firstly introduced by the dissertation of Cardeneo (2005), where a basic version of the problem with alternative delivery locations for each customer is proposed. This foundational work was later expanded by Los et al. (2018), who integrated service levels and customer preferences into a generalized pickup and delivery problem with time windows. Since its introduction, VRPDO has been recognized for its ability to model innovative delivery systems, such as parcel lockers, shared delivery locations, and roaming delivery options. However, despite its potential, it has not been widely explored in the literature, with only a limited number of publications addressing its capabilities.

In 2021, Tilk et al. (2021) introduce and analyze the VRPDO in the context of reducing delivery failures in e-commerce. The authors propose a branch-price-and-cut algorithm to solve the problem exactly, focusing on minimizing the carrier's overall cost while respecting customer preferences and shared location capacities. The authors compare two alternative modeling for the subproblem pricing problem and provide benchmarks and several optimal solutions. At the same time, by Dumez et al. (2021a), propose a Large Neighborhood Search (LNS) algorithm that combines ruin-and-recreate operators with a set partitioning approach to reassemble routes for tackling the VRPDO. The method improves the known solutions from the literature on particular cases of the problem under consideration, such as the Vehicle Routing Problem with Time Windows (VRPTW), the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) and the Vehicle Routing Problem with Home and Roaming Delivery Locations (VRPHRDL). For the VRPDO the authors provide solutions for new instances up to 400 customers. Simultaneously, Dumez et al. (2021b) extend previous work by proposing an improved LNS scheme with two exact components: first, a set-partitioning formulation for combining previously found routes to new solutions and second, the Balas-Simonetti neighborhood for improving already good solutions. The approach is benchmarked on several vehicle routing problems with time windows such as the GVRPTW, the VRPRDL, VRPHRDL, the VRPMTW and the VRPDO.

3. Vehicle Routing Problem with Delivery Options

The Vehicle Routing Problem with Delivery Options (VRPDO) is a short-term delivery planning problem, typically formulated over a single operational day. In this setting, each customer can specify multiple potential delivery options, where each option is characterized by three attributes: a delivery location, a preference level, and a service duration. Delivery locations may correspond either to individual addresses (e.g., home deliveries) or shared facilities (e.g., parcel lockers or collection points). Each location is associated with a service time window and an additional preparation time to account for parking, unloading, or access operations. While individual delivery points can accommodate only a single option and are generally restricted by narrow time windows, shared locations can handle multiple customers and typically offer wider temporal flexibility. These shared facilities, however, are also subject to capacity constraints that limit the number of deliveries executable within the planning horizon.

The model integrates a preference-based service system. The customers rank their delivery options according to preference levels. Service quality is evaluated through service levels, defined as the proportion of customers receiving service through an option of a given preference level or better, with each level subject to a minimum coverage requirement. The fleet is assumed to be homogeneous and to operate from a central depot, with known travel times and routing costs between all service locations.

The objective of the VRPDO is to determine optimal vehicle routes and assign exactly one delivery option to each customer, while satisfying several operational constraints, including vehicle capacity, location capacity, time window feasibility, and the single-service requirement (i.e., no split deliveries). The optimization is performed in lexicographic order, firstly minimizing the number of vehicles used and subsequently the total transportation cost.

For completeness, we restate the mathematical formulation of the VRPDO following the framework introduced by Dumez et al. (2021a). Let N denote the set of customers and O the set of all available options. Each customer $c \in N$ is associated with a subset of options $O_c \subset O$, such that every option belongs exclusively to one customer, i.e., $\bigcup_{c \in N} O_c = O$. Let L represent the set of locations, and for each location $l \in L$, define $O_l \subset O$ as the subset of options that occur at location l.

The VRPDO is defined on an option-based complete graph $G = (V, A)$. The vertex set $V = O \cup 0, 0'$ includes one vertex per option, in addition to the starting depot 0 and the ending depot $0'$. For every arc $(i, j) \in A$, $t_{i,j}$ denotes the travel time, and $c_{i,j}$ represents the corresponding travel cost. Each vertex $i \in V$ has an associated service duration s_i, which captures the time required to perform the service corresponding to option i. Furthermore, each option $i \in O$ is constrained by a time window $[a_i, b_i]$, determined by its associated location.

Let P denote the set of preference levels. For each level $p \in P$, β_p represents the minimum proportion of customers that must be served with an option of preference level p or better. Each location $l \in L$ has a limited capacity C_l, defined as the maximum number of options that can be executed at that site. Every option $o \in O$ is associated with a preference level p_o and a demand q_o, where q_o corresponds to the demand of the customer linked to that option. The fleet consists of homogeneous vehicles, each with capacity Q. Let K be the set of vehicles.

For the decision variables, let $x^k_{i,j}$ be a binary variable equal to 1 if vehicle $k \in K$ traverses arc $(i, j) \in A$, and 0 otherwise. Similarly, define y_o as a binary variable equal to 1 if option $o \in O$ is selected for service. The variable h^k_i represents the service start time of vehicle k at vertex $i \in V$.

The mathematical model is formulated as follows:

$$\textbf{lex} \ \min(z_1, z_2) \tag{1}$$

$$z_1 = \sum_{j \in V} x^k_{0,j} \tag{2}$$

$$z_2 = \sum_{k \in K} \sum_{(i,j) \in A} c_{i,j} x^k_{i,j} \tag{3}$$

$$\sum_{o \in O_c} y_o = 1 \qquad \forall c \in N \tag{4}$$

$$\sum_{o \in O} \sum_{(i,o) \in A} x^k_{i,o} q_o \leq Q \qquad \forall k \in K \tag{5}$$

$$\sum_{o \in O_l} y_o \leq C_l \qquad \forall l \in L \tag{6}$$

$$\sum_{o \in O | p_o \leq p} y_o \geq \beta_p |N| \qquad \forall p \in P \tag{7}$$

$$\sum_{(i,o) \in A} \sum_{k \in K} x^k_{i,o} = y_o \qquad \forall o \in O \tag{8}$$

$$\sum_{j \in V} x^k_{0,j} = 1 \qquad \forall k \in K \tag{9}$$

$$\sum_{i \in V} x^k_{i,0'} = 1 \qquad \forall k \in K \tag{10}$$

$$\sum_{(i,j) \in A} x^k_{i,j} - \sum_{(j,i) \in A} x^k_{j,i} = 0 \qquad \forall k \in K,\ j \in O \tag{11}$$

$$x^k_{i,j}(h^k_i - h^k_j + t_{i,j} + s_i) \leq 0 \qquad \forall k \in K,\ (i,j) \in A \tag{12}$$

$$a_i \leq h^k_i \leq b_i \qquad \forall k \in K,\ i \in V \tag{13}$$

$$x^k_{i,j} \in \{0,1\} \qquad \forall k \in K,\ (i,j) \in A \tag{14}$$

$$y_o \in \{0,1\} \qquad \forall o \in O \tag{15}$$

$$h^k_i \geq 0 \qquad \forall k \in K,\ i \in V \tag{16}$$

The objective function (1) calls for the lexicographic minimization of firstly the number of vehicles utilized (2) and secondly the total transportation costs (3). Constraints (4) guarantee that exactly one option is selected for each customer. Constraints (5) enforce the vehicle capacity restriction, ensuring that the total demand assigned to a route does not exceed the vehicle capacity Q. Constraints (6) impose location capacity limits, restricting the number of options that can be executed at a given location. Constraint (7) ensures service level compliance, requiring that at least a fraction β_p of customers are served by an option of preference level p or better. Constraints (8) link the selection and routing decisions by ensuring that each served option is included in exactly one route. Constraints (9) and (10) ensure that every vehicle route begins and ends at the depot, while constraints (11) impose flow conservation, ensuring route continuity. Constraint (12) guarantees that travel and service times are respected between consecutive visits, and constraints (13) enforce time window constraints. Finally, constraints (14)–(16) define the binary and non-negativity domains for the decision variables.

4. Methodology

This section presents the proposed optimization algorithm for the VRPDO. The overall scheme is firstly outlined. Then the various components of the algorithm are discussed separately: the constructive algorithm, the local search framework (including the route utilization metric for minimizing the number of used vehicles), the tabu policy, and the local search operators.

4.1. Overview

The proposed VRPDO metaheuristic follows a multi-restart scheme. In each restart an initial solution is constructed by a minimum insertion algorithm that, at every step, builds a fixed-size Restricted Candidate List (RCL) to ensure diversification. The solution is allowed to be infeasible with respect to service levels as the local search is able to restore feasibility. Then, the initial solution is iteratively improved by the local search framework, which employs several neighborhood operators to explore the solution space. A key aspect of the search is the route utilization metric, which guides the local search towards solutions that minimize the number of vehicles used. The search process is controlled by a promise implemented tabu policy to prevent cycling and encourage diversification. The algorithm terminates upon reaching a predefined criterion, such as a maximum number of iterations or a time limit.

Algorithm 1 Overall Scheme

$S^* \leftarrow \emptyset$
for $i \leftarrow 1$ to *restarts* **do**
    $S_i \leftarrow \mathit{MinimumInsertion}()$
    $S_i^* \leftarrow \mathit{LocalSearch}(S_i)$
    if $Z(S_i^*) < Z(S^*)$ **then**
        $S^* \leftarrow S_i^*$
    end if
end for
return S^*

The overall process is outlined in Algorithm (1). At restart i, S_i denotes the constructed solution, S_i^* its locally improved version and S^* the incumbent best; $Z(\cdot)$ is the objective value.

