CIRRELT

CIRRELT-2024-24

The Flexible Park-and-Loop Routing Problem

Panca Jodiawan
Jean-François Côté
Leandro C. Coelho

July 2024

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2024-002

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

The Flexible Park-and-Loop Routing Problem

Panca Jodiawan[1], Jean-François Côté[1], Leandro C. Coelho[1,2,*]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, Université Laval, Québec, Canada
2. Canada Research Chair in Integrated Logistics, Université Laval, Québec, Canada

Abstract. This work investigates a variant of the vehicle routing problem inspired from last-mile delivery operations, incorporating explicit parking considerations and various types of flexibility. The first type of flexibility stems from the availability of parking spots, i.e., each driver may select from one of the available time windows to park at a parking spot. Time-based and space-based flexibility from customers are simultaneously exploited to reduce overall operational costs. To solve the problem, we propose a Mixed-Integer Linear Programming (MILP) model and devise a Hybrid Large Neighborhood Search (HLNS) algorithm, which embeds problem-specific destroy and repair operators, as well as a tailored dynamic programming to improve the configuration of a route. Within the HLNS, a set partitioning model is solved periodically as an attempt to find a combination of routes of better quality by utilizing the pool of routes collected so far. The effectiveness of HLNS is demonstrated on a set of newly generated instances and two special cases existing in the literature. Additionally, we demonstrate the impact of parking-related flexibility toward the solutions and show the difference in impacts resulting from considering each customer-related flexibility.

Keywords: Last-mile delivery, park-and-loop, hybrid metaheuristics

Acknowledgements. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) [grants 2021-04037 and 2019-00094]. This support is greatly acknowledged. We thank the Digital Research Alliance of Canada for providing high-performance parallel computing facilities.

---

* Corresponding author: leandro.coelho@fsa.ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2024

1 Introduction

In recent years, the e-commerce industry has experienced continuous growth, e.g., B2C in Canada generated a total revenue of more than US$52 billion in 2023 and is projected to increase to US$81 billion by 2028 [Statista, 2023]. The proportion of the global population who made online purchases increases over time, leading to a higher volume of orders needed to be fulfilled. Moreover, technological advancements and competition among e-commerce providers drive customer expectations toward shorter delivery times, e.g., next- or same-day deliveries [Boysen et al., 2021, Joerss et al., 2016]. These phenomena significantly impact last-mile deliveries, resulting in an escalated complexity of managing the operations.

Leveraging customer flexibility is a promising alternative to enhance the performance of existing last-mile delivery operations. Reyes et al. [2017] showed that car trunk delivery, in addition to home delivery, leads to a reduction in the total mileage traveled by delivery vehicles. Parcel lockers are also considered as alternative delivery locations, distinguished by two main differences, i.e., customers are required to pick up their packages at designated lockers in exchange for compensation, and deliveries can be performed at any time within the planning period [Mancini and Gansterer, 2021]. In addition to the flexibility defined by alternative delivery locations, flexibility in terms of delivery time is also favorable to reduce operational costs. Taş et al. [2014] investigated a delivery system that allows early or late deliveries. By adopting flexible time windows, thereby extending time windows originally imposed by customers, there is an increased likelihood of reducing total distances traveled by delivery vehicles, leading to cost and environmental savings.

Last-mile delivery operations in dense cities commonly face complex challenges, e.g., the considerable time to find parking spots [Reed et al., 2024]. In the operational context, effective and efficient last-mile delivery plans involve solving various vehicle routing problems (VRPs). However, most of them overlook parking decisions and implicitly assume that parking times at customers' locations are negligible. Recent works [Martinez-Sykora et al., 2020, Cabrera et al., 2023, Le Colleter et al., 2023] have demonstrated the potential benefits of integrating parking decisions into classical VRPs, including cost reductions and time savings. Despite this advancement, research in this domain is still in its early stages, with assumptions limiting its application, e.g., parking spots are always available. By introducing multiple time windows for each parking spot, our work extends previous literature to further represent real-world conditions of parking spot availability.

We then introduce the *Flexible Park-and-Loop Routing Problem* (FPLRP) that incorporates two sources of flexibility: customer-related and parking-related flexibility. Three types of customer-related flexibility are considered: (1) each customer can define a set of *roaming delivery locations*, each with a *preferable time window*; (2) a set of *lockers* is available for selection by each customer; and (3) each customer can also define an *acceptable time window* at a roaming delivery location, which is wider than the preferable time window. A customer receives compensation when their package is delivered to a locker or to a roaming location earlier or later than its preferable time window but still within its acceptable time window. When delivering a package to a roaming location, a driver has two options: directly driving to the location or parking the vehicle at an available parking spot and walking to the location. Parking-related flexibility is reflected in multiple time windows from which the driver selects one. Our contributions are summarized as follows.

- We introduce a new variant of the Park-and-Loop Routing Problem, which incorporates customer- and parking-related flexibility, aiming to more accurately capture real-world operations and exploiting various types of flexibility to reduce the overall cost.
- We formulate and solve the problem exactly on small instances; we also devise a Hybrid Large Neighborhood Search (HLNS) which relies on problem-specific destroy and repair operators, a tailored dynamic programming heuristic to find efficient configurations of parking and driving decisions, and a set partitioning model to generate a high-quality solution by utilizing the historical records of solutions.
- We evaluate the effectiveness of the proposed method by solving two special cases found in the literature and newly generated instances, improving many solutions from the literature.
- We present sensitivity analyses serving as managerial insights for decision makers by studying the impact of customer- and parking-related flexibility toward the solution characteristics.

The paper is structured as follows. Section 2 describes several related works. Section 3 formally defines FPLRP and presents the MILP formulation. Section 4 explains the development of our HLNS. Section 5 shows the computational results to verify the effectiveness of HLNS and presents sensitivity analyses to several characteristics of the FPLRP. Finally, Section 6 concludes the work and presents the possible future directions.

2 Literature Review

This section briefly reviews related works in the literature, aiming to bridge the gap between existing research and the problem investigated in our work. Section 2.1 describes variants of the VRP where certain types of flexibility are considered, and Section 2.2 discusses recent advances in incorporating parking decisions into various routing-related problems.

2.1 Flexibility in Routing Problems

Two types of customer-related flexibility are commonly integrated into routing problems, offering additional benefits to existing delivery systems. The first type of flexibility is defined in time windows specified by customers. Taş et al. [2014] extended the VRP with Time Windows (VRPTW) by allowing a delivery to occur earlier or later than its preferable time window within a certain limit. Customers whose packages are delivered outside their preferable time windows receive compensation, which is proportional to the total time deviation. They imposed a higher compensation for late deliveries, reflecting practical considerations. The results demonstrate that reductions on total distance traveled and/or the number of vehicles are achievable by integrating this type of flexibility. A concept related to the aforementioned flexibility is the postponement strategy, wherein deliveries to customers are deferred beyond the considered planning horizon, incurring certain penalties. This approach was initially introduced by Archetti et al. [2015] within the context of a multi-period VRP variant, characterized by a planning horizon extending over several days. Later on, Archetti et al. [2021] adapted this concept for the dynamic VRP variant to handle limitations in resources (e.g., drivers) and customers who appear dynamically in the system.

The second type of flexibility focuses on alternative delivery locations. Reyes et al. [2017] investigated a car-trunk delivery system in which the availability of delivery locations for a customer depends on the number of locations their car visits. This approach increases the density of delivery locations, thereby enhancing the potential to devise delivery plans with lower total traveled distances. Mancini and Gansterer [2021] presented a variant of VRPTW by incorporating lockers as alternative delivery locations. The presence of flexible customers who opt for both home attended and locker deliveries provides the system a greater flexibility to devise an efficient delivery plan. While car-trunk delivery ensures direct package receipt in a customer's car, locker utilization requires a customer to pick up their packages later. Consequently, compensations are typically provided to customers as an incentive for utilizing lockers [Enthoven et al., 2020]. The following works used the idea of alternative delivery locations by incorporating various features. Dumez et al. [2021a] integrated the overall satisfaction measurement in the delivery system proposed by Reyes et al. [2017], where the locations set by a customer may include lockers whose space are shared with packages of other customers. Each location is assigned a preference level by the customer, and the delivery plan must satisfy a minimum overall preference. Dragomir et al. [2022] integrated the concept into a pickup and delivery routing problem. The proposed system not only permits customers to specify their preferable delivery locations but also enables an additional recipient to determine their preferable delivery locations. Furthermore, the system incorporates a set of lockers available for receiving packages at any moment within the planning horizon, enhancing the flexibility and accessibility of delivery locations.

The aforementioned literature investigates flexibility in one-echelon distribution networks where packages are directly delivered from the depot to customers. In fact, several works involving two-echelon distribution networks have also considered similar strategies. Zhou et al. [2018] proposed an extension of the two-echelon VRP by considering locker utilization in the second echelon. Darvish et al. [2019] investigated two types of flexibility in the two-echelon distribution network: flexibility in network design and in due dates. The planning horizon spans over several days where the flexibility in network design allows a supplier to select which distribution center to be utilized in a particular day. The flexibility in due dates allows the delivery postponement to several customers, similar to the concept in Archetti et al. [2015].

2.2 Routing Problems with Parking Decisions

We classify the following literature based on their research purposes, i.e., (1) parking decisions arising from one logistics provider, and (2) investigating systems to manage parking spots shared by multiple logistics providers.

In the first category, two variants of routing problems are commonly considered: the traveling Salesman Problem (TSP) and the VRP. Inspired by a real-world delivery system, Nguyên et al. [2019] investigated a clustered TSP with time windows. In this variant, each cluster is visited by a vehicle, while walking is performed to visit other locations located within the cluster. The number of clusters and the locations associated with each cluster are predetermined. Martinez-Sykora et al. [2020] addressed a similar problem with one notable difference: the clusters are not predefined. Additionally, in the works of Nguyên et al. [2019] and Martinez-Sykora et al. [2020], only one walking subtour is allowed. While previous studies aim to minimize the weighted total of driving and walking times, Reed et al. [2024] investigated a similar problem to minimize the total completion time. Moreover, they allowed several walking subtours and only allowed the driver to park at designated parking spots. The results show that considering parking time as a part of the objective function is the key to achieving a delivery plan with the minimum completion time. Cabrera et al. [2023] considered the integration of parking-related decisions into VRPs by allowing a driver to park at a customer location to begin a walking subtour. Le Colleter et al. [2023] addressed a similar problem but this study only allows parking at available parking spots, similar to the assumption investigated in Reed et al. [2024]. The objective function in Cabrera et al. [2023] involves the total fixed costs and driving costs without walking costs, showing the interest of maximizing the number of customers visited on foot. Motivated by the findings in Reed et al. [2024], Le Colleter et al. [2023] proposed an objective function of lexicographically minimizing the number of vehicles and the total completion time. Most recently, Senna et al. [2024] investigated another variant in which each vehicle may carry the multiple deliverymen to reduce the overall operational costs.

All the aforementioned works assume that parking spots are available at all times during the planning horizon. However, due to space limitations, drivers may find that the places they intend to use are unavailable. Consequently, a centralized parking reservation system emerges as a potential solution. Mor et al. [2020] presented a study on a booking system for parking spots used by delivery vehicles, where logistics providers sequentially book available parking spots. From the perspective of a provider, a parking spot may have multiple time windows reflecting the times when parking is available. Zhang et al. [2023] analyzed a system in which a central authority controls the scheduling of parking spots, incorporating various real-world features, e.g., stochastic travel time and routing decisions for delivery on foot. Based on the systems considered, Mor et al. [2020] arguably demonstrated greater flexibility for logistics providers in the long run, as providers can perform the booking process themselves while Zhang et al. [2023] dealt with a centralized system that manages the utilization of parking spots. Despite these differences, the parking reservation system proves to be crucial in managing the scarcity of parking spots.