4.2. Initial solution construction

The initial solution is generated using a construction heuristic that iteratively builds routes by inserting options for still unserved customers. At each iteration, all time-window and capacity feasible insertions (i.e., moves that respect only the time windows and capacity constraints) are generated. These candidate moves are then sorted incrementally based on their contribution to the overall objective cost.

To balance the greedy nature of the search with the need for diversification, a randomized selection strategy is applied. Specifically, the best $k = 3$ insertions (or all if fewer exist) are inserted into a Restricted Candidate List (RCL). A single move is then selected uniformly at random from this list and applied to the current solution. This approach effectively captures the diversification benefits of a randomized threshold without necessitating extensive parameter tuning.

The initial solution generation deliberately relaxes the service-level constraints to guarantee that a feasible starting solution can be produced. Without this relaxation step, ensuring consistent feasibility during construction becomes considerably challenging. In addition, the heuristic is allowed to utilize an effectively unlimited number of vehicles, which yields a solution that is not fleet-efficient but can be obtained rapidly. This strategy is justified because the subsequent local search phase is specifically designed to (i) consolidate routes to reduce the number of vehicles, and (ii) resolve any service-level violations while simultaneously improving routing costs. Consequently, the local search procedure compensates for the relaxed nature of the initial construction and progressively drives the solution toward feasibility and efficiency.

4.3. Local Search

The central component of the proposed optimization approach is the local search procedure, whose objective is to iteratively enhance and restore feasibility to the initial solution produced by the constructive heuristic. The procedure relies on three fundamental elements: a route utilization metric, a set of neighborhood search operators, and a tabu mechanism that guides the exploration process.

Algorithm 2 begins by initializing all relevant variables (line 1). It then executes an iterative improvement loop that terminates when one of two stopping criteria is met (line 2): (i) the maximum number of iterations (*max_iterations*) is reached, or (ii) no improvement in the incumbent solution has been observed for a predefined number of iterations (*max_iterations_not_improved*). At each iteration, the neighborhoods of the current solution are examined with respect to the tabu policy, i.e., promise mechanism (Section 4.3.5) and a move is selected (line 3). All moves are evaluated with respect to the original objective function and a metric mechanism promoting route minimization Section 4.3.1). The structure of these neighborhoods, as well as the selection strategy, are described in Section 4.3.2. The chosen move is then applied to the current solution and the promises are updated accordingly (lines 4-5), producing a new candidate. If this results into a new best solution the best solution is updated (line 7).

4.3.1. Route utilization metric

The VRPDO has a lexicographical objective: the minimization of the total number of active routes takes precedence over the minimization of travel cost. To enforce this behavior, a Route Utilization Metric (U_S) is employed within the local search phase to strategically reward solutions that promote fleet

Algorithm 2 *Local Search*

$S^* \leftarrow S, iter \leftarrow 0, nonImpIter \leftarrow 0, promises \leftarrow \emptyset$
while $MaxIterNotReached(iter)$ & $MaxNonImpIterNotReached(nonImpIter)$ **do**
  $move \leftarrow$ *ExploreNeighborhoods(S, promises)*
  $S \leftarrow$ *ApplyMove(S, move)*
  $promises \leftarrow UpdatePromises(move)$
  if $Z(S) < Z(S^*)$ **then**
    $S^* \leftarrow S$
    $nonImpIter \leftarrow 0$
  end if
  $iter \leftarrow iter + 1$
  $nonImpIter \leftarrow nonImpIter + 1$
end while
return S^*

consolidation. This metric is active until the number of routes in the incumbent solution reaches the theoretical minimum route count for the given total demand, computed as $\lceil \text{Total Demand}/Q \rceil$, where Q is the vehicle capacity. During this phase, the metric strategically rewards efficient moves and aggressively promotes route consolidation.

For any route $k \in K$ in a solution S with capacity Q and total load $\text{Load}_k = \sum_{o \in O} \sum_{(i,o) \in A} x^k_{i,o} q_o$, the route contribution U_k is defined based on the unutilized capacity:

$$U_k = (Q - \text{Load}_k)^\gamma, \quad \text{where } \gamma > 0 \text{ (typically } \gamma = 2) \tag{17}$$

The total metric for a solution S is the sum of the individual route contributions:

$$U_S = \sum_{k \in K} U_k. \tag{18}$$

During the local search, the acceptance criteria for a candidate move S' from an incumbent solution S is affected by a ratio applied as a multiplier to the move's change in routing cost:

$$\text{Ratio} = \frac{U_{\text{S}}}{U_{\text{S'}}}, \tag{19}$$

A candidate sulution S' that successfully increases the overall metric (i.e., $U_{S'} > U_S$) will result in a $ratio < 1$. This ratio is multiplied by the cost of the move, making consolidation moves highly favored even if they slightly increase the travel cost, thereby strictly prioritizing fleet reduction.

Consider a vehicle capacity $Q = 100$ and $\gamma = 2$. Suppose a solution S with balanced loads 50 and 50 and a move that constructs solution S' with one nearly full route and one nearly empty route (loads 90 and 10), a favorable precursor to eliminating the empty route.

The metrics are calculated as follows:

- $U_S = (100 - 50)^2 + (100 - 50)^2 = 5,000.$

- $U_{S'} = (100 - 90)^2 + (100 - 10)^2 = 8,200.$

In this specific example, $U_{S'} > U_S$, resulting in a Ratio $= 8,200/5,000 = 0.609$. This demonstrates that the metric encourages the near-full and near-empty routes. Once the solution reaches the theoretical minimum route count (1 in the second example), the metric U_S is disabled, and the search evaluates moves solely based on routing cost.

4.3.2. Local search operators

At each iteration of the local search, the neighborhoods of the following five operators are exhaustively explored:

1. **Swap:** Exchanges the positions of two options, either within the same route or across different routes.
2. **Relocate:** Removes one option from its current position and inserts it into a different position, within the same route or another route.
3. **Two-Opt:** Reverses a subsequence of visits within a route or across two routes to reduce travel distance and eliminate route crossings.
4. **Flip:** Replaces a selected option with an alternative option associated with the same customer.
5. **Priority Swap:** Simultaneously replaces two selected options with their respective alternative options associated with the same customers.

It is important to distinguish the nature of the first three operators from that of the latter two. Specifically, the first three operators—*swap*, *relocate*, and *two-opt*—modify only the route structure while keeping the selected option for each customer fixed. These moves therefore perform intensification, exploring routing-based neighborhoods as usually seen in classical VRP settings.

In contrast, the *flip* and *priority swap* operators modify the underlying problem network by changing the selected delivery options for one or more customers. Since each option corresponds to a distinct location and time window (and potentially priority level), applying these operators alters travel distances, time-window constraints, and service-level feasibility. As such, each of these moves effectively triggers a re-evaluation of a new CVRPTW instance, shifting the search landscape beyond pure route refinement.

At each iteration, all five neighborhood structures are evaluated in parallel, and the move yielding the lowest resulting solution cost is executed. Consequently, option-selection moves compete directly with routing moves, rather than being applied in a secondary or repair phase. Usually, after an option related operator changes the options, the routing operators yield better costs for the subsequent iterations until a routing local optimum is reached. This allows time for the routing operators to optimize solutions before experimenting with another option set.

To mitigate stagnation, three diversification parameters are used. Let I^{div}_{stagn} denote the iteration limit without improvement that triggers diversification, p_{div} the probability of performing a diversification step, and I^{stop}_{stagn} the maximum stagnation threshold after which the search terminates and restarts. If no improvement has been observed for I^{div}_{stagn} iterations, a Bernoulli trial with probability p_{div} is performed. Upon success, the algorithm forces a *flip* move —even if it worsens the objective— to perturb the option configuration and escape local minima. The search then resumes until improvement occurs or the hard stagnation limit I^{stop}_{stagn} is reached.

4.3.3. Flip operator

As described above, the *flip* operator selects a delivery option in the current solution belonging to a customer with multiple eligible alternatives. It then evaluates whether assigning the customer to a different location and service configuration may yield a better solution. Beyond direct cost improvements, the *flip* operator plays a crucial role in feasibility management, enabling the search to modify service assignments in ways that route-based operators alone cannot.

A central requirement of the VRPDO is satisfying the minimum service level guarantees: at least β_1 of customers must receive first-priority service and β_2 must receive second-priority service. The *flip* operator supports this requirement by allowing reassignment to higher-priority options when beneficial. Specifically, when the solution violates service level constraints, only *flip* moves that strictly improve the overall service level are permitted. Thus, the operator functions as a targeted repair mechanism that incrementally drives the search toward feasibility.

When service level constraints are already satisfied, the mechanism enforcing service-level-improving moves is deactivated. In this phase, *flip* moves are evaluated purely on their contribution to the objective function, with the additional safeguard that only moves preserving feasibility are considered. In this way, the *flip* operator transitions seamlessly from feasibility restoration to cost-focused optimization, providing flexibility to reshape the underlying problem instance when beneficial.

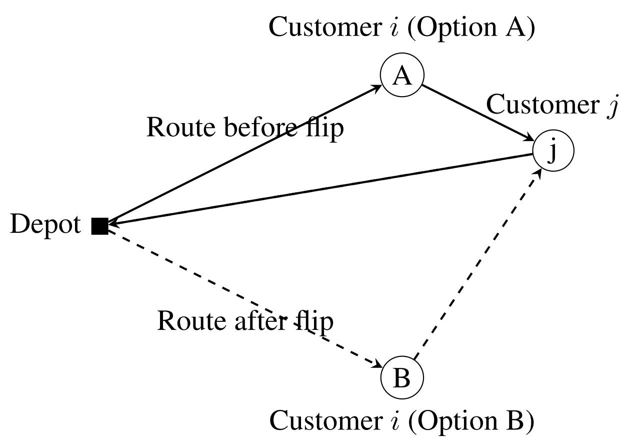

Fig. 1: Illustration of a *flip* move reassigning customer i from Option A to Option B, thereby modifying travel distances, time windows, and feasibility conditions.

4.3.4. Priority Swap operator

The priority-swap operator is a domain-specific diversification mechanism designed to explore alternative option configurations while maintaining the current service-level distribution. In contrast to the flip operator, which may increase or decrease service levels, this operator enforces a priority-preserving exchange between two customers. For each customer currently served by an option that has available alternatives, the operator identifies other customers with similarly flexible choices. A swap is considered only if the following condition holds: The priority level of the option removed for customer i must match the priority level of the option inserted for customer j, and the priority level of the option removed for customer j, must match the priority level of the option inserted for i.

This priority-matching constraint ensures that service-level feasibility is preserved. Once such a candidate pair is found, the corresponding move is evaluated and compared with the current best move. The operator therefore provides structured diversification without jeopardizing compliance with priority-based service requirements.

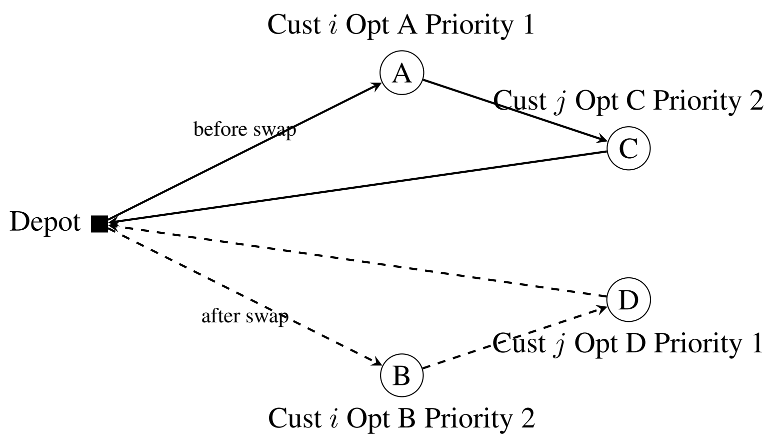

Fig. 2: Priority-swap move: Customer i switches from Option A (Priority 1) to B (Priority 2), while Customer j switches from C (Priority 2) to D (Priority 1). Service-level structure is preserved while route geometry changes.

4.3.5. Tabu policy - Promises

We employ a promise-based tabu mechanism (Zachariadis et al., 2015), previously shown effective in classical and profitable routing settings (Zachariadis et al., 2013; Manousakis et al., 2022). The memory is a matrix of option pairs (i, j) (consecutive options / arcs). When a candidate move (relocate, swap, two-opt, flip, or priority swap) is evaluated, it is discarded if it would recreate any adjacency (i, j) whose stored promise value $P_{i,j}$ is strictly lower than the candidate solution cost $Z(S^{cand})$; otherwise it is admissible. After an accepted move, each newly created adjacency (i, j) receives an updated promise $P_{i,j} \leftarrow Z(S^{new})$, committing the search not to reintroduce that edge unless it yields a strictly better cost. This cost-based filter suppresses short cycling while still allowing improving returns to earlier structural components, a property that is particularly important after option-changing moves. To prevent over-restriction as more promises accumulate, the entire matrix is reinitialized after $|O|\,\rho_r$ iterations, releasing all arcs and enabling renewed intensification. The approach is tenure-free, purely quality-driven, and imposes negligible overhead since only a constant number of adjacencies per move are queried and updated.

5. Computational Results

This section presents the computational experiments conducted to assess the effectiveness of the proposed local search framework. The discussion begins with an overview of the VRPDO benchmark instances used for performance evaluation. This is followed by the comparative algorithms used for benchmarking. The subsequent subsection contains experiments for parameter tuning and evaluating the role of the dedicated operators, as well as the utilization metric. Lastly, the results of the algorithm are discussed. The algorithm is implemented in C#. All computational experiments were carried out on a Windows 10 system equipped with an Intel Core i7-8700 @ 3.20 GHz processor and 16 GB RAM, utilizing a single thread.

5.1. Datasets

The computational evaluation of the proposed methodology employs two established benchmark sets to provide a comprehensive performance assessment.

The first benchmark set, developed by Tilk et al. (2021), was specifically generated for the Vehicle Routing Problem with Delivery Options (VRPDO). This set comprises instances organized into twelve distinct classes, each containing ten instances. These classes explore the problem space through controlled variation of three fundamental characteristics. The problem scale varies between instances that include 25 or 50 requests. Service flexibility differs between Class V instances, which average 1.5 delivery options per request, and Class U instances, which feature two options. Temporal constraint intensity encompasses time-window configurations categorized as small, medium, or large. Specifically, for individual delivery locations, the time-window widths are 60 minutes (small), 120 minutes (medium), and 240 minutes (large), while for shared delivery locations, the corresponding widths are 240 minutes (small), 480 minutes (medium) and 600 minutes (large). All instances implement a three-level priority hierarchy uniformly distributed across delivery options and operate within a standardized 12-hour (720 minutes) planning horizon.

Locations are randomly distributed within a 50×50 coordinate system, with travel costs and times derived from Euclidean distances scaled by a factor of 10 and rounded up. The framework incorporates realistic operational elements that enhance the model's practical applicability. Location-specific preparation times are set to 6 minutes for individual delivery locations and 4 minutes for shared delivery locations, reflecting the faster accessibility of shared points like shops or lockers. Differentiated service times are implemented such that options at shared delivery locations have a service time of 2 minutes, while those at individual delivery locations have a service time of 5 minutes, acknowledging the efficiency advantages of consolidated delivery points. Shared delivery locations are configured as one-fifth of the total locations ($|N|/5$), each with a capacity ranging from three to five requests. Vehicle parameters include a capacity set to 150, with demands drawn uniformly at random from the interval $[10, 20]$. A sufficient number of vehicles is assumed to be available at the depot. Fixed costs per vehicle are set to $100,000$, dominating travel costs to model a hierarchical objective of minimizing the number of vehicles first, followed by total travel cost. This benchmark set is publicly available at `https://logistik.bwl.uni-mainz.de/research/benchmarks`.

The second benchmark set, introduced by Dumez et al. (2021a), comprises 120 purpose-built instances designed to explore diverse operational scenarios in the absence of prior benchmark data for the VRPDO. Instances are organized into three main configuration classes —U, V, and UBC— each evaluated at four different problem sizes (50, 100, 200 and 400 customers), resulting in ten instances per size-class combination. These instances vary systematically across key structural features that capture different operational scenarios.

Spatial distribution follows a consistent pattern where all delivery locations are randomly placed within a 50×50 coordinate grid, with the depot fixed at the origin. Travel distances are computed using unit-cost Euclidean metrics. A consistent 12-hour operational timeframe (720 time units) serves as the planning horizon across all instances. Service flexibility and capacity constraints vary significantly across the three configuration classes. U instances feature customers with 1–3 delivery options (averaging 2), with vehicle capacities allowing service to approximately 10 customers per route. Lockers serve as shared delivery locations with capacities between 3 and 5 parcels. V instances maintain identical vehicle and locker capacity specifications to U instances, but incorporate reduced flexibility through 1–2 options per customer, averaging 1.5. UBC instances represent high-capacity scenarios, where vehicles accommodate approximately 25 customers per route. In these instances, locker availability is reduced fivefold, with increased capacity per unit to maintain operational feasibility. Time window related constraints are implemented through diverse window configurations that reflect realistic operational requirements. Individual delivery locations operate within structured windows such as morning periods spanning $[0, 360]$ and afternoon periods covering $[360, 720]$, or alternatively through randomly generated intervals. Shared locations operate either on full-day availability or random time windows to introduce diverse temporal constraints that challenge the optimization algorithm's ability to efficiently coordinate deliveries across different location types.

5.2. Comparative algorithms

To rigorously evaluate the performance of the proposed local search framework, three established state-of-the-art algorithms developed for the Vehicle Routing Problem with Delivery Options (VRPDO) were selected as benchmarks. The first is the Exact Branch-Price-and-Cut (BPC) method by Tilk et al. (2021), which serves as the reference exact method for the VRPDO and establishes the Best Known Solutions (BKS) for the first dataset. Its implementation utilizes C++ compiled with Microsoft Visual Studio 2015, relying on IBM ILOG CPLEX 12.9.0 for solving restricted master problems, and was executed on a Intel i7-5930K system.

The second benchmark is the Large Neighborhood Search with Set-Partitioning (LNS-SPP) matheuristic by Dumez et al. (2021a), designed to scale the solution process to larger instances and defining the BKS for the second dataset. As a third comparator, we include the Enhanced Matheuristic with Exact Neighborhoods (LNS-Balas) by Dumez et al. (2021b), an improved variant of the LNS-SPP that generated several new best solutions for the 100- and 200-customer groups. Both matheuristics are coded in C++ (g++ 5.4.0) and use IBM ILOG CPLEX 12.8.0 for embedded MIP models, with experiments conducted on single-threaded Intel Xeon X5650 systems under Ubuntu 16.04 LTS.