3 Problem Description

The FPLRP is formally defined on a directed graph $G = (V, A)$ where V is the set of locations and A is the set of arcs. Set V consists of the origin and destination depots $\{0, 0'\}$, the set N of roaming delivery locations of customers, the set L of lockers, and the set P of parking spots. We define C as the set of customers. Each customer $i \in C$ is associated with a set of roaming delivery locations $N_i \subseteq N$ where $N = \cup_{i \in C} N_i$ and can be categorized based on their willingness to pick up their packages at lockers. Thus, set C can be decomposed into two subsets of customers, i.e., the set C^R of customers who require their packages being sent to one of predefined roaming delivery locations and the set C^F of customers who also willingly pick up their packages at selected lockers in addition to the deliveries to roaming delivery locations. Each customer $i \in C^F$ selects a locker $j \in L$ and the selection is represented by a set of binary parameters m_{ij}. The arcs set A is partitioned into the set of driving arcs A^d and the set of walking arcs A^w. Each arc (i, j) in both subsets satisfies the triangular inequality and is associated with a distance d_{ij}.

Each customer $i \in C$ has a demand q_i. If the demand of customer i is delivered to location $j \in N_i$, it is preferable to be performed in a given time window $[a_j, b_j]$. The planning period takes the range $[0, T]$ which also serves as the depot operational time. An earlier or later delivery is possible with a certain flexibility level. In particular, another pair of time windows $[\hat{a}_j, \hat{b}_j]$ is defined by adding the flexibility for earliness and delay at each location. Penalty factors β'_i and β''_i are applied for each time unit violated by early and late delivery to customer i, respectively. If the package of customer i is delivered to locker $l \in L$, a compensation $\hat{\beta}_{il}$ is given to the customer. Each locker is assumed to be large enough and available throughout the entire planning period.

A set K of vehicles with homogeneous capacity Q^d is available at the depot. A delivery to a customer location can be performed by either driving or walking mode while a delivery to a locker is only performed in a driving mode. Each vehicle travels over an arc $(i, j) \in A^d$ with a given driving time t^d_{ij}. The driver must visit and park at parking spot $i \in P$ before performing the walking mode. We assume that each parking spot can only be visited at most once by any driver as in Reed et al. [2024], Cabrera et al. [2023], and Le Colleter et al. [2023].

A driver requires walking time t^w_{ij} to travel over an arc $(i, j) \in A^w$. If a vehicle visits parking spot i, the vehicle can only park during time window $[\tilde{a}_{ih}, \tilde{b}_{ih}]$ defined by the set H_i of time windows of parking spot i. Visiting more than one customer is possible before returning to the parking spot. We hence define a *walking subtour* as a subtour performed by a driver from a parking spot, visiting at least one customer, and returning to the same parking spot. We allow more than one walking subtours from one visited parking spot. A driver may carry up to Q^w units of demand in one walking subtour and may walk up to a maximum walking distance r^{max} over all walking subtours. In addition, it requires g_i time units to park a vehicle at location $i \in N \cup P \cup L$. We note that the driving time t^d_{ij} has integrated the parking time at node j. Finally, the objective of the FPLRP is to minimize the total operational costs consisting of the sum of driving costs, walking costs, compensations paid to customers assigned to lockers, and penalties from early and late deliveries. The formulation of the FPLRP and the notation used in the sets, parameters, and decision variables are all presented in Appendix A (Tables 8 –10).

4 Hybrid Large Neighborhood Search

Large Neighborhood Search (LNS) was originally proposed by Shaw [1998] to solve the capacitated VRP and the VRPTW by iteratively removing customers from a complete solution based on some criteria and re-inserting them at favorable positions. Ropke and Pisinger [2006] extended the idea by proposing Adaptive Large Neighborhood Search (ALNS) where several removal and repair operators with different criteria are embedded into the algorithm to enlarge the explored solution space. At each iteration, these operators are selected based on their historical performance in producing high-quality solutions. Various works have developed LNS with several operators to maintain strength in exploring a wider solution space, while dropping adaptiveness to preserve simplicity [Turkeš et al., 2021, Dumez et al., 2021a,b, Le Colleter et al., 2023, Arda et al., 2024]. Inspired from the aforementioned literature, we devise a metaheuristic called Hybrid Large Neighborhood Search (HLNS).

Our HLNS requires a feasible initial solution s as the input, obtained by a regret-2 insertion described later. In addition, other types of solutions are set as s, i.e., the incumbent solution s', the current solution s^c, and the best solution s^*. The main procedure of our HLNS is described in Algorithm 1. In the beginning of every iteration, a number of customers is selected to be removed. Destroy and repair operators are randomly selected and applied to s^c (line 4, detailed in Sections 4.1 and 4.2). A dynamic programming (DP) algorithm described in Section 4.3 is developed to improve the configuration of walking and driving modes of a route given a fixed sequence of visited locations. The acceptance of a newly generated solution is described in lines 7–13. We implement a Linear Threshold Acceptance in which a worse solution is still accepted to replace s' if its objective is less than a threshold of $\alpha^{\text{acc}}\sigma(s')$. Then, we periodically solve a Set Partitioning (SP) model formulated for the FPLRP (lines 15–20, described in Section 4.4), to utilize the historical information (i.e., generated solutions) for producing a high-quality solution. Finally, we employ the maximum number of iterations η^{iter} as the stopping criterion.

We now briefly describe the concept of an FPLRP solution representation used in HLNS. We define a solution $s = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{\tilde{k}}\}$ where $\mathbf{r}_i$ represents the route planned for driver i and $\tilde{k}$ is the number of assigned drivers. Each route consists of a sequence of locations starting and ending with the depot and visiting a roaming delivery location of a customer, a parking spot, and/or a locker. A route may consist of a set of walking subtours, each represented by a sequence of locations: starting with a parking spot, followed by at least one roaming delivery location, and ending with the parking spot. We also maintain a binary matrix $\mathbf{z}$ where $\mathbf{z}_{ij}$ represents whether customer i is assigned to locker j. Lastly, we keep track of the assignments of customers to vehicles. In this way, we know packages carried by a driver to a locker, if any.

Algorithm 1: HLNS Pseudocode

Input: a feasible initial solution s
Output: s^*
$s' \leftarrow s, s^* \leftarrow s, itr \leftarrow 0$
while $(itr < \eta^{\text{iter}})$ **do**
    $\eta^{\text{rem}} \leftarrow$ **RandNumRemoved**$(\eta^{\min}, \eta^{\max})$
    $s^w \leftarrow$ **DestroyRepair**(s^w) // see Sections 4.1 and 4.2
    if $\sigma(s^w) < (1 + \alpha^{\text{dp}})\sigma(s^*)$ **then**
        $s^w \leftarrow$ **DynamicProgramming**(s^w) // see Section 4.3
    end
    if $\sigma(s^w) < \alpha^{\text{acc}}\sigma(s')$ **then**
        $\xi^{\text{col}} \leftarrow$ **ConvertSoltoColumns**(s^w)
        $s' \leftarrow s^w$
        if $\sigma(s^w) < \sigma(s^*)$ **then**
            $s^* \leftarrow s^w$
        end
    end
    else
        $s^w \leftarrow s'$
    end
    $\alpha^{\text{dp}} \leftarrow$ **UpdateDPThreshold**(α^{dp})
    if itr **modulo** $\eta^{\text{SP}} = 0$ **then**
        $s^w \leftarrow$ **SolveSP**(ξ^{col}) // see Section 4.4
        if $\sigma(s^w) < \sigma(s')$ **then**
            $s' \leftarrow s^w$
            if $\sigma(s^*) < \sigma(s')$ **then**
                $s^* \leftarrow s^w$
            end
        end
    end
    $\alpha^{\text{acc}} \leftarrow$ **UpdateAccThreshold**(α^{acc})
    $itr \leftarrow itr + 1$
end

Figure 1 illustrates of a simplified FPLRP solution by only showing visited locations, i.e., 10 customers, 3 visited parking spots, and 1 visited locker. The resulting solution representation depicted in Figure 1 is $\mathbf{r}_1 = \{\mathrm{D0}, \mathrm{C1}, \mathrm{P2}_1, \mathrm{C8}, \mathrm{C6}, \mathrm{P2}_2, \mathrm{P2}_3, \mathrm{C3}, \mathrm{P2}_4, \mathrm{P3}_1, \mathrm{C4}, \mathrm{P3}_2, \mathrm{L1}_1, \mathrm{D0}\}$ and $\mathbf{r}_2 = \{\mathrm{D0}, \mathrm{L1}_2, \mathrm{P1}_1, \mathrm{C10}, \mathrm{C5}, \mathrm{P1}_2, \mathrm{D0}\}$. In r_1, four copies of parking spot P2 are created to accommodate two walking subtours. A similar approach is applied to parking spots P1 and P3. Two copies of lockers, i.e., $\mathrm{L1}_1$ and $\mathrm{L1}_2$, are created to represent the locker L1, visited by drivers 1 and 2, respectively. In matrix $\mathbf{z}$, only $\mathbf{z}_{\mathrm{C2L1}}, \mathbf{z}_{\mathrm{C7L1}}$, and $\mathbf{z}_{\mathrm{C9L1}}$ equal 1, indicating that packages of customers C2, C7, and C9 are delivered to L1.

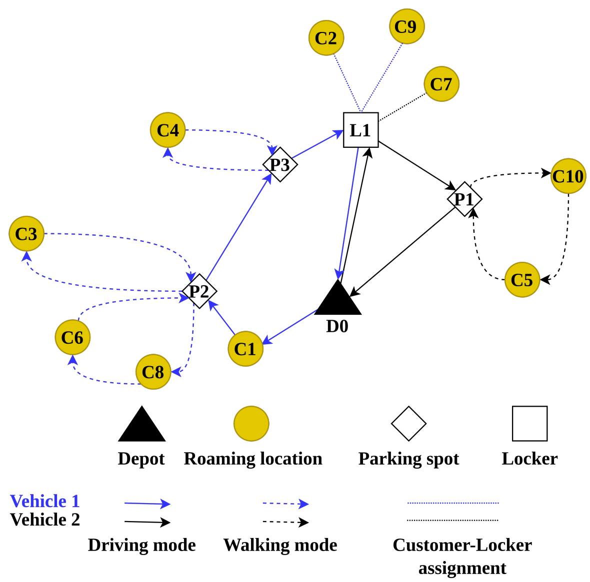


Figure 1: An illustration of an FPLRP solution

4.1 Destroy Phase

A removal operator implemented in the destroy phase generally aims to remove a subset of visited customers (lockers and parking spots are possibly removed from the solution). The removed customers are then placed in the list $\mathcal{L}^r$. In addition, we define $\mathcal{L}^e$ as the list of remaining customers in the solution. The implemented destroy operators (adopted from Ropke and Pisinger [2006], Pisinger and Ropke [2007], Christiaens and Vanden Berghe [2020]) are further explained as follows.

- **Random Removal** aims to diversify the search by randomly removing a subset of customers.