Together, these three algorithms provide a comprehensive and challenging baseline spanning both exact and heuristic approaches across all VRPDO benchmark sets.

5.3. Experiments

This section begins by presenting tuning experiments for the core parameters of the basic algorithm, focusing on the interaction between restarts and the promises-based tabu mechanism. We then analyze the behavior and effectiveness of the newly introduced diversification operators—flip and priority swap—and finally examine the contribution of the utilization metric in guiding search toward well-balanced and cost-effective solutions.

5.3.1. Parameter tuning

Promises act as dynamic quality thresholds in the tabu policy: when a customer option is removed, its best-seen cost is stored, and reinsertion is blocked unless a strictly better solution is found. This discourages cycling and helps explore new structures. Promise values are periodically reset to avoid excessive move forbidding. Restarts rebuild randomized initial solutions and trigger a new local search phase; this injects global diversification while avoiding reliance on a single starting point. The number of restarts is a key trade-off: too few risk premature convergence, while too many add time with diminishing returns, so it is jointly tuned with promise resets.

The following table reports the results of the tuning experiments, which examine the joint effect of the number of restarts and the promises reset parameter. The values indicate the relative gap to the baseline configuration, where more negative values correspond to larger improvements. The baseline in each case corresponds to the best published solution for the given instance. For these experiments, we evaluated our algorithm on a representative subset of 33 benchmark instances, consisting of small, medium, and large instances from the U and V families introduced by Tilk et al. (2021), together with U, V, and UBC instances of sizes 50 and 100 from Dumez et al. (2021a).

		Promises (ρ_r)			
		1	1.5	3	5
Restarts	5	-0.05	0.11	0.41	0.93
	10	-0.22	-0.18	0.00	0.29
	15	-0.44	-0.59	-0.13	-0.02

Table 1: Effect of restarts and promise reset on performance relative to baseline.

Table 1 shows the optimality gap from best known solutions of the literature. A negative value indicates that a lower cost solution was found. Towards fair comparison, only instances for which a matching number of used vehicles is found. With few restarts, aggressive promise resets consistently harm performance, indicating overly disruptive resets when global diversification is limited. With moderate restarts, lower reset scales (1, 1.5) yield modest gains, while larger scales remain neutral to mildly negative. The strongest performance occurs at 15 restarts, especially at 1.5, which achieves the largest average cost reduction. This setting balances intensification within each restart and structured diversification across them. Accordingly, we adopt 15 restarts and $\rho_r = 1.5$ as the default configuration.

5.3.2. *Role of Flip and Priority Swap operators*

This experiment evaluates the marginal contribution of the two diversification/intensification operators: (i) the flip move that toggles the activation status of an option and (ii) the priority swap that exchanges the served option of a customer with another candidate of different preference level. We executed two algorithm configurations on representative instance $U\_25small\_2$: (a) full configuration (Flip + Priority Swap Mechanism enabled) and (b) configuration without these components. All other parameters (initial solution construction, neighborhood sequence, stopping criteria) were kept identical. Importantly, both operators are active only until the first feasible solution (meeting vehicle and service constraints) is obtained; afterwards they are deactivated so that the search focuses purely on cost–reducing intra-feasible refinements.

Figure 3 depicts the evolution of the best-so-far solution cost for the two configurations. The solid curve (With Flip & PSM) corresponds to the full algorithm, whereas the dashed curve (Without Flip & PSM) represents the variant with these operators removed. Once a first feasible solution (satisfying all constraints) is reached, both operators are deactivated in the full configuration. Despite this, the full configuration secures at least one further improvement and terminates with lower final cost than the restricted variant. The red dots represent that the selected options have changed resulting in a different CVRPTW instance.

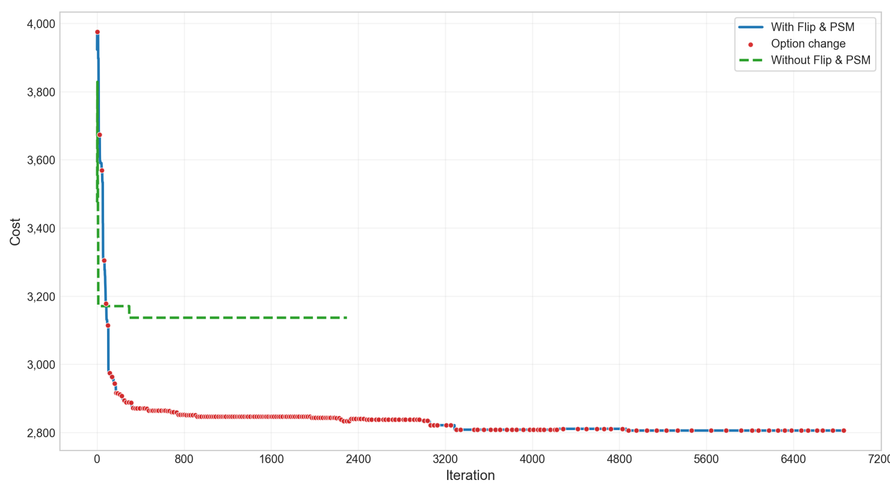

Fig. 3: Cost evolution on instance $U\_25small\_2$ comparing the full algorithm (solid) against the variant without Flip and Priority Swap (dashed).

The simple scheme uses the operators until a feasible solution is reached. Then only routing operators are used reaching in a plateau once the routing solutions reaches a local optimum. On the other hand, the full configuration after reaching feasibility, it continues finding additional improvements after feasibility is achieved, and terminates with a signifantly lower final cost than the restricted configuration. Both operators allow the search to examine different options combinations. Each combination yields a different CVRPTW setting, whereas at the same time the routing operators optimize the routing aspect. The synergy between the two distinct operator families ensure a balanced diversification and intensification throughout the search. Also, it is important to underline that the two operators are responsible for driving the search towards the feasible space. This enables the use of simple solution construction heuristics that relax constraints and higher quality initial solutions.

5.3.3. Role of route utilization metric

Empirically, the vast majority of candidate moves encountered after a few iterations are non-improving in pure routing cost terms: their raw move cost is positive because the search is already near a local minimum for the current fleet size and options. Accepting such moves would drift the solution without strategic purpose. The route utilization metric supplies that purpose by selectively softening the evaluation of consolidation moves that reorganize load so as to make a future route removal feasible.

When a tentative move redistributes demand in a way that increases the route utilization metric (i.e., some routes become more tightly loaded while another approaches emptiness, the ratio becomes smaller than 1. Scaling the (typically small) positive move cost by a factor below 1 boosts enables acceptance of a controlled sequence of such consolidation moves. These steps deliberately steer one route toward becoming redundant, after which it can be removed. This behaviour aligns with the optimization hierarchy: reducing the fleet precedes marginal travel cost refinement. After the number of routes reaches the computed lower bound, the mechanism is deliberately neutralized by fixing the ratio to 1 to avoid unnecessary diversions. From that point onward every move is assessed solely on routing cost and the search transitions to classical intra-fleet intensification. This two-phase behavior (guided consolidation followed by pure cost refinement) is precisely what the trajectories in Figure 4 illustrate.

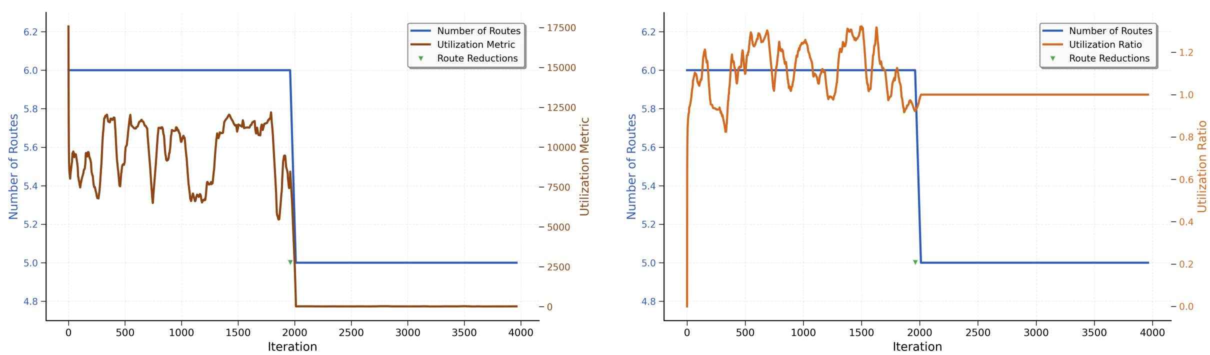

Fig. 4: Progression of route utilization metric and utilization ratio during local search.

The utilization ratio plot shows frequent periods where the ratio value is less than 1 (e.g., first iterations). These periods denote the guided consolidation phase, where the local search accepts moves that improve utilization by softening their perceived cost. The ratio then flattens to 1.0 once the minimum fleet size is achieved, indicating the clean transition to the second phase of pure routing cost minimization. In summary, the route utilization metric and its ratio-based scaling provide a disciplined way to accept a limited set of strategically useful, slightly worsening moves whose near-term cost trade-off is justified by the higher-level goal of reducing fleet size; once that goal is achieved the mechanism cleanly disengages.

5.4. Results and comparisons

This section compares the results of the proposed algorithm with the benchmark algorithms on the datasets as discussed in section 5.2. Results are aggregated at the class level, with each class containing ten instances.

Performance evaluation employs five primary metrics: *Cost* represents the mean objective value achieved by our algorithm; *BKS Cost* denotes the average best-known solution values from the literature under identical lexicographic ordering; *Routes* indicates the mean fleet size utilized; *Time (s)* averages the restart computation time in seconds. The second part of the table compares the results with the BKS. Column *Gap* quantifies relative solution quality as $100 \times (\text{Cost} - \text{BKS Cost})/\text{BKS Cost}$. Negative gap values therefore indicate improvements over existing best-known solutions, while positive values represent the optimality gap. For fair comparison, we only compare the instances for which our algorithm and the BKS employ the same number of vehicles. Finally, columns *#Matched* and *#New BKS* indicate the number of solutions matched to the BKS and the number of new best solutions reported.

Table 2: Summary Results for Tilk et al. (2021) instances

Group	**Cost**	**BKS Cost**	**Routes**	**Time (s)**	**Gap (%)**	**#Matched**	**#New BKS**
U_25small	2,458.6	2,456.9	3.0	42.6	0.11	9	0
U_25medium	2,185.4	2,186.3	3.0	43.7	-0.04	6	1
U_25large	2,436.6	2,441.0	3.0	56.8	-0.18	8	1
U_50small	3,839.1	3,895.8	5.6	139.3	-1.47	2	5
U_50medium	3,995.9	4,166.0	5.5	161.6	-4.07	1	8
U_50large	3,928.9	3,981.0	5.5	165.8	-1.28	0	8
V_25small	2,629.0	2,618.0	3.0	26.4	0.42	9	0
V_25medium	2,448.3	2,443.9	3.0	44.1	0.18	7	0
V_25large	2,116.5	2,115.0	3.0	32.0	0.07	9	0
V_50small	4,379.5	4,379.5	5.6	113.3	0.00	0	4
V_50medium	3,828.2	3,832.0	5.3	128.1	-0.10	3	4
V_50large	3,773.4	3,858.7	5.4	133.8	-2.22	0	7
Overall	3,168.3	3,189.1	4.2	1,415	-0.71	54	38

On the instances originally solved by the Exact BPC method (Tilk et al., 2021) (Table 2), our approach achieves an overall average gap of -0.71%, demonstrating a strong capacity for cost reduction despite the exact nature of the BKS baseline. The algorithm matches 54 existing BKS and finds 38 new BKS across the 120 instances underlining its capability to find optimal solutions despite its non-exact nature. The two dedicated operators usage is responsible for this behaviour.

Cost reductions are most pronounced in instances with relaxed time constraints, where the expanded feasibility space allows for superior route sequencing and option selection (e.g., U_50medium at -4.07% and V_50large at -2.22%). Conversely, small positive gaps (e.g., U_25small at 0.11%) are observed in the smallest and most tightly constrained configurations at which the exact baseline excels.

Table 3: Summary Results for Dumez et al. (2021a) instances

Group	Cost	BKS Cost	Routes	Time (s)	Gap (%)	#Matched	#New BKS
U_50	391.323	392.8	5.5	98.2	-1.47	0	2
U_100	664.404	649.5	10.5	397.6	2.32	0	0
U_200	1,215	1,178.6	20.6	1297.9	3.65	0	0
U_400	2,318	2,514.9	40.9	5077.8	-8.47	0	7
UBC_50	228.381	229.3	2.0	229.7	-0.42	3	4
UBC_100	358.837	359.4	4.0	929.5	-0.16	0	6
UBC_200	619.858	611.7	8.0	2,826.0	1.32	0	5
UBC_400	1,140	1,266.8	15.7	9,766.6	-9.69	0	9
V_50	378.678	374.2	5.4	93.4	1.19	2	2
V_100	702.1	700.6	10.6	318.5	-0.31	0	1
V_200	1,381	1,353.5	20.7	1,122.0	3.55	0	0
V_400	2,296	2,484.3	40.8	4,496.1	-1.66	0	5
Overall Average	974.648	1,009.6	15.4	2,221.1	-0.85	5	41

On the large-scale instances (Table 3), our approach manages an average gap from the BKS (as set by the LNS-SPP matheuristics (Dumez et al., 2021a) and (Dumez et al., 2021b)) of -0.85%. In total 5 solutions are matched and 41 new BKS are reported.

The most substantial improvements are achieved in the largest and most complex instance groups, highlighting the strength of the utilization-guided acceptance criteria: U_400 recorded a gap of -8.47%, and UBC_400 (featuring high-capacity vehicles) recorded -9.69%. These results confirm that the strategic consolidation moves are highly effective in exploiting the structural complexity of large VRPDO instances. Also, as the exact powered matheuristic of (Dumez et al., 2021b) pushes the envelope for 100 and 200 customer instances, making it harder for our algorithm to provide as substantial improvements.

6. Conclusions

This paper addresses the complex Vehicle Routing Problem with Delivery Options (VRPDO), a critical last-mile logistics challenge defined by a strict lexicographical objective (fleet size over cost). The model is realistic and reflects modern last mile delivery practices, incorporating multiple operational constraints. This renders the development of a solutions framework especially challenging. Our main contribution is a novel Local Search metaheuristic that utilizes two families of operators: option-oriented and routing-oriented, designed to promote diversification and intensification. Moreover, a route utilization metric is designed to modify move cost to drive consolidation throughout the search.

Empirical results validate the profound impact of the utilization metric, which promotes accepting non-improving moves that promote near empty and near full vehicles instead of balanced, leading to vehicle elimination. The algorithm delivered satisfactory solution quality, achieving average gaps of -0.71% on small/medium instances and -0.85% on large instances against BKS. In total, 79 new BKS across the combined datasets (38/120 and 41/120, respectively), confirming its strength in both fleet minimization and subsequent cost refinement.

Several future directions can be useful for the VRPDO model. Firstly, the algorithm is simple to tune metaheuristic and shows merit in competing against more complex exact and matheuristic algorithms. However, it is worth investigating the integration of exact aspect for subproblem solving or for as a set-partitioning component for targeting consolidation and route elimination. With respect to the VRPDO model, an extension to incorporate stochastic elements (e.g., uncertain demand or travel times) will further increase its practical relevance in dynamic last-mile environments.

Acknowledgements

The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number: 02562).

References

Angelelli, E., Archetti, C., Vindigni, M., 2014. The Clustered Orienteering Problem. *European Journal of Operational Research* 238, 2, 404–414.

Applegate, D., Bixby, R., Chvatal, V., Cook, W., 2006. Concorde tsp solver.

Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J., 2007. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, USA.

Archetti, C., Carrabs, F., Cerulli, R., 2018. The set orienteering problem. *European Journal of Operational Research* 267, 264–272.

Baldacci, R., Bartolini, E., Laporte, G., 2010. Some applications of the generalized vehicle routing problem. *Journal of the Operational Research Society* 61, 7, 1072–1077. https://doi.org/10.1057/jors.2009.51.

Bernardino, R., Paias, A., 2018. Solving the family traveling salesman problem. *European Journal of Operational Research* 267, 2, 453–466.

Boyer, R.S., Moore, J.S., 1977. A fast string searching algorithm. *Commun. ACM* 20, 10, 762–772.

Bräysy, O., Gendreau, M., 2005. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science* 39, 1, 104–118.

Cardeneo, A., 2005. Modellierung und Optimierung des B2C-Tourenplanungsproblems mit alternativen Lieferorten und -zeiten. Dissertation, Universität Karlsruhe (TH), Karlsruhe, Germany. (in German).

Carrabs, F., 2020. A biased random-key genetic algorithm for the set orienteering problem. *European Journal of Operational Research* 292, 3, 830–854.

Chao, I.M., Golden, B.L., Wasil, E.A., 1996. A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research* 88, 3, 475–489.

Chisman, J.A., 1975. The clustered traveling salesman problem. *Computers Operations Research* 2, 2, 115–119.

Drexl, M., 2012. Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science* 46, 3, 297–316.

Dumez, D., Lehuédé, F., Péton, O., 2021a. A large neighborhood search approach to the vehicle routing problem with delivery options. *Transportation Research Part B: Methodological* 144, 103–132.

Dumez, D., Tilk, C., Irnich, S., Lehuédé, F., Péton, O., 2021b. Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows. *EURO Journal on Transportation and Logistics* 10, 100040.

Eldrandaly, K., Ahmed, A., 2008. Routing problems: A survey.

Favaretto, D., Moretti, E., Pellegrini, P., 2007. Ant colony system for a vrp with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics* 10, 2, 263–284. https://doi.org/10.1080/09720502.2007.10700491.

Felch, V., Karl, D., Asdecker, B., Niedermaier, A., Sucky, E., 2019. Reconfiguration of the last mile: consumer acceptance of alternative delivery concepts. In *Logistics Management: Strategies and Instruments for digitalizing and decarbonizing supply chains-Proceedings of the German Academic Association for Business Research, Halle, 2019*, Springer, pp. 157–171.

Fischetti, M., Gonzalez, J.J.S., Toth, P., 1998. Solving the Orienteering Problem through branch-and-cut. *INFORMS Journal on Computing* 10, 2, 133–148.

Fischetti, M., Salazar González, J.J., Toth, P., 1997. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research* 45, 3, 378–394. https://doi.org/10.1287/opre.45.3.378.

Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2017. A matheuristic for the electric vehicle routing problem with capacitated charging stations. Ph.D. thesis, Centre interuniversitaire de recherche sur les reseaux d'entreprise, la ....

Grabenschweiger, J., Doerner, K.F., Hartl, R.F., Savelsbergh, M.W., 2021. The vehicle routing problem with heterogeneous locker boxes. *Central European Journal of Operations Research* 29, 113–142.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2021. The vehicle routing problem with cross-docking and resource constraints. *Journal of Heuristics* 27, 31–61.