- **Worst Removal** removes customers that significantly contribute to the objective function of the solution. The contribution of a customer is measured by the difference of objective values between the original solution and the solution where the customer is removed. In each iteration, the contribution of each remaining customer in the modified route is recalculated. Then, customers in $\mathscr{L}^e$ are sorted in descending order based on their contributions, and the operator selects the customer in the $\lfloor U(0,1)^{p_{\text{worst}}}|\mathscr{L}^e|\rfloor$-th position to be removed. In our work, we exclude earliness and delay penalties from the calculation of contribution to reduce the computational burden in the implementation.

- **Proximity-based Removal** removes a set of customers located close to each other. In each iteration, a customer is selected from $\mathscr{L}^r$ as a seed and customers in $\mathscr{L}^e$ is sorted based on the distance to the seed customer in ascending order. The customer in the $\lfloor U(0,1)^{p_{\text{related}}}|\mathscr{L}^e|\rfloor$-th position is then removed from $\mathscr{L}^e$ and placed in $\mathscr{L}^r$. In the beginning (when $\mathscr{L}^r = \emptyset$), the first customer to be removed is randomly selected from $\mathscr{L}^e$ and is stored in $\mathscr{L}^r$.

- **Time-based Removal** has a similar procedure to that of the Proximity-based Removal but uses a different sorting criterion: the difference in service start time.

- **Cluster Removal** iteratively selects a route and executes the Kruskal Algorithm to build a minimum spanning tree over visited locations (customers and lockers). The Kruskal Algorithm terminates once two connected components are found and all locations from a randomly selected component are removed from the route. We exclude parking spots from consideration to prevent the removal of customers from the other connected component. A parking spot is removed only when all of its associated customers are removed.

- **Split String Removal** iteratively removes a sequence of locations (string) in a route by preserving a subset of locations (substring). Two parameters, i.e., the maximum cardinality of the removed strings $S^{\max}$ and average number of removed locations $\bar{\ell}$, are required to determine the cardinality of strings to be removed and the number of routes to be considered while parameter $\mathscr{B}$ is needed to determine the preserved substring. In our implementation, a string consists of locations visited by driving mode. If the selected location is a parking spot (locker), then its associated customers are all removed along with the parking spot (locker).

The first four operators terminate after η^{rem} customers are removed while the cluster removal stops when at least η^{rem} customers are removed. Lastly, the split string removal utilizes the original criterion in Christiaens and Vanden Berghe [2020], i.e., either all customers have been considered or the number of routes to be considered is reached.

In our work, we create two additional removal operators derived from the random removal and worst removal operators to specifically remove lockers and visited parking spots. These operators remove lockers (parking spots) alongside with their associated customers until at least η^{rem} customers have been removed or no lockers (parking spots) remain in the solution.

4.2 Repair Phase

After removing customers from the solution, they are re-inserted by a randomly selected repair operator, listed in Section 4.2.1. Section 4.2.2 explains various insertion mechanisms that can be implemented depending on a particular position in a route.

4.2.1 Repair Operators

We implement two repair operators proposed in Ropke and Pisinger [2006] in which insertion mechanisms described in Section 4.2.2 are embedded.

- **Best Insertion** iteratively selects the customer in the list $\mathcal{L}^r$ resulting in the least insertion cost. A perturbed version is also considered by adding a noise factor as described in Ropke and Pisinger [2006].

- **Regret Insertion** improves the Best Insertion by considering a criterion that incorporates look-ahead information, i.e., regret value. A regret value is calculated by summing up the differences between the lowest insertion cost and the j-th lowest insertion cost, where $j \in 2, \ldots, k$. In our HLNS, we also implement the perturbed version and the considered k values are 2 and $|K|$.

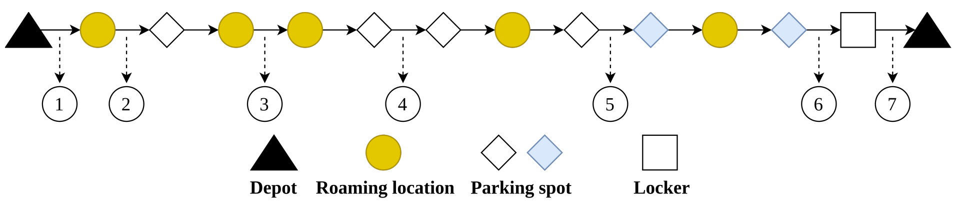
4.2.2 Insertion Mechanism

A feasible route in an FPLRP solution contains a sequence of locations visited in either driving or walking mode. The visited locations may consist of customers' roaming delivery locations, parking spots, and/or lockers. Considering the diversity of locations visited, we propose a set of insertion mechanisms which are executed depending on the given insertion position in a route.

- **T-1 Insertion** evaluates the insertion of each roaming delivery location of a customer at the given insertion position. Additionally, we adopt the idea proposed in Reyes et al. [2017] to further extend the search space by evaluating all combinations of roaming delivery locations among the currently evaluated customer, the predecessor of the insertion position, and the successor. In other words, if neither the predecessor nor the successor is a customer, this insertion mechanism will evaluate only the roaming delivery locations of the customer to be inserted.

- **T-2 Insertion** evaluates the insertion of each roaming delivery location of a customer with every currently unused parking spot at the given insertion position. This type of insertion is allowed only at positions where both the predecessor of the insertion position and the successor are visited in driving mode. Each time window of the parking spot is evaluated alongside the given roaming delivery location. Indeed, all combinations between time windows of the parking spot and roaming delivery locations of the customer are evaluated.

- **T-3 Insertion** is similar to Type-2 Insertion but the considered parking spot is the location visited at the predecessor or the successor of a given position. In addition, to perform this insertion, the driver must be in the driving mode.
- **T-4 Insertion** evaluates the insertion of a locker, which is compatible to the currently evaluated customer, into the given position. This insertion considers only a subset of lockers that have not been visited in the route associated with the given position. Evaluating roaming delivery locations of the customer becomes unnecessary because the visited location is a locker.
- **T-5 Insertion** evaluates the assignment of the currently evaluated customer to an existing locker, provided the locker is compatible. This insertion occurs only if the predecessor of the specified position is a locker.



Position	Insertion Mechanisms
1	T-1, T-2, T-4
2	T-1, T-2, T-3, T-4
3	T-1
4	T-3
5	T-1, T-2, T-3, T-4
6	T-1, T-2, T-3, T-4
7	T-1, T-2, T-4,T-5

Figure 2: An illustration of positions requiring various insertion mechanisms

Figure 2 provides the illustration of different positions requiring different types of insertion in a route. This illustration is derived from $\mathbf{r}_1$ in Figure 1 by dropping the indices and uses different colors to represent different parking spots. T-1 insertion is implemented in almost all positions in Figure 1 (except position 4). Inserting a customer in position 3 illustrates the case when all combinations involving roaming deliveries of a given customer, its predecessor, and its successor are evaluated while evaluating the combinations of only the given customer is portrayed in the insertion scenario in position 5. In position 4, only T-3 Insertion can be implemented, i.e., inserting the given customer by considering only the first visited parking spot (pictured as a white diamond). Performing other types of insertion mechanisms is forbidden since the associated driver still has another walking subtour originating from the first visited parking spot.

After each insertion is performed, a new neighbor solution is created, which might not be feasible. As our HLNS requires a feasible solution to continue, we implemented a procedure to verify the feasibility of the solution with respect to time-related constraints, maximum walking distance, walking capacity, and driving capacity. If the solution is feasible, then we evaluate its objective. Otherwise, the inserted customer is removed again, added to list $\mathcal{L}^r$, its infeasible position becomes forbidden for the particular insertion mechanism that generated the infeasible solution, and the same repair operator is applied again. This process continues until all customers are inserted and the solution is feasible. Note that because we have an unlimited fleet, at least one feasible position exists, i.e., opening a new route serving only the customer to be inserted.

4.3 Dynamic Programming

In order to improve the configuration of driving and walking modes, we propose a tailored DP algorithm inspired from Schiffer and Walther [2018] and Parragh and Cordeau [2017] into our HLNS. Given a route in an FPLRP solution, we first extract the sequence of locations visited involving customers (and lockers if any) and the list of parking spots visited in other routes of the given solution. Then, the DP works by expanding a limited search tree on the given sequence. Since a customer may have more than one roaming delivery location, the search tree expands by considering combinations of locations. Given two consecutive locations, i.e., i and j, where each location can be either a roaming delivery location of a customer or a locker, four types of extensions are presented as follows.

- **E1**: the driver visits j from i by driving mode. If location i is visited in walking mode, then the driver returns to the last parking spot before performing the driving mode.
- **E2**: the driver visits j from i by walking mode under the current walking subtour of the last visited parking spot. This extension can only be performed if the driver is in the walking mode.
- **E3**: the driver visits j from i by walking mode under a new walking subtour from the last visited parking spot. Similar to E2, this extension can only be performed if the driver is in the walking mode.
- **E4**: the driver visits j from i by walking mode from an unused parking spot $p \in P$ with a selected time window $h \in H_p$. If i is visited in walking mode, then the driver returns to the last parking spot and visits the new parking spot by driving mode.

An extension requires a label with a set of components described in Table 1. In order to perform the four extensions, we propose a set of resource extension functions described in B to extend θ, a label of location i, to θ', a label of location j, under a particular extension.

The expansion of the search tree can be limited (thereby, reducing computational time) by implementing the feasibility procedures and dominance criteria. In our DP, the feasibility procedures presented in Table 2 depend on which extension is selected to produce θ' from θ where $B_j = T, \forall j \in L; B_j = \hat{b}_j$ otherwise. Regarding the dominance rules,

Table 1: Label components for REFs

Notation	Description
$\mathcal{T}^{\text{cur}}(\theta)$	the current location of label θ
$\mathcal{T}^{\text{cost}}(\theta)$	the total costs from the depot up to $\mathcal{T}^{\text{cur}}(\theta)$
$\mathcal{T}^{\text{pen}}(\theta)$	the minimum penalty of the partial route in label θ
$\mathcal{T}^{\text{time}}(\theta)$	the arrival time at $\mathcal{T}^{\text{cur}}(\theta)$
$\mathcal{T}^{\text{vParks}}(\theta)$	visited parking spots up to $\mathcal{T}^{\text{cur}}(\theta)$
$\mathcal{T}^{\text{mode}}(\theta)$	current mode when $\mathcal{T}^{\text{cur}}(\theta)$ is visited, i.e., $\mathscr{d}$ denotes driving mode and $\mathscr{w}$ represents walking mode
$\mathcal{T}^{\text{cPark}}(\theta)$	the last parking spot visited if current mode is walking mode
$\mathcal{T}^{\text{dist}}(\theta)$	total walking distance up to $\mathcal{T}^{\text{cur}}(\theta)$
$\mathcal{T}^{\text{wd}}(\theta)$	total demands up to $\mathcal{T}^{\text{cur}}(\theta)$ in the current walking subtour if current mode is walking mode
$\mathcal{T}^{\text{ParkTW}}(\theta)$	the selected TW of the last parking spot if current mode is walking mode

label θ_2 is dominated by θ_1 if conditions (1)–(5) hold. These rules can be implemented if several conditions are met, further detailed in Algorithm 2.