Gunawan, A., Lau, H.C., Vansteenwegen, P., 2016. Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research* 255, 2, 315–332.

Helsgaun, K., 2000a. Effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research* 126, 1, 106–130.

Helsgaun, K., 2000b. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research* 126, 1, 106–130.

Helsgaun, K., 2015. Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun Algorithm. *Mathematical Programming Computation* 7, 3, 269–287.

Hempsch, C., Irnich, S., 2008. Vehicle routing problems with inter-tour resource constraints. *The vehicle routing problem: latest advances and new challenges* pp. 421–444.

Janinhoff, L., Klein, R., Scholz, D., 2024. Multitrip vehicle routing with delivery options: A data-driven application to the parcel industry. *OR Spectrum* 46, 2, 241–294.

Jongens, K., Volgenant, T., 1985. The symmetric clustered traveling salesman problem. *European Journal of Operational Research* 19, 1, 68–75.

Ke, L., Zhai, L., Li, J., Chan, F.T., 2016. Pareto mimic algorithm: An approach to the team orienteering problem. *Omega (United Kingdom)* 61, 155–166.

Keshtkaran, M., Ziarati, K., 2016. A novel grasp solution approach for the orienteering problem. *Journal of Heuristics* 22.

Keshtkaran, M., Ziarati, K., Bettinelli, A., Vigo, D., 2016. Enhanced exact solution methods for the Team Orienteering Problem. *International Journal of Production Research* 54, 2, 591–601.

Kılıç, K., Meterelliyoz, M., Güvenç Pelit, , Soysal, M., 2025. Modeling a humanitarian-aid covering tour problem with location selection and vehicle assignment decisions. *International Transactions in Operational Research*. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.70088, doi: https://doi.org/10.1111/itor.70088.

Laporte, G., Asef-Vaziri, A., Sriskandarajah, C., 1996. Some applications of the generalized travelling salesman problem. *The Journal of the Operational Research Society* 47.

Laporte, G., Nobert, Y., 1983. Generalized travelling salesman problem through n sets of nodes: An integer programming approach. *INFOR Journal* 21.

Last Mile Experts, 2024. Green Last Mile Europe 2024 Report. Technical report, Last Mile Experts.

Lin, S., Kernighan, B., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Oper. Res.* 21, 498–516.

Lokin, F., 1979. Procedures for travelling salesman problems with additional constraints. *European Journal of Operational Research* 3, 2, 135–141.

Lombard, A., Tamayo-Giraldo, S., Fontane, F., 2018. Vehicle routing problem with roaming delivery locations and stochastic travel times (vrprdl-s). *Transportation research procedia* 30, 167–177.

Los, J., Spaan, M.T., Negenborn, R.R., 2018. Fleet management for pickup and delivery problems with multiple locations and preferences. In *Dynamics in Logistics: Proceedings of the 6th International Conference LDIC 2018, Bremen, Germany*, Springer, pp. 86–94.

Mancini, S., Gansterer, M., 2021. Vehicle routing with private and shared delivery locations. *Computers & Operations Research* 133, 105361.

Manousakis, E., Repoussis, P., Zachariadis, E., Tarantilis, C., 2021. Improved branch-and-cut for the Inventory Routing Problem based on a two-commodity flow formulation. *European Journal of Operational Research* 290, 3, 870–885.

Manousakis, E.G., Kasapidis, G.A., Kiranoudis, C.T., Zachariadis, E.E., 2022. An infeasible space exploring matheuristic for the Production Routing Problem. *European Journal of Operational Research* 298, 2, 478–495.

Moccia, L., Cordeau, J.F., Laporte, G., 2012. An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. *Journal of the Operational Research Society* 63, 2, 232–244.

Morganti, E., Dablanc, L., Fortin, F., 2014. Final deliveries for online shopping: The deployment of pickup point networks in urban and suburban areas. *Research in Transportation Business & Management* 11, 23–31.

Morán-Mirabal, L., González-Velarde, J., Resende, M., 2014. Randomized heuristics for the family traveling salesperson problem. *International Transactions in Operational Research* 21, 1, 41–57. https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.12026.

Nguyê n, T.B.T., Bektaş, T., Cherrett, T.J., McLeod, F.N., Allen, J., Bates, O., Piotrowska, M., Piecyk, M., Friday, A., Wise, S., 2019. Optimising parcel deliveries in london using dual-mode routing. *Journal of the Operational Research Society* 70, 6, 998–1010.

Ozbaygin, G., Karasan, O.E., Savelsbergh, M., Yaman, H., 2017. A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological* 100, 115–137.

Pěnička, R., Faigl, J., Saska, M., 2019. Variable Neighborhood Search for the Set Orienteering Problem and its application to other Orienteering Problem variants. *European Journal of Operational Research* 276, 3, 816–825.

Pop, P., Matei, O., Pintea, C., 2018. A two-level diploid genetic based algorithm for solving the family traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, New York, NY, USA, p. 340–346.

Rai, H.B., Cetinkaya, A., Verlinde, S., Macharis, C., 2020. How are consumers using collection points? evidence from brussels. *Transportation Research Procedia* 46, 53–60.

Raucq, J., Sörensen, K., Cattrysse, D., 2019. Solving a real-life roll-on–roll-off waste collection problem with column generation. *Journal on Vehicle Routing Algorithms* 2, 41–54.

Reinelt, G., 1995. Tsplib 95. *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg* 338, 1–16.

Repoussis, P.P., Tarantilis, C.D., 2010. Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming. *Transportation Research Part C: Emerging Technologies* 18, 5, 695–712.

Reyes, D., Savelsbergh, M., Toriello, A., 2017. Vehicle routing with roaming delivery locations. *Transportation Research Part C: Emerging Technologies* 80, 71–91.

Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation science* 50, 2, 579–590.

Savelsbergh, M.W., 1985. Local search in routing problems with time windows. *Annals of Operations research* 4, 285–305.

Shi, X., Liang, Y., Lee, H., Lu, C., Wang, Q., 2007. Particle swarm optimization-based algorithms for tsp and generalized tsp. *Information Processing Letters* 103, 5, 169–176.

Sitek, P., Wikarek, J., 2019. Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research* 273, 1, 257–277.

Smith, S.L., Imeson, F., 2017. Glns: An effective large neighborhood search heuristic for the generalized traveling salesman problem. *Computers and Operations Research* 87, 1–19.

Snyder, L.V., Daskin, M.S., 2006. A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research* 174.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., Van Oudheusden, D., 2013. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science* 47, 1, 53–63.

Statista, 2024. Retail e-commerce sales worldwide from 2022 to 2028.

Taillard, É.D., Gambardella, L.M., Gendreau, M., Potvin, J.Y., 2001. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research* 135, 1, 1–16.

Taillard, É.D., Helsgaun, K., 2019. POPMUSIC for the travelling salesman problem. *European Journal of Operational Research* 272, 2, 420–429.

Tarantilis, C.D., 2005. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers and Operations Research* 32, 9, 2309–2327.

Tilk, C., Olkis, K., Irnich, S., 2021. The last-mile vehicle routing problem with delivery options. *Or Spectrum* 43, 4, 877–904.

Tsai, J.F., Ngo, H.N., Che, Z.H., 2024. Last-mile delivery during covid-19: A systematic review of parcel locker adoption and consumer experience. *Acta Psychologica* 249, 104462.

Tsiligirides, T., 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society* 35, 9, 797–809.

Vakulenko, Y., Hellström, D., Hjort, K., 2018. What's in the parcel locker? exploring customer value in e-commerce last mile delivery. *Journal of Business Research* 88, 421–427.

Vansteenwegen, P., Souffriau, W., Oudheusden, D.V., 2011a. The orienteering problem: A survey. *European Journal of Operational Research* 209, 1, 1–10.

Vansteenwegen, P., Souffriau, W., Oudheusden, D.V., 2011b. The orienteering problem: A survey. *European Journal of Operational Research* 209, 1, 1–10.

Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231, 1, 1–21.

Yuen, K.F., Wang, X., Ma, F., Wong, Y.D., 2019. The determinants of customers' intention to use smart lockers for last-mile deliveries. *Journal of Retailing and Consumer Services* 49, 316–326.

Zachariadis, E.E., Kiranoudis, C.T., 2011. Local search for the undirected capacitated arc routing problem with profits. *European Journal of Operational Research* 210, 2, 358–367.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2013. Integrated distribution and loading planning via a compact metaheuristic algorithm. *European Journal of Operational Research* 228, 1, 56–71.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2015. The load-dependent vehicle routing problem and its pick-up and delivery extension. *Transportation Research Part B: Methodological* 71, 158–181.

Zhang, S., Lee, C., 2016. Flexible vehicle scheduling for urban last mile logistics: The emerging technology of shared reception box. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE, pp. 1913–1917.

Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research* 265, 2, 765–778.

Appendix A: Detailed results

The appendix reports detailed results for the first benchmark dataset corresponding to the instances introduced by Tilk et al. (2021). For each instance, the table lists the reference values from the literature (*Cost*, #Veh) and the outcomes obtained by our algorithm (*Cost*, #Veh, *Time*). Reported computation times correspond to the time of restart (in seconds).