Table 2: Feasibility procedures

Condition	**Rules**
$e_{ij} \in \text{E1} \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w}$	$(\mathcal{T}^{\text{dist}}(\theta') \leq r^{max}) \wedge (\mathcal{T}^{\text{time}}(\theta) + t''_{i\mathcal{T}^{\text{cPark}}(\theta)} \leq \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta)\mathcal{T}^{ParkTW}(\theta)}) \wedge (\mathcal{T}^{\text{time}}(\theta') \leq B_j)$
$e_{ij} \in \text{E1} \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d}$	$\mathcal{T}^{\text{time}}(\theta') \leq B_j$
$e_{ij} \in \text{E2} \cup \text{E3}$	$(\mathcal{T}^{\text{wd}}(\theta') \leq Q'') \wedge (\mathcal{T}^{\text{dist}}(\theta') \leq r^{max}) \wedge (\mathcal{T}^{\text{time}}(\theta') \leq \hat{b}_j) \wedge (\mathcal{T}^{\text{time}}(\theta') + t''_{j\mathcal{T}^{\text{cPark}}(\theta')} < \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta')\mathcal{T}^{\text{ParkTW}}(\theta')})$
$e_{ij} \in \text{E4} \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w}$	$(\mathcal{T}^{\text{wd}}(\theta') \leq Q'') \wedge (\mathcal{T}^{\text{dist}}(\theta') \leq r^{max}) \wedge (\mathcal{T}^{\text{time}}(\theta) + t''_{i\mathcal{T}^{\text{cPark}}(\theta)} \leq \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta)\mathcal{T}^{\text{ParkTW}}(\theta)}) \wedge (\mathcal{T}^{\text{time}}(\theta') \leq \hat{b}_j) \wedge (\mathcal{T}^{\text{time}}(\theta') + t''_{j\mathcal{T}^{\text{cPark}}(\theta')} \leq \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta')\mathcal{T}^{\text{ParkTW}}(\theta')})$
$e_{ij} \in \text{E4} \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d}$	$(\mathcal{T}^{\text{wd}}(\theta') \leq Q'') \wedge (\mathcal{T}^{\text{dist}}(\theta') \leq r^{max}) \wedge (\mathcal{T}^{\text{time}}(\theta) + t'_{i\mathcal{T}^{\text{cPark}}(\theta')} \leq \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta')\mathcal{T}^{\text{ParkTW}}(\theta')}) \wedge (\mathcal{T}^{\text{time}}(\theta') \leq \hat{b}_j) \wedge (\mathcal{T}^{\text{time}}(\theta') + t''_{j\mathcal{T}^{\text{cPark}}(\theta')} \leq \tilde{b}_{\mathcal{T}^{\text{cPark}}(\theta')\mathcal{T}^{\text{ParkTW}}(\theta')})$

$$\mathcal{T}^{\text{cost}}(\theta_1) \leq \mathcal{T}^{\text{cost}}(\theta_2) \tag{1}$$

$$\mathcal{T}^{\text{time}}(\theta_1) \leq \mathcal{T}^{\text{time}}(\theta_2) \tag{2}$$

$$\mathcal{T}^{\text{dist}}(\theta_1) \leq \mathcal{T}^{\text{dist}}(\theta_2) \tag{3}$$

$$\mathcal{T}^{\text{wd}}(\theta_1) \leq \mathcal{T}^{\text{wd}}(\theta_2) \tag{4}$$

$$\mathcal{T}^{\text{vParks}}(\theta_1) \subseteq \mathcal{T}^{\text{vParks}}(\theta_2). \tag{5}$$

We note that the proposed dominance rules are heuristic due to the presence of $\mathcal{T}^{\text{pen}}$ in $\mathcal{T}^{\text{cost}}$. Criterion (1) performs the comparison with minimum penalty costs between two

Algorithm 2: Dominance rules condition

Input: Label θ_1, Label θ_2
Output: ApplyDominance

ApplyDominance $\leftarrow$ *false*
if $\mathcal{T}^{\text{mode}}(\theta_1) = \mathscr{d}$ **and** $\mathcal{T}^{\text{mode}}(\theta_2) = \mathscr{d}$ **then**
    ApplyDominance $\leftarrow$ *true*
end
else if $\mathcal{T}^{\text{mode}}(\theta_1) = \mathscr{w}$ **and** $\mathcal{T}^{\text{mode}}(\theta_2) = \mathscr{w}$ **then**
    if $\mathcal{T}^{\text{cPark}}(\theta_1) = \mathcal{T}^{\text{cPark}}(\theta_2)$ **and** $\mathcal{T}^{\text{ParkTW}}(\theta_1) = \mathcal{T}^{\text{ParkTW}}(\theta_2)$ **then**
        ApplyDominance $\leftarrow$ *true*
    end
end
return ApplyDominance

labels which can occur at different times. In addition, to further reduce the computational time of DP implementation, we set the cost upper bound ub_{DP} obtained from the cost of the original route serving as the DP input. Any label θ with $\mathcal{T}^{\text{cost}}(\theta) > ub_{\text{DP}}$ is not further extended.

4.4 Set Partitioning Model

We recall the column pool ξ^{col} in which routes from generated FPLRP solutions are collected. Let λ_j represent the decision variable whether the j-th column in ξ^{col} is selected and γ_j is the cost associated with λ_j. Binary parameter ω_{ij} indicates whether a roaming delivery location of customer i is visited in λ_j and binary parameter ϕ_{ij} indicates whether parking spot i is visited in λ_j. The SP model for FPLRP is formulated in (6)–(9) with the objective of minimizing the total costs of selected columns.

$$\text{Min Z} = \sum_{j \in \xi^{col}} \gamma_j \lambda_j \tag{6}$$

subject to:

$$\sum_{j \in \xi^{col}} \omega_{ij} \lambda_j = 1, \forall i \in C \tag{7}$$

$$\sum_{j \in \xi^{col}} \phi_{ij} \lambda_j \leq 1, \forall i \in P \tag{8}$$

$$\lambda_j \in \{0, 1\}, \forall j \in \xi^{col}. \tag{9}$$

Constraints (7) ensure that the selected columns visit all customers while constraints (8) guarantee that each parking spot is utilized at most once over all selected columns. Lastly, constraints (9) define the possible values of decision variables λ_j.

5 Computational Results

Our experiments were conducted in a computing cluster with AMD EPYC^{TM} Rome 7532 processors running at 2.4 GHz using 12 threads and up to 48 GB of RAM. Both MILP and HLNS are coded in C++ and compiled using g++. Since FPLRP is a new variant of routing problem, a new set of benchmark instances is generated by following the literature. We generated two sets of instances, i.e., small and large, based on the number of customers. The small instances involve 5, 10, and 15 customers, with each scenario comprising 10 different instances. Similarly, the large instances consist of 20, 40, 60, and 80 customers, with each scenario also including 10 different instances. In total, we have 70 FPLRP instances. All locations are generated within a square grid of 10×10 km where the depot is located in the center with an operational time of 420 min (7 hours).

The procedure for generating customer-related information is inspired from Reyes et al. [2017] and Taş et al. [2014]. Each customer in a small instance has one to two locations, while that in a large instance has one to five locations. One of the generated locations is designated as the home location. If only one location is generated for a customer, it automatically becomes the home location and the associated time windows (both preferable and acceptable) equal the operational time of the depot. Otherwise, the customer follows a sequence starting from the home location, visiting the other generated locations, and then returning to the home location. The preferable time window of each location of a customer is designed in a way that no time overlapping among locations occurs, i.e., $a_j = t'_{ij} + b_i + \delta_{\text{pref}}$ where $\delta_{\text{pref}} = 60$ and $\delta_{\text{pref}} = 30$ for small and large instances, respectively, i and j are both locations of a customer, and location j is visited after location i. The acceptable time window is adjusted from the preferable time window by adding parameter $\delta_{\text{acc}} = 10$, i.e., $\hat{a}_i = \max(0, a_i - \delta_{\text{acc}})$, and $\hat{b}_i = \min(T, b_i + \delta_{\text{acc}}), \forall i \in N$. The demand of each customer is randomly generated within the range of 1 to 5.

The number of parking spots generated for small and large instances are 4 and $\lceil 0.3 \times (N+1) \rceil$, respectively. The value 0.3 is inspired by the proportion of available parking spots considered in Le Colleter et al. [2023], and $N+1$ represents the total number of locations, including the depot. Each parking spot is generated within a radius of 300 m from a randomly selected location, and each location is selected only once for this purpose to ensure that parking spots are not concentrated around a single location. The number of time windows for each parking spot is set to 2 for small instances and up to 3 for large instances. In order to generate the time window, we utilize a parameter π_{unavail} representing the percentage of each parking spot's unavailable time, set at 50%. We then distribute the total unavailable time according to the number of unavailable period assigned to each parking spot.

Four lockers are generated for large instances by following the locations mentioned in Mancini and Gansterer [2021], while each small instance has one locker. The number of customers who willingly pick up their packages is set to $\lceil \pi_{\text{flex}} \times C \rceil$ where π_{flex} equal to 0.25 and 0.50 for small and large instances, respectively. Selected customers are assumed to select lockers close to their home locations.

Vehicles travel at 30 km/h and the driver walks at 4 km/h. It takes 5 min to park at a parking spot and the time to park at a location of a customer and a locker is set to 10 min. The maximum walking distance for each driver over the operational time is 5 km. The vehicle capacity is set to 50 and the walking capacity for each walking subtour is set to 10. We set $(c^d, c^w) = (1.0, 0.5)$ and $(c^d, c^w) = (1.0, 0.75)$ for small and large instances, respectively, $(\beta'_i, \beta''_i) = (0.5, 1.0)$ as proposed in Taş et al. [2014], and $\tilde{\beta} = 5.0$.

The method utilized to obtain final configuration of HLNS parameters is preented in Section 5.1; evaluations on the effectiveness of HLNS are discussed in Sections 5.2 and 5.3; and managerial insights derived by performing sensitivity analysis over various problem parameters are presented in Section 5.4.

5.1 Parameter Setting

We tune HLNS parameters by varying one parameter at a time while fixing the values of remaining ones. Five instances from each set of 20, 40, 60, and 80 customers (20 instances in total) are selected and HLNS is executed five times on the selected instances. We set the final value for each parameter resulting in the lowest average of overall best objective values. Parameters in removal operators, i.e., $p_{\text{worst}}, p_{\text{related}}, S^{\max}, \bar{\ell}$, and $\mathscr{B}$, are fixed based on the values set in Ropke and Pisinger [2006] and Christiaens and Vanden Berghe [2020]. The detailed results of parameter tuning are reported in Appendix C and the final configuration is reported in Table 3.

Table 3: Final configuration for HLNS parameters

Notation	Definition	Value
$(\eta^{\min}, \eta^{\max})$	Range of customers to be removed	(0.1, 0.4)
$(\alpha^{\text{dp}}_{\text{init}}, \alpha^{\text{dp}}_{\text{min}})$	Initial and minimum thresholds of DP	(1.025, 1.01)
η^{dp}	Maximum nonimprovement of DP	5
ξ^{dp}	Threshold multiplier of DP	0.5
$(\alpha^{\text{acc}}_{\text{init}}, \alpha^{\text{acc}}_{\text{final}})$	Initial and final acceptance thresholds	(1.025, 1.001)
η^{SP}	Number of iterations to solve SP model	2000
η^{iter}	HALNS iterations	20000
$p_{\text{worst}}*$	Randomness parameter of worst removal operator	3
$p_{\text{relatedness}}*$	Randomness parameter of proximity-based & time-based operators	6
$S^{\max}*$	Maximum cardinality of the removed strings	10
$\bar{\ell}*$	Average number of removed locations	10
$\mathscr{B}*$	Probability of preserving a node in a selected string	0.9

* denotes values from the literature [Ropke and Pisinger, 2006, Christiaens and Vanden Berghe, 2020].

5.2 Evaluating HLNS effectiveness on special cases of the FPLRP

We evaluate the performance of HLNS by solving two special cases from the literature, namely the vehicle routing with roaming delivery locations (VRPRDL) [Reyes et al., 2017] and the park-and-loop routing problem (PLRP) [Coindreau et al., 2019]. For the VRPRDL, we utilized four datasets used in Ozbaygin et al. [2017]. Overall, there are

120 instances of the VRPRDL with the number of customers ranging from 15 to 120. The PLRP datasets can be classified into small, medium, and large. The PLRP small and medium datasets are proposed by Coindreau et al. [2019] with number of customers ranging from 20 up to 50. The PLRP large datasets, with the number of customers ranging from 60 to 90, are developed by Cabrera et al. [2023], following the data generation rules in Coindreau et al. [2019]. In total, there are 80 PLRP instances. All these datasets are publicly accessible.

We run the HLNS five times for each VRPRDL and PLRP instance. Before executing the HLNS, we adjust the algorithm with minimal adjustments to solve the VRPRDL and PLRP. First, we only consider the T-1 insertion in each repair operator, as the other insertion mechanisms are unnecessary. Regarding the PLRP, since there is a limitation on number of drivers and walking cost is ignored from the objective function, our preliminary experiments show that considering other insertions (i.e., T-2 and T-3 insertions) may result in infeasible solutions. In order to cope with the limitation of vehicles in VRPRDL and PLRP instances, we introduce penalty to the objective value of a solution for each unserved customer. The penalty is set to $2.0 \times \max_{(i,j)\in A} d_{ij}$. Lastly, we fixed α^{dp} at 1.2 by deactivating the function to update its value as expressed in line 14 of Algorithm 1. The value is significantly larger than the final configuration used for solving PLRP as the computational burden of the DP for VRPRDL and PLRP is lower (no computation related to penalties for early and late deliveries). Moreover, the HLNS solely relies on the DP to improve the configuration of driving and walking modes for PLRP solutions since T-1 insertion only considers the insertion of customer locations.

Table 4 shows the results produced by HLNS on the VRPRDL instances and existing state-of-the-art algorithms. In particular, we include the results found by YCOSV21 [Yuan et al., 2021], DTILP21 [Dumez et al., 2021b], and PHVN22 [Pham et al., 2022]. Column BKS shows the average of best known solutions obtained from the state-of-the-art algorithms for each set of instances. We present three metrics, namely the average best solutions f^b, the overall average f^{ave}, and the average computation time in seconds $\mathsf{T(s)}$. Yuan et al. [2021] proposed a deterministic heuristic, thus only one replication was conducted for each instance, and only solved three sets of instances (i.e., B1, B2, and B3). Based on Table 4, our HLNS managed to match all BKSs in both f^b and f^{ave} of sets B1, B3, and B4. Compared to other algorithms which also obtain all BKSs for those sets [Dumez et al., 2021b, Pham et al., 2022], our HLNS requires the lowest computation time. Regarding set B2, our HLNS only failed to match 4 out of 40 instances with a gap ranging from 0.03% up to 1.82%. The detailed results of our HLNS on solving VRPRDL instances are shown in Appendix C.

Table 4: Performance of HLNS on solving VRPRDL benchmark instances

Instance	BKS	YCOSV21			DTILP21			PHVN22			HLNS (Our work)				
Set		f^b	f^{ave}	T(s)	f^b	f^{ave}	T(s)	f^b	f^{ave}	T(s)	f^b	f^{ave}	T(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
B1	3061.65	3062.40	-	17.0	3061.65	3062.33	120.0	3061.65	3061.73	530.6	3061.65	3061.69	49.3	**0.00**	**0.00**
B2	2253.35	2256.95	-	65.0	2255.40	2258.28	120.0	2253.32	2255.32	568.5	2255.88	2256.11	62.3	0.07	0.08
B3	2585.05	2585.25	-	3.3	2585.05	2585.20	60.0	2585.05	2585.05	168.1	2585.05	2585.05	17.6	**0.00**	**0.00**
B4	2447.70	-	-	-	2447.70	2450.71	60.0	2447.70	2448.20	170.3	2447.70	2447.70	13.2	**0.00**	**0.00**

Table 5 presents the results obtained by HLNS on the PLRP instances. The results from existing state-of-the-art algorithms, two heuristics, namely CCM22 [Cabrera et al., 2022], CDLP23 [Le Colleter et al., 2023], and one exact algorithm, i.e., CCM23 [Cabrera et al., 2023], are also presented. CCM22 and CDLP23 only provided solutions to small and medium instances while CCM23 provided solutions to all instances. Based on Table 5, our HLNS outperforms state-of-the-art heuristics in terms of both average best solution and computation time. While state-of-the-art heuristics require more than 100 seconds on average to solve an instance of 50 customers, our HLNS requires 20.7 seconds on average. CCM23 outperforms our HLNS only in the set of 50 customers while it requires a significantly longer computation time to obtain such solutions. For larger instances (i.e., sets with 60 to 90 customers), our HLNS outperforms CCM23 in terms of both solution quality (f^b and f^{ave}) and computation time. Moreover, our HLNS only requires 64.5 seconds on average to solve an instance from the largest set while CCM23 requires 2-hour computation time. The detailed results for each PLRP instance are provided in the Appendix C. In total, we obtain 37 new BKSs for PLRP instances with the largest improvement of up to 6.12%. To sum up, our HLNS shows a highly competitive performance compared to state-of-the-art algorithms specifically developed for solving the VRPRDL and PLRP.

Table 5: Performance of HLNS on solving PLRP benchmark instances

Instance	BKS	CCM22		CDLP23		CCM23		HLNS (Our work)				
Sets		f^b	T(s)	f^b	T(s)	f^b	T(s)	f^b	f^{ave}	T(s)	Gap^b(%)	Gap$^{\text{ave}}$(%)
20	36.7138	36.7138	13.4	36.7138	15.0	36.7138	28.1	36.7138	36.7257	2.9	0.00	0.03
30	45.3240	45.3279	35.9	45.3262	30.0	45.3240	75.9	45.3240	45.3700	6.5	0.00	0.10
40	56.8403	56.9252	56.1	56.9649	60.0	56.8403	288.7	56.8403	56.9540	10.7	0.00	0.20
50*	61.7972	62.3528	110.5	61.8851	120.0	61.7972	1877.4	61.8283	62.2243	20.7	0.05	0.69
60*	67.0129	-	-	-	-	67.0129	5460.1	66.2473	66.3897	30.8	**-1.14**	**-0.93**
70*	77.7436	-	-	-	-	77.7436	7200.0	76.5586	76.8256	37.9	**-1.52**	**-1.18**
80*	93.4416	-	-	-	-	93.4416	7200.0	89.5140	90.0480	46.6	**-4.20**	**-3.63**
90*	91.6101	-	-	-	-	91.6101	7200.0	87.7335	88.2030	64.5	**-4.23**	**-3.72**

* denotes the set in which new BKSs are found.

5.3 Evaluating the HLNS performance on FPLRP instances

In order to evaluate the performance of HLNS on solving the newly generated FPLRP instances, we solved the MILP proposed in Section 3 using Gurobi for up to 2h. We executed our HLNS five times. One parameter that we need to set to execute the MILP in Gurobi is the number $|W|$ of walking subtours allowed for each driver. Based on our preliminary experiments, setting $|W|$ to two is enough since increasing its value did not change the number of walking subtours performed by employed drivers in the given solutions. Table 6 shows the detailed results. Gurobi was able to obtain optimal results for instances with up to 10 customers in which the number of locations (column N) ranges between 7–26 and the number of parking spots (column P) equals to 4. Our HLNS obtained all those optimal solutions, as shown in Table 6. Gurobi failed to provide optimal solutions for instances with 15 customers but successfully obtained feasible solutions, with the optimality gap ranging from 2.93% to 23.40%, as shown in column Opt(%). For these instances, our HLNS obtained the same solutions for 5 instances and improved the upper bound for the remaining 5. The computation time required by Gurobi increased significantly as the size of instances increased while the increase in computation time of our HLNS was reasonable in comparison. On average, Gurobi required 2443.7 seconds to solve a small instance, while our HLNS only required 5.20 seconds.

Table 6: Performance of HLNS on solving FPLRP small instances

Instance	C	N	P	Gurobi				HLNS			Gap^b(%)	Gap^{ave}(%)
				UB	LB	Opt (%)	T(s)	f^b	f^{ave}	T(s)		
FPLRP_5_1	5	9	4	92.27	92.27	0.00	1.5	92.27	92.27	0.8	0.00	0.00
FPLRP_5_2	5	9	4	88.52	88.52	0.00	0.8	88.52	88.52	0.8	0.00	0.00
FPLRP_5_3	5	11	4	75.94	75.94	0.00	1.8	75.94	75.94	0.8	0.00	0.00
FPLRP_5_4	5	9	4	94.21	94.21	0.00	1.2	94.21	94.21	0.7	0.00	0.00
FPLRP_5_5	5	15	4	79.28	79.28	0.00	7.6	79.28	79.28	0.7	0.00	0.00
FPLRP_5_6	5	7	4	85.41	85.41	0.00	1.4	85.41	85.41	0.6	0.00	0.00
FPLRP_5_7	5	11	4	86.33	86.33	0.00	6.3	86.33	86.33	0.9	0.00	0.00
FPLRP_5_8	5	11	4	79.69	79.69	0.00	2.7	79.69	79.69	0.8	0.00	0.00
FPLRP_5_9	5	7	4	76.36	76.36	0.00	4.8	76.36	76.36	1.4	0.00	0.00
FPLRP_5_10	5	11	4	90.38	90.38	0.00	3.8	90.38	90.38	0.9	0.00	0.00
FPLRP_10_1	10	26	4	152.03	152.03	0.00	554.1	152.03	152.03	3.3	0.00	0.00
FPLRP_10_2	10	18	4	152.97	152.97	0.00	116.5	152.97	152.97	2.0	0.00	0.00
FPLRP_10_3	10	22	4	143.51	143.51	0.00	36.7	143.51	143.51	3.1	0.00	0.00
FPLRP_10_4	10	18	4	142.22	142.22	0.00	56.2	142.22	142.22	3.3	0.00	0.00
FPLRP_10_5	10	18	4	126.60	126.60	0.00	35.7	126.60	126.60	4.4	0.00	0.00
FPLRP_10_6	10	20	4	139.34	139.34	0.00	166.6	139.34	139.34	3.8	0.00	0.00
FPLRP_10_7	10	22	4	144.22	144.22	0.00	92.8	144.22	144.22	3.2	0.00	0.00
FPLRP_10_8	10	18	4	154.69	154.69	0.00	17.4	154.69	154.69	2.1	0.00	0.00
FPLRP_10_9	10	22	4	145.46	145.46	0.00	42.0	145.46	145.92	3.3	0.00	0.31
FPLRP_10_10	10	22	4	145.57	145.57	0.00	152.5	145.57	145.57	3.8	0.00	0.00
FPLRP_15_1	15	35	4	197.76	188.40	4.73	7201.1	197.76	197.76	10.3	0.00	0.00
FPLRP_15_2	15	29	4	190.48	168.16	11.72	7200.5	189.60	189.60	9.7	-0.46	-0.46
FPLRP_15_3	15	27	4	199.88	185.88	7.00	7200.5	199.64	199.64	9.0	-0.12	-0.12
FPLRP_15_4	15	33	4	202.11	196.18	2.93	7200.8	202.11	202.11	9.7	0.00	0.00
FPLRP_15_5	15	33	4	202.93	155.45	23.40	7200.9	197.74	197.74	9.1	-2.56	-2.56
FPLRP_15_6	15	31	4	199.95	182.88	8.54	7200.7	198.52	198.52	7.7	-0.72	-0.72
FPLRP_15_7	15	23	4	201.80	191.80	4.96	7200.6	201.80	201.80	5.9	0.00	0.00
FPLRP_15_8	15	33	4	207.52	187.83	9.49	7200.7	207.41	207.41	17.5	-0.05	-0.05
FPLRP_15_9	15	25	4	185.55	165.34	10.89	7200.8	185.55	185.55	10.2	0.00	0.00
FPLRP_15_10	15	31	4	193.27	168.65	12.74	7200.7	193.27	193.27	26.2	0.00	0.00
Average							2443.7			5.2	-0.13	-0.12

We also validate the performance of the algorithmic components which we embedded in our HLNS, namely DP and SP, for the purpose of further improving the quality of solutions. We modified HLNS into two variants, i.e., LNS_{DP} (with only DP algorithm) and LNS_{SP} (with only SP model). Our experimental results were conducted over all large instances with the number of customers ranging from 20 to 80. Table 7 summarizes the results while the detailed results are shown in the Appendix C. Three metrics are recorded, namely the average of best solutions (f^b), the overall average (f^{ave}), and the computation time in seconds (T(s)). Based on Table 7, integrating DP and SP resulted in solutions of better quality, as shown in the columns of f^b and f^{ave}. We also observed that the impact of DP is more significant compared to that of SP as both f^b and f^{ave} of LNS_{DP} are all lower than those of LNS_{SP}. Both DP and SP contribute to the overall HLNS complexity, hence an increase of computation time is expected. As shown in Table 7, on average, the HLNS required 5 minutes longer compared to its counterparts (LNS_{DP} and LNS_{SP}) which is acceptable given the improvement in solution quality they provide, justifying their development and use.

Table 7: Performance of HLNS on solving FPLRP large instances

Instance	C	N	P	HLNS			LNS_{DP}			LNS_{SP}		
				f^b	f^{ave}	T(s)	f^b	f^{ave}	T(s)	f^b	f^{ave}	T(s)
FPLRP_20	20	65-95	20-29	239.84	241.78	82.2	240.13	242.74	68.9	242.64	244.59	64.5
FPLRP_40	40	122-167	37-51	395.95	400.25	384.2	400.49	403.34	259.4	404.21	407.46	269.7
FPLRP_60	60	207-249	63-75	532.51	540.31	1054.0	541.45	550.53	648.5	549.13	554.10	717.2
FPLRP_80	80	266-325	81-98	653.53	665.09	2238.9	665.01	677.83	1504.1	670.97	677.83	1563.6
Average				455.46	461.86	939.8	461.77	468.61	620.2	466.74	471.00	653.8

5.4 Managerial Insights

We now discuss a detailed analysis regarding parking- and customer-related flexibility. First, we analyze the impact of different cost components impacting the operations and their interactions. Second, we describe conditions in which considering the walking mode is important to reduce the total operational costs and help decrease the environmental footprint. Third, we quantify the cost savings resulted from each type of customer-related flexibility. All the tests are performed on the instances with 60 customers. Sections 5.4.1 and 5.4.2 present the results from parking- and customer-related flexibility. We measured three metrics, i.e., total costs, number of customers served in a particular method (walking mode, driving mode, or being assigned to lockers), driving, and walking times. The numbers of customers served on foot, by driving mode, and being assigned to lockers are denoted by "#DCust", "#WCust", and "#LCust", respectively.

5.4.1 Parking-related flexibility

Two types of parking-related parameters are varied, namely the number of available parking spots and the associated time windows, reflecting various flexibility levels for parking. For the number of available parking spots, we generated four scenarios by reducing or increasing their number from the original values. Specifically, scenarios R20 and R40 consist of instances with a 20% and 40% reduction in available parking spots, respectively, while scenarios A20 and A40 denote instances with a 20% and 40% increase. For their time windows, we generated two scenarios, i.e., instances in which each parking spot is available throughout the entire planning period (NTW), and instances in which a single time window exists in each parking spot (STW). For the latter, we randomly selected one of the existing time windows from the original instances.

Figures 3 and 4 show the results of varying the number of available parking spots. The total costs is increased by 4.33% (1.62%) in R40 (R20), respectively, and decreased by 1.22% (0.06%) in A40 (A20), respectively, as shown in Figure 3. A relatively significant cost increase occurs when the number of available parking spots decreases. However, the cost does not decrease significantly when the number of available parking spots increases, indicating that it is unnecessary to have a large number of available parking spots to achieve operational efficiency. We also observed that the number of customers assigned to lockers increased by 10.40% (6.66%) in scenario R40 (R20) and decreased by 13.73% (1.14%) in scenario A40 (A20). This result indicates that lockers become more attractive as the number of available parking spots decreases. Figure 4 presents the main cause of cost changes. We recall that total driving times and total walking times are part of the total costs. As shown in Figure 4, the driving time is increased by 11.92% while the walking time is decreased by 29.39% when the number of parking spots is at the lowest level. Indeed, the delivery plans include a higher number of locations visited in the driving mode when the number of parking spots is low, as serving customers on foot is more costly (or infeasible due to the lack of relatively close parking spots and the presence of time windows).

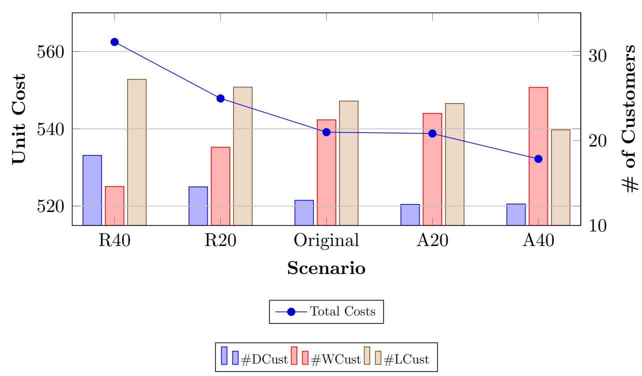


Figure 3: The impact of varying the number of available parking spots toward total costs and the method utilized to serve customers

Figures 5 and 6 show the results of varying time windows of parking spots. As expected, the total costs increase as the time availability of parking spots increase. The total costs increased (decreased) by 4.37% (3.89%) in scenario STW (NTW). In scenario STW, the highest proportion of customers is assigned to lockers, i.e., 16.17% higher than the original scenario, again showing the usefulness of considering lockers as alternative delivery locations when the flexibility of parking spots is at low level.

In summary, the flexibility of parking spots significantly influences total costs and the operations. To achieve an efficient delivery plan, it is essential to have a sufficient number of parking spots with the largest possible time availability. Lockers utilization becomes a more promising alternative in scenarios with a low flexibility parking spots. Regarding the

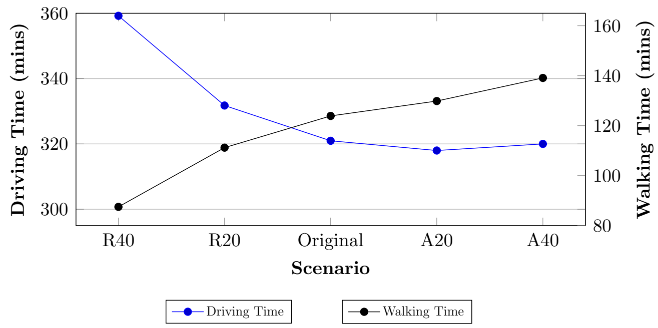


Figure 4: The impact of varying the number of available parking spots toward driving and walking times

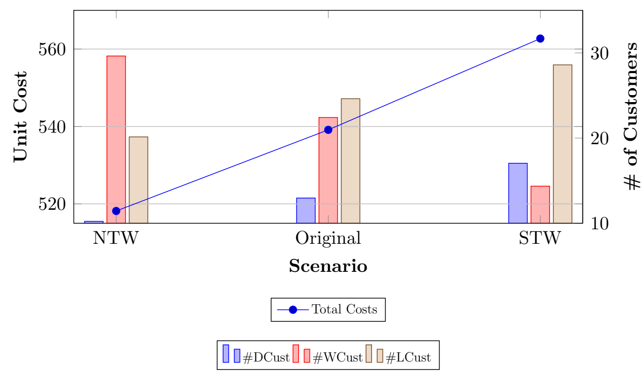


Figure 5: The impact of varying the time windows of parking spots toward total costs and the method utilized to serve customers

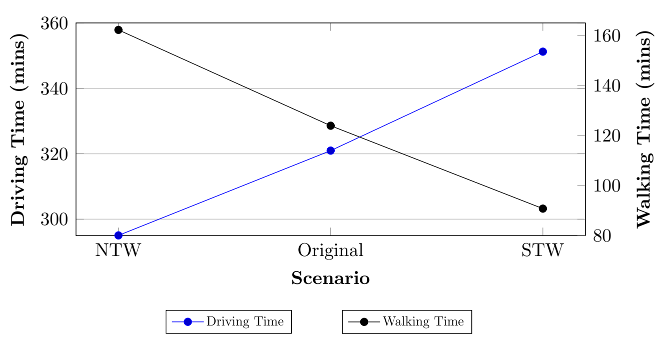


Figure 6: The impact of varying the time windows of parking spots toward driving and walking times

availability of parking spots, an interesting strategy to consider is horizontal collaboration among e-commerce logistics providers. We recall the booking system in Mor et al. [2020], in which all providers book parking spots in sequence. Therefore, providers who book at the beginning of the sequence may gain advantages from the high flexibility level of parking spots (scenario NTW), while providers who book last may experience the low flexibility level of parking spots (scenario STW). As shown in Figures 5 and 6, the least cost delivery plan results from the scenarios in which each parking spot is available all the time. By collaborations, e.g., allowing a provider to deliver packages for customers from other providers, they may achieve more efficient plans.

5.4.2 Customer-related flexibility

Since the flexibility of customers may vary over planning periods, we are also interested in their types of flexibility. For the number of flexible customers, the original proportion is 50%, and we vary the parameters by creating other instances from 30% (FC30) to 70% (FC70). For their time windows, the original instances are modified by reducing and increasing the width of acceptable time windows. We recall δ_{acc}, a parameter used to create $\hat{a}_i$ and $\hat{b}_i$ of each location $i \in N$. We created five instances for each original instance by increasing and decreasing δ_{acc} by 25% and 50%. The scenarios with increased (decreased) δ_{acc} are represented by A25 and A50 (R25 and R50). The last variant is generated by setting δ_{acc} to 0. Lastly, the number of roaming delivery locations is also varied by increasing and reducing it by 1 or 2. The scenarios with increased (decreased) number of roaming delivery locations are represented by A1 and A2 (R1 and R2). We apply boundaries to this rule, i.e., for each customer who has 5 roaming delivery locations, the number of locations is not further increased. Similarly, for each customer who only has 1 roaming delivery location (i.e., only the home location), no further reduction is applied. We note that Figures 7–12 are produced with the same scales in order to enable comparison among the three types of customer-related flexibility.