The column *Gap (%)* measures the relative deviation of our cost from the reference value, computed as

$$100 \times \frac{\mathrm{Cost}_{\mathrm{ours}} - \mathrm{Cost}_{\mathrm{BKS}}}{\mathrm{Cost}_{\mathrm{BKS}}}.$$

Negative values therefore indicate improvements over the best-known solutions, whereas positive values represent an optimality gap. To ensure a fair comparison, the gap is reported only for instances where our method matches or decreases the used number of vehicles.

Table A1: First Dataset Results

Instance	Tilk et al. (2021)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Time	
U_25small_1	2188	3	2188	3	43.9	0
U_25small_2	2805	3	2805	3	40.0	0
U_25small_3	2460	3	2460	3	40.0	0
U_25small_4	2994	3	2994	3	35.9	0
U_25small_5	2580	3	2580	3	44.8	0
U_25small_6	1689	3	1689	3	45.9	0
U_25small_7	2454	3	2482	3	42.6	1.14
U_25small_8	1861	3	1861	3	47.1	0
U_25small_9	2992	3	2992	3	42.0	0
U_25small_10	2535	3	2535	3	43.7	0
U_25medium_1	2400	3	2400	3	46.8	0
U_25medium_2	2290	3	2290	3	53.6	0
U_25medium_3	1525	3	1525	3	43.8	0
U_25medium_4	2805	4	**2578**	**3**	42.8	-8.09
U_25medium_5	2074	3	2074	3	45.2	0
U_25medium_6	2566	3	2594	3	44.4	1.09
U_25medium_7	1566	3	1587	3	23.1	1.34
U_25medium_8	2065	3	2174	3	43.9	5.28
U_25medium_9	1916	3	1916	3	45.7	0
U_25medium_10	2716	3	2716	3	47.1	0
U_25large_1	2423	3	2423	3	54.2	0
U_25large_2	2797	3	2801	3	47.5	0.14
U_25large_3	2107	3	2107	3	61.1	0
U_25large_4	2411	3	2411	3	62.5	0
U_25large_5	1995	3	1995	3	59.0	0
U_25large_6	2316	3	**2272**	**3**	53.0	-1.9

Table A1 – continued

Instance	Tilk et al. (2021)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Time	
U_25large_7	2237	3	2237	3	58.1	0
U_25large_8	2586	3	2586	3	58.4	0
U_25large_9	2996	3	2996	3	56.6	0
U_25large_10	2538	3	2538	3	58.1	0
U_50small_1	—	—	**4627**	**5**	130.9	—
U_50small_2	5303	7	**4636**	**6**	119.3	-12.58
U_50small_3	3940	5	**3763**	**5**	138.5	-4.49
U_50small_4	3742	6	**3732**	**6**	135.3	-0.27
U_50small_5	—	—	**3847**	**5**	146.7	—
U_50small_6	3188	6	3250	6	146.9	1.94
U_50small_7	3066	6	3066	6	143.0	0
U_50small_8	4488	6	4526	6	146.2	0.85
U_50small_9	3460	5	3557	5	137.3	2.8
U_50small_10	3387	6	3387	6	148.8	0
U_50medium_1	4706	6	**4410**	**6**	177.7	-6.29
U_50medium_2	3432	6	**3397**	**6**	159.2	-1.02
U_50medium_3	4296	6	4296	6	155.9	0
U_50medium_4	3662	6	**3426**	**6**	159.1	-6.44
U_50medium_5	3387	5	3391	5	174.0	0.12
U_50medium_6	6136	6	**5528**	**6**	158.7	-9.91
U_50medium_7	4652	5	**4577**	**5**	163.0	-1.61
U_50medium_8	3979	5	**3581**	**5**	149.2	-10
U_50medium_9	—	—	**3085**	**5**	157.7	—
U_50medium_10	4332	6	**4268**	**5**	161.9	-1.48
U_50large_1	—	—	**4409**	**6**	168.2	—
U_50large_2	—	—	**4238**	**6**	185.0	—
U_50large_3	—	—	**3418**	**5**	169.2	—
U_50large_4	—	—	**4318**	**6**	184.3	—
U_50large_5	3678	6	**3555**	**6**	182.9	-3.34
U_50large_6	—	—	**4100**	**5**	201.0	—
U_50large_7	3394	5	**3336**	**5**	93.5	-1.71
U_50large_8	3917	5	3938	5	179.9	0.54
U_50large_9	3803	5	**3725**	**5**	112.2	-2.05
U_50large_10	4244	6	4252	6	182.0	0.19
V_25small_1	2876	3	2876	3	23.2	0
V_25small_2	2638	3	2638	3	29.7	0
V_25small_3	2716	3	2716	3	24.9	0
V_25small_4	2781	3	2781	3	26.2	0
V_25small_5	2401	3	2401	3	36.3	0
V_25small_6	2723	3	2723	3	22.2	0
V_25small_7	1941	3	1941	3	25.9	0
V_25small_8	2366	3	2366	3	27.5	0
V_25small_9	2817	3	2817	3	23.5	0
V_25small_10	2910	3	3031	3	24.3	4.16
V_25medium_1	2320	3	2339	3	45.6	0.82
V_25medium_2	2177	3	2177	3	44.7	0

Table A1 – continued

Instance	Tilk et al. (2021) Cost	Tilk et al. (2021) #Veh	This paper Cost	This paper #Veh	This paper Time	Gap (%)
V_25medium_3	2299	3	2299	3	46.6	0
V_25medium_4	2904	3	2927	3	38.2	0.79
V_25medium_5	2797	3	2797	3	48.2	0
V_25medium_6	2501	3	2501	3	46.2	0
V_25medium_7	2474	3	2474	3	44.1	0
V_25medium_8	2423	3	2423	3	43.1	0
V_25medium_9	2099	3	2104	3	45.5	0.24
V_25medium_10	2442	3	2442	3	38.4	0
V_25large_1	1990	3	1990	3	23.3	0
V_25large_2	2342	3	2342	3	51.1	0
V_25large_3	1627	3	1627	3	24.1	0
V_25large_4	2444	3	2462	3	60.7	0.74
V_25large_5	2064	3	2064	3	25.0	0
V_25large_6	2129	3	2129	3	35.5	0
V_25large_7	1533	3	1533	3	24.8	0
V_25large_8	1930	3	1930	3	22.6	0
V_25large_9	2484	3	2484	3	24.8	0
V_25large_10	2604	3	2604	3	27.9	0
V_50small_1	5111	5	**5084**	**5**	120.5	-0.53
V_50small_2	5117	7	**4691**	**6**	127.7	-8.33
V_50small_3	3934	6	3938	6	119.7	0.1
V_50small_4	3387	5	3749	5	133.5	10.69
V_50small_5	3353	5	**3234**	**5**	125.9	-3.55
V_50small_6	4343	6	4397	6	134.9	1.24
V_50small_7	4726	5	**4661**	**5**	119.4	-1.38
V_50small_8	5338	6	5399	6	63.8	1.14
V_50small_9	4082	6	4085	6	66.7	0.07
V_50small_10	4531	6	4557	6	121.0	0.57
V_50medium_1	—	—	**4172**	**5**	141.3	—
V_50medium_2	3576	5	3576	5	153.3	0
V_50medium_3	3526	5	3584	5	69.6	1.64
V_50medium_4	4275	6	4275	6	82.1	0
V_50medium_5	3945	5	**3937**	**5**	153.3	-0.2
V_50medium_6	—	—	**4376**	**5**	134.6	—
V_50medium_7	3078	6	3086	6	143.2	0.26
V_50medium_8	3574	5	3602	5	132.2	0.78
V_50medium_9	3709	6	3709	6	137.0	0
V_50medium_10	4098	5	**3965**	**5**	134.7	-3.25
V_50large_1	—	—	**4278**	**5**	106.8	—
V_50large_2	3634	7	**3473**	**6**	179.5	-4.43
V_50large_3	4021	5	4034	5	168.4	0.32
V_50large_4	3882	5	**3558**	**5**	162.0	-8.35
V_50large_5	—	—	**3428**	**5**	159.2	—
V_50large_6	—	—	**3899**	**5**	81.0	—
V_50large_7	3766	5	**3734**	**5**	153.6	-0.85
V_50large_8	3893	6	3903	6	97.2	0.26

Table A1 – continued

Instance	Tilk et al. (2021)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Time	
V_50large_9	3777	6	**3678**	**6**	123.0	-2.62
V_50large_10	3745	6	3749	6	106.9	0.11

Table A2: Second Dataset Results

Instance	Dumez et al. (2021a)		Dumez et al. (2021b)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Cost	#Veh	Time	
U_50_1	433.081	6	–	–	438.224	6	98.07	1.19
U_50_2	423.228	6	–	–	**422.111**	**6**	96.55	-0.26
U_50_3	329.138	6	–	–	**333.563**	**5**	101.85	–
U_50_4	427.183	5	–	–	429.772	6	97.59	–
U_50_5	353.179	6	–	–	357.665	6	98.59	1.27
U_50_6	399.620	5	–	–	405.904	5	106.08	1.57
U_50_7	332.481	5	–	–	336.601	5	103.76	1.24
U_50_8	384.635	5	–	–	390.274	5	94.47	1.47
U_50_9	359.888	5	–	–	376.910	5	93.89	4.73
U_50_10	422.048	6	–	–	422.206	6	91.85	0.04
U_100_1	479.972	11	479.972	11	486.244	11	438.79	1.31
U_100_2	683.365	10	691.132	10	700.400	10	394.11	2.49
U_100_3	630.215	11	630.020	11	649.560	11	382.48	3.1
U_100_4	592.860	10	609.721	10	624.075	10	383.48	5.27
U_100_5	673.844	10	673.312	10	701.850	10	399.10	4.24
U_100_6	586.810	10	598.608	10	604.311	10	431.06	2.98
U_100_7	746.400	11	763.644	11	753.039	11	363.64	0.89
U_100_8	847.533	11	845.885	11	854.831	11	386.84	1.06
U_100_9	687.659	10	682.765	10	691.918	10	409.42	1.34
U_100_10	574.328	11	573.197	11	577.813	11	387.86	0.81
U_200_1	1687.420	21	1491.370	21	1546.526	21	1082.18	3.7
U_200_2	1287.250	21	1189.360	21	1234.093	21	1332.00	3.76
U_200_3	1565.130	20	1289.470	20	1334.751	20	1370.70	3.51
U_200_4	943.988	21	920.695	21	957.088	21	1339.11	3.95
U_200_5	1112.500	21	1050.600	21	1096.419	21	1359.79	4.36
U_200_6	1261.360	20	1087.790	20	1127.671	20	1342.80	3.67
U_200_7	1108.010	21	1007.830	21	1031.583	21	1273.15	2.36
U_200_8	1259.660	20	1079.740	20	1170.952	20	1282.30	8.45