Based on Figures 7 and 8, the total costs are increased and decreased by 3.04% and 5.93% when the number of flexible customers changes. In scenarios FC30 and FC40, the proportion of customers who prefer their packages to be sent to one of the roaming delivery locations is higher. Thus, the number of customers whose packages are assigned to lockers is reduced by 44.60% and 85.13% in scenarios FC40 and FC30, respectively. Due to the lower cost resulted by serving customers on foot, the increase of customers served on foot (40.86% and 75.20% in scenarios FC40 and FC30, respectively) is higher than that of customers served by the driving mode (14.04% and 31.64% in scenarios FC40 and FC30, respectively). The opposite occurs in scenarios FC60 and FC70. By increasing the number of flexible customers (scenarios FC60 and FC70), the number of customers assigned to lockers is increased by 37% and 68%. We draw two observations from these results: (1) the importance of a consolidation strategy at lockers to achieve operational efficiency grows as the number of flexible customers increases, and (2) less benefits are resulted from serving customers on foot. The latter can be seen in scenario FC70, where the number of customers served on foot is the lowest. Since most of customers are assigned to lockers, the number of customers served at their roaming delivery locations is reduced. Hence, the number of customers served from a visited parking spot also reduces to the point where serving a customer on driving mode is more efficient.

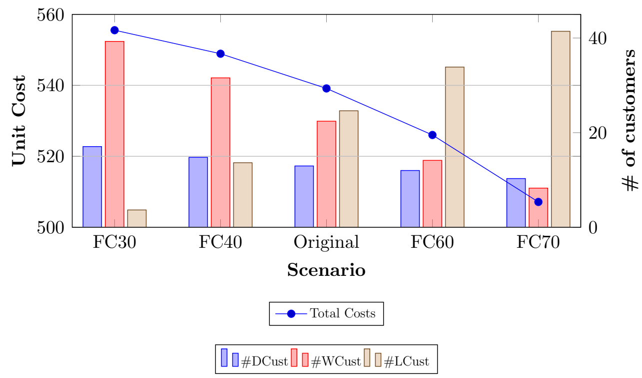


Figure 7: The impact of varying the number of flexible customers toward total costs and the method utilized to serve customers

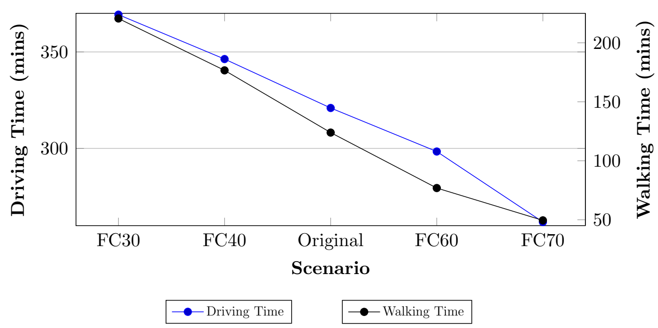


Figure 8: The impact of varying the number of flexible customers toward driving and walking times

A slight change in total costs results from the change in acceptable time windows of customers, as shown from Figures 9 and 10. Wider acceptable time windows do not necessarily translate into lower operational costs as other types of flexibility contribute to the operational efficiency, as shown by scenarios A25 and A50. As illustrated in scenarios Original, A25, and A50, the total costs remain at a similar level. In the extreme case where time window flexibility is entirely removed (scenario NFTW), the total costs increase by 0.60%, as illustrated in Figure 9. The cost increase occurs because the number of customers served on foot decreases by 5.62% and the number of customers served by driving mode increases by 7.25%. Consequently, the driving time increases while the walking time decreases, as shown in Figure 10. This phenomenon is reasonable because tighter time windows increase the number of situations where the driving mode is the only option for ensuring on-time deliveries.

When the number of roaming delivery locations of a customer is increased, each location has a tighter time window. Thus, analyzing the varying number of roaming delivery

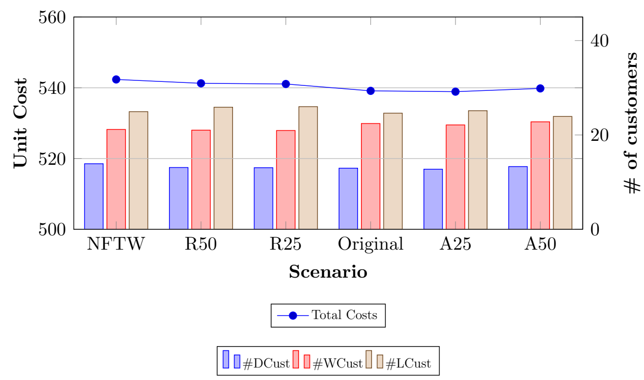


Figure 9: The impact of varying the time windows of customers toward total costs and the method utilized to serve customers

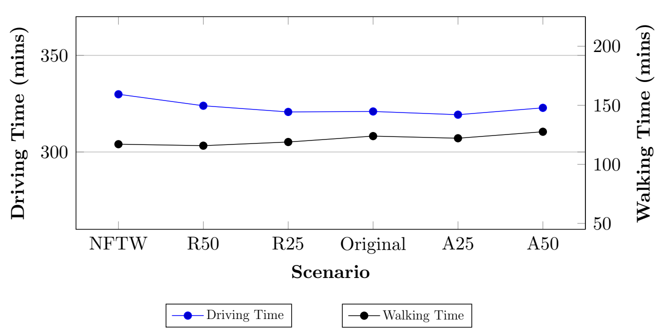


Figure 10: The impact of varying the time windows of customers toward driving and walking times

locations results in two types of analysis, i.e., the density of potential delivery locations and the width of time windows at those locations. As shown in Figure 11, the total costs in each generated scenario are lower than that of the original scenario. When the number of roaming delivery locations is reduced, the density of delivery locations decreases. However, the time window at each location, on average, becomes wider. Conversely, the density of delivery locations increases in scenarios A1 and A2. In all these four scenarios, more customers can be served on foot (as shown in Figures 11 and 12) due to the loose time windows (scenarios R1 and R2) or more customer locations that can be visited once a driver arrives at a parking spot (scenarios A1 and A2). Since increasing the number of roaming delivery locations results in more locations to be selected, one may expect the cost savings from scenarios A1 and A2 (2.54% and 3.42%, respectively) to be higher on average compared to those of scenarios R1 and R2 (0.11% and 1.98%, respectively).

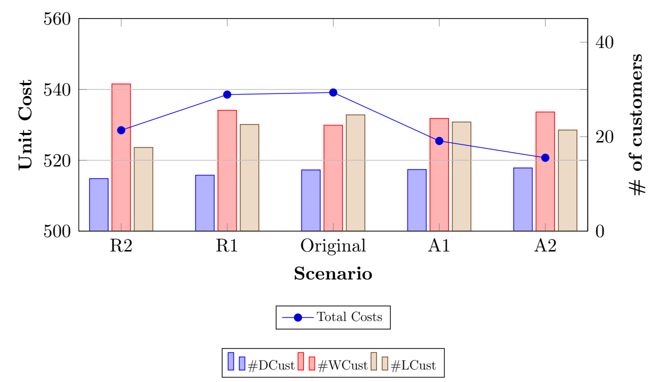


Figure 11: The impact of varying the number of roaming delivery locations toward total costs and the methods utilized to serve customers

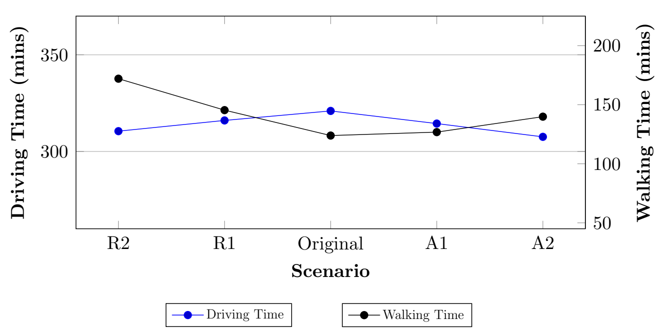


Figure 12: The impact of varying the number of roaming delivery locations toward driving and walking times

In summary, all types of flexibility result in different magnitudes of cost savings. Based on our sensitivity analyses, space-based customer flexibility, i.e., lockers and roaming delivery locations, results in higher cost savings compared to time-based customer flexibility. The sensitivity analysis provides valuable information for decision-makers to prioritize which strategy to develop in order to enhance their customers' flexibility. For instance, one may choose to first implement delivery to lockers as the utilization of lockers leads to the highest cost saving among all considered types of flexibility. The savings in terms of driving and walking times experienced by drivers from utilizing lockers can be exploited to further improve operational efficiency.

6 Conclusion

In this work, we have investigated a flexible park-and-loop routing problem (FPLRP) that integrates two types of flexibility stemming from parking spots and customers. This problem addresses the research gap in previous literature where parking spots were assumed to be available throughout the planning period, and various types of customer flexibility were exploited to achieve the highest operational efficiency.

We proposed a MILP, obtaining optimal solutions for instances with up to 10 customers and feasible for instances with 15 customers. Our proposed Hybrid Large Neighborhood Search (HLNS) obtains solutions that are equal or better in only a fraction of the time of the solver. The proposed HLNS also obtained comparable results for two benchmark problems, i.e., the vehicle routing with roaming delivery locations and the park-and-loop routing problem. In particular, we improved 37 best-known solutions for the large instances of the park-and-loop routing problem from the literature. We have also presented a sensitivity analysis regarding parking- and customer-related flexibility, showing the importance of parking-related flexibility and the significance of space-based customer flexibility over time-based customer flexibility in achieving cost savings.

While our problem comprehensively integrates a wide range of flexibility into the park-and-loop routing problem, various future research directions can still be explored. First, detailed extensions of space-based customer flexibility can be considered, such as time-dependent capacity, different capacities for each package size, and data-driven compensation schemes. Furthermore, studying various policies to address the dynamic version of this problem can also be an interesting direction, as a large number of e-commerce companies offer same-day delivery. Lastly, explicitly considering the environmental impact of utilizing walking mode in addition to driving mode can be beneficial for wider acceptance of this delivery concept.

References

Claudia Archetti, Ola Jabali, and M Grazia Speranza. Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134, 2015.

Claudia Archetti, Francesca Guerriero, and Giusy Macrina. The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, 127:105144, 2021.

Yasemin Arda, Diego Cattaruzza, Véronique François, and Maxime Ogier. Home chemotherapy delivery: An integrated production scheduling and multi-trip vehicle routing problem. *European Journal of Operational Research*, 317(2):468–486, 2024.

Nils Boysen, Stefan Fedtke, and Stefan Schwerdfeger. Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1):1–58, 2021.

Nicolás Cabrera, Jean-François Cordeau, and Jorge E Mendoza. The doubly open park-and-loop routing problem. *Computers & Operations Research*, 143:105761, 2022.

Nicolás Cabrera, Jean-François Cordeau, and Jorge E Mendoza. Solving the park-and-loop routing problem by branch-price-and-cut. *Transportation Research Part C: Emerging Technologies*, 157:104369, 2023.

Jan Christiaens and Greet Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020.

Marc-Antoine Coindreau, Olivier Gallay, and Nicolas Zufferey. Vehicle routing with transportable resources: Using carpooling and walking for on-site services. *European Journal of Operational Research*, 279(3):996–1010, 2019.

Maryam Darvish, Claudia Archetti, Leandro C Coelho, and M Grazia Speranza. Flexible two-echelon location routing problem. *European Journal of Operational Research*, 277 (3):1124–1136, 2019.

Alina G Dragomir, Tom Van Woensel, and Karl F Doerner. The pickup and delivery problem with alternative locations and overlapping time windows. *Computers & Operations Research*, 143:105758, 2022.

Yvan Dumas, François Soumis, and Jacques Desrosiers. Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science*, 24(2): 145–152, 1990.

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

Dorian Dumez, Christian Tilk, Stefan Irnich, Fabien Lehuédé, and Olivier Péton. Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows. *EURO Journal on Transportation and Logistics*, 10: 100040, 2021b.

David LJU Enthoven, Bolor Jargalsaikhan, Kees Jan Roodbergen, Michiel AJ Uit het Broek, and Albert H Schrotenboer. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. *Computers & Operations Research*, 118:104919, 2020.

Martin Joerss, Florian Neuhaus, and Jürgen Schröder. How customer demands are reshaping last-mile delivery. *The McKinsey Quarterly*, 17:1–5, 2016.

Théo Le Colleter, Dorian Dumez, Fabien Lehuédé, and Olivier Péton. Small and large neighborhood search for the park-and-loop routing problem with parking selection. *European Journal of Operational Research*, 308(3):1233–1248, 2023.

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

Antonio Martinez-Sykora, Fraser McLeod, Carlos Lamas-Fernandez, Tolga Bektaş, Tom Cherrett, and Julian Allen. Optimised solutions to the last-mile delivery problem in London using a combination of walking and driving. *Annals of Operations Research*, 295:645–693, 2020.

A Mor, MG Speranza, and JM Viegas. Efficient loading and unloading operations via a booking system. *Transportation Research Part E: Logistics and Transportation Review*, 141:102040, 2020.

Thu Ba T Nguyên, Tolga Bektaş, Tom J Cherrett, Fraser N McLeod, Julian Allen, Oliver Bates, Marzena Piotrowska, Maja Piecyk, Adrian Friday, and Sarah Wise. Optimising parcel deliveries in London using dual-mode routing. *Journal of the Operational Research Society*, 70(6):998–1010, 2019.

Gizem Ozbaygin, Oya Ekin Karasan, Martin Savelsbergh, and Hande Yaman. A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 100:115–137, 2017.

Sophie N Parragh and Jean-François Cordeau. Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers & Operations Research*, 83:28–44, 2017.

Quang Anh Pham, Minh Hoàng Hà, Duy Manh Vu, and Huy Hoang Nguyen. A hybrid genetic algorithm for the vehicle routing problem with roaming delivery locations. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 297–306, 2022.

David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

Sara Reed, Ann Melissa Campbell, and Barrett W Thomas. Does parking matter? the impact of parking time on last-mile delivery optimization. *Transportation Research Part E: Logistics and Transportation Review*, 181:103391, 2024.

Damián Reyes, Martin Savelsbergh, and Alejandro Toriello. Vehicle routing with roaming delivery locations. *Transportation Research Part C: Emerging Technologies*, 80:71–91, 2017.

Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4): 455–472, 2006.

Maximilian Schiffer and Grit Walther. An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52(2):331–352, 2018.

Fernando Senna, Leandro C Coelho, Reinaldo Morabito, and Pedro Munari. An exact method for a last-mile delivery routing problem with multiple deliverymen. *European Journal of Operational Research*, 317(2):550–562, 2024.

Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 417–431. Springer, 1998.

Statista. Retail e-commerce revenue in canada from 2017 to 2028 (in billion u.s. dollars). `https://www.statista.com/statistics/289741/canada-retail-e-commerce-sales/`, December 2023. Accessed: 2024-03-13.

Duygu Taş, Ola Jabali, and Tom Van Woensel. A vehicle routing problem with flexible time windows. *Computers & Operations Research*, 52:39–54, 2014.

Renata Turkeš, Kenneth Sörensen, and Lars Magnus Hvattum. Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search. *European Journal of Operational Research*, 292(2):423–442, 2021.

Yuan Yuan, Diego Cattaruzza, Maxime Ogier, Frédéric Semet, and Daniele Vigo. A column generation based heuristic for the generalized vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 152:102391, 2021.

Lele Zhang, Pengyuan Ding, and Russell G Thompson. A stochastic formulation of the two-echelon vehicle routing and loading bay reservation problem. *Transportation Research Part E: Logistics and Transportation Review*, 177:103252, 2023.

Lin Zhou, Roberto Baldacci, Daniele Vigo, and Xu Wang. 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, 2018.

A Mixed-Integer Linear Program for FPLRP

$$\textbf{(M1)}\ \text{Minimize Z} = c^d \sum_{k\in K} \sum_{(i,j)\in A^d} t^d_{ij} x_{ijk} + c^w \sum_{k\in K} \sum_{p\in P} \sum_{(i,j)\in A^w} \sum_{o\in W} t^w_{ij} y_{ijpok} +$$

Table 8: Description of the sets

Notation	Description
$\{0, 0'\}$	Origin and destination depots
C	Set of customers
N_i	Set of roaming delivery locations for customer $i \in C$
N	Set of roaming delivery locations, $N = \cup_{i \in C} N_i$
P	Set of parking spots
L	Set of lockers
K	Set of vehicles
H_i	Set of time windows at parking spot $i \in P$
A^d	Set of driving arcs, where $A^d = \{(i,j) \vert \forall i \in 0 \cup N \cup L \cup P, \forall j \in N \cup L \cup P \cup 0', i \neq j\}$
A^w	Set of walking arcs, where $A^w = \{(p,i) \vert \forall p \in P, i \in N\} \cup \{(i,j) \vert \forall i,j \in N, i \neq j\} \cup \{(j,p) \vert \forall j \in N, p \in P\}$
W	Set of walking subtours originating from a parking spot, $\vert W \vert \leq \vert C \vert$

Table 9: Description of the parameters

Parameters	Description
t^d_{ij}	Driving time from node i to node j, where $\forall (i,j) \in A^d$
t^w_{ij}	Walking time from node i to node j, where $\forall (i,j) \in A^w$
d_{ij}	Distance from node i to node j, where $\forall (i,j) \in A$
g_i	Parking time at location $i \in N \cup P \cup L$
q_i	Demand of customer $i \in C$
Q^d	Vehicle capacity
Q^w	Walking capacity
r^{max}	Maximum walking distance of a driver
$[a_i, b_i]$	Preferable time windows of location $i \in N$
$[\hat{a}_i, \hat{b}_i]$	Acceptable time windows of location $i \in N$
$[\tilde{a}_{ih}, \tilde{b}_{ih}]$	The h-th time window at parking spot $i \in P$ where $h \in H_i$
$[0, T]$	Planning period
m_{ij}	A binary parameter representing the selected locker of customer $i \in C$, i.e., 1 if locker $j \in L$ is selected by customer i, 0 otherwise
c^d	Weight for total driving time
c^w	Weight for total walking time
$\tilde{\beta}_{ij}$	Compensation given to customer $i \in C$ who is assigned for picking up their package at locker $j \in L$
β'_i	Penalty per one unit of earliness experienced by customer $i \in C$
β''_i	Penalty per one unit of delay experienced by customer $i \in C$

$$\tilde{\beta}_{ij} \sum_{k \in K} \sum_{j \in L} \sum_{i \in C} z_{ijk} + \sum_{i \in C} \beta'_i \rho'_i + \sum_{i \in C} \beta''_i \rho''_i \tag{10}$$

Subject to

$$\sum_{k \in K} \sum_{\substack{(i,j) \in A^w \\ \vert j \in N_l}} \sum_{p \in P} \sum_{o \in W} y_{ijpok} + \sum_{k \in K} \sum_{\substack{(i,j) \in A^d \\ \vert j \in N_l}} x_{ijk} = 1, \forall l \in C^R \tag{11}$$

Table 10: Description of the decision variables

Decision variables	Description
x_{ijk}	Binary variable, i.e., 1 if arc $(i,j) \in A^d$ is traversed by vehicle $k \in K$
y_{ijpok}	Binary variable, i.e., 1 if arc $(i,j) \in A^w$ is traversed by driver $k \in K$ in the o-th walking subtour while the associated vehicle is parked at parking spot $p \in P$
z_{ijk}	Binary variable indicating the demand of customer $i \in C$ is delivered by driver $k \in K$ to locker $j \in L$
u_i	Non-negative integer variable indicating the remaining walking capacity of a driver after considering demand of customer $i \in C$
τ'_i	Non-negative continuous variable indicating the arrival time of a vehicle at location $i \in C \cup P \cup L \cup 0'$
τ''_i	Non-negative continuous variable indicating the departure time of a vehicle from location $i \in 0 \cup C \cup P \cup L$
w_i	Non-negative continuous variable indicating the arrival time of a driver at any location of customer $i \in C$ by walking mode
v_{po}	Non-negative continuous variable indicating the arrival time of a driver at parking spot $p \in P$ after completing o-th walking subtour
ρ'_i	Non-negative continuous variable indicating total earliness on serving customer $i \in C$
ρ''_i	Non-negative continuous variable indicating total delay on serving customer $i \in C$
s_{ph}	Binary variable indicating the selection of time windows $h \in H_p$ at parking spot $p \in P$

$$\sum_{k \in K} \sum_{\substack{(i,j) \in A^w \\ |j \in N_l}} \sum_{p \in P} \sum_{o \in W} y_{ijpok} + \sum_{k \in K} \sum_{\substack{(i,j) \in A^d \\ |j \in N_l}} x_{ijk} + \sum_{k \in K} \sum_{\substack{j \in L \\ |m_{lj}=1}} z_{ljk} = 1, \forall l \in C^F \tag{12}$$

$$z_{ljk} \leq \sum_{(i,j) \in A^d} x_{ijk}, \forall k \in K, l \in C, j \in L \tag{13}$$

$$\sum_{(i,j) \in A^d} x_{ijk} = \sum_{(j,i) \in A^d} x_{jik}, \forall k \in K, j \in N \cup P \cup L \tag{14}$$

$$\sum_{(0,j) \in A^d} x_{ijk} = \sum_{(j,0') \in A^d} x_{jik} = 1, \forall k \in K \tag{15}$$

$$\sum_{k \in K} \sum_{(p,j) \in A^d} x_{pjk} \leq 1, \forall p \in P \tag{16}$$

$$\sum_{(p,j) \in A^w} y_{pjpok} \leq 1, \forall k \in K, p \in P, o \in W \tag{17}$$

$$\sum_{(i,j) \in A^w} y_{ijpok} = \sum_{(j,i) \in A^w} y_{jipok}, \forall k \in K, j \in N \cup P, p \in P, o \in W \tag{18}$$

$$\sum_{\substack{(i,j) \in A^w \\ |i \in P, i \neq p}} y_{ijpok} = \sum_{\substack{(j,i) \in A^w \\ |i \in P, i \neq p}} y_{jipok} = 0, \forall k \in K, p \in P, o \in W \tag{19}$$

$$\sum_{(p,j)\in A^w} y_{pjpok} \leq \sum_{(j,p)\in A^w} y_{jppo-1k}, \forall k \in K, p \in P, o \in \{2,\dots