Table A2 – continued

Instance	Dumez et al. (2021a)		Dumez et al. (2021b)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Cost	#Veh	Time	
U_200_9	1575.460	20	1191.320	20	1160.984	21	1288.19	–
U_200_10	1840.350	20	1478.390	20	1495.917	20	1309.33	1.19
U_400_1	2014.570	41	–	–	**1962.364**	**41**	4635.83	-2.59
U_400_2	3614.200	41	–	–	**3183.072**	**41**	4793.11	-11.93
U_400_3	2641.320	41	–	–	**2301.902**	**41**	5896.37	-12.85
U_400_4	1971.310	40	–	–	1838.725	41	4710.84	–
U_400_5	2437.530	40	–	–	**2286.915**	**40**	5171.10	-6.18
U_400_6	2145.120	40	–	–	**2082.359**	**40**	4942.60	-2.93
U_400_7	2909.060	41	–	–	2572.926	42	4655.46	–
U_400_8	2032.460	41	–	–	**2029.188**	**41**	5945.03	-0.16
U_400_9	3058.030	40	–	–	2660.119	41	5284.55	–
U_400_10	2325.750	41	–	–	**2266.808**	**41**	4743.59	-2.53
UBC_50_1	256.042	2	–	–	256.042	2	213.94	0
UBC_50_2	301.963	2	–	–	**295.588**	**2**	199.41	-2.11
UBC_50_3	241.696	2	–	–	**241.377**	**2**	263.20	-0.13
UBC_50_4	223.522	2	–	–	**221.100**	**2**	202.77	-1.08
UBC_50_5	208.096	2	–	–	208.172	2	264.78	0.04
UBC_50_6	241.846	2	–	–	**233.149**	**2**	230.33	-3.6
UBC_50_7	246.186	2	–	–	252.342	2	243.44	2.5
UBC_50_8	196.079	2	–	–	196.079	2	210.14	0
UBC_50_9	183.804	2	–	–	183.804	2	224.69	0
UBC_50_10	194.297	2	–	–	196.156	2	245.22	0.96
UBC_100_1	374.862	4	368.237	4	370.196	4	868.23	0.53
UBC_100_2	351.220	4	350.970	4	**350.733**	**4**	845.32	-0.07
UBC_100_3	323.464	4	323.464	4	**323.215**	**4**	977.59	-0.08
UBC_100_4	334.615	4	334.615	4	337.279	4	864.60	0.8
UBC_100_5	371.859	4	371.549	4	372.125	4	1007.07	0.16
UBC_100_6	357.611	4	363.770	4	**351.658**	**4**	930.54	-1.66
UBC_100_7	337.902	4	337.902	4	339.255	4	1007.45	0.4
UBC_100_8	419.714	4	415.075	4	**411.526**	**4**	899.73	-0.86
UBC_100_9	386.871	4	384.456	4	**382.547**	**4**	923.49	-0.5
UBC_100_10	350.272	4	362.556	4	**349.839**	**4**	971.60	-0.12
UBC_200_1	601.353	8	590.562	8	**581.809**	**8**	2788.57	-3.25
UBC_200_2	511.301	8	493.489	8	514.043	8	2578.46	4.17
UBC_200_3	842.696	8	802.842	8	**787.418**	**8**	2812.91	-1.92
UBC_200_4	634.515	8	647.167	8	648.169	8	2641.54	2.15
UBC_200_5	669.268	8	633.661	8	**625.097**	**8**	2754.29	-1.35
UBC_200_6	649.461	8	627.195	8	631.630	8	2717.80	0.71
UBC_200_7	630.953	8	607.539	8	**607.325**	**8**	3320.64	-0.04
UBC_200_8	640.022	8	642.490	8	**628.265**	**8**	3104.96	-1.84
UBC_200_9	562.377	8	525.656	8	554.968	8	2727.21	5.58
UBC_200_10	562.377	8	661.724	8	619.852	8	2813.73	10.22
UBC_400_1	1019.140	16	–	–	**1004.391**	**16**	9049.03	-1.45
UBC_400_2	1247.220	16	–	–	**1162.810**	**16**	8545.62	-6.77
UBC_400_3	1433.390	15	–	–	1103.797	16	10250.74	–
UBC_400_4	1243.220	16	–	–	**1158.374**	**16**	9465.74	-6.82

Table A2 – continued

Instance	Dumez et al. (2021a)		Dumez et al. (2021b)		This paper			Gap (%)
	Cost	#Veh	Cost	#Veh	Cost	#Veh	Time	
UBC_400_5	1045.200	16	–	–	**1044.968**	**16**	11902.64	-0.02
UBC_400_6	1534.190	15	–	–	**1328.510**	**15**	9866.35	-13.41
UBC_400_7	1111.400	15	–	–	**1023.571**	**15**	10151.98	-7.9
UBC_400_8	1270.820	16	–	–	**1197.563**	**16**	8136.01	-5.76
UBC_400_9	1231.080	16	–	–	**1150.186**	**16**	8793.64	-6.57
UBC_400_10	1533.240	15	–	–	**1226.997**	**15**	11504.39	-19.97
V_50_1	416.349	5	–	–	428.474	5	87.20	2.91
V_50_2	345.635	6	–	–	345.635	6	102.43	0
V_50_3	398.662	5	–	–	402.893	5	98.96	1.06
V_50_4	374.501	5	–	–	394.770	5	90.63	5.41
V_50_5	328.387	5	–	–	329.887	5	81.06	0.46
V_50_6	387.196	5	–	–	**383.565**	**5**	98.13	-0.94
V_50_7	365.985	5	–	–	375.970	5	92.03	2.73
V_50_8	387.645	6	–	–	387.777	6	99.15	0.03
V_50_9	366.000	6	–	–	366.000	6	91.09	0
V_50_10	372.228	6	–	–	**371.814**	**6**	94.21	-0.11
V_100_1	647.566	11	647.566	11	652.306	11	325.50	0.73
V_100_2	895.205	10	848.567	10	**831.152**	**10**	304.20	-2.05
V_100_3	733.412	10	706.331	10	655.541	11	318.83	–
V_100_4	594.605	10	594.605	10	607.076	10	341.13	2.1
V_100_5	765.491	10	802.718	10	820.791	10	320.54	7.22
V_100_6	597.288	11	597.288	11	599.459	11	304.31	0.36
V_100_7	702.181	11	703.341	11	705.060	11	308.04	0.41
V_100_8	744.855	11	745.349	11	758.518	11	325.05	1.83
V_100_9	794.041	10	800.959	10	778.346	11	317.74	–
V_100_10	614.412	10	605.617	10	613.352	10	319.85	1.28
V_200_1	1288.280	21	1293.090	21	1338.739	21	1265.77	3.92
V_200_2	1420.190	20	1203.660	20	1193.027	21	1208.95	–
V_200_3	1346.210	21	1243.620	21	1282.745	21	1095.99	3.15
V_200_4	1488.720	21	1329.730	21	1402.938	21	1064.70	5.51
V_200_5	1420.810	21	1369.240	20	1343.927	21	758.47	–
V_200_6	1519.580	21	1445.760	21	1451.018	21	1138.94	0.36
V_200_7	1171.590	20	1163.270	20	1192.483	20	1293.91	2.51
V_200_8	1799.830	20	1486.380	20	1462.550	21	1204.32	–
V_200_9	1798.400	20	1623.930	20	1677.210	20	1096.94	3.28
V_200_10	1527.620	20	1381.510	20	1465.981	20	1092.04	6.11
V_400_1	2763.730	41	–	–	**2572.908**	**41**	4507.92	-6.9
V_400_2	3065.630	41	–	–	**2776.938**	**41**	4465.85	-9.42
V_400_3	2118.930	40	–	–	1870.058	41	4401.48	–
V_400_4	2072.650	40	–	–	1938.820	41	4549.74	–
V_400_5	1759.510	41	–	–	1826.650	41	4571.18	3.82
V_400_6	3281.810	40	–	–	**2872.594**	**40**	4416.18	-12.47
V_400_7	2391.090	41	–	–	2057.619	42	4534.94	–
V_400_8	1963.520	41	–	–	1990.453	41	4575.13	1.37
V_400_9	2448.350	40	–	–	**2341.601**	**40**	4434.64	-4.36
V_400_10	2978.620	40	–	–	**2721.569**	**40**	4503.99	-8.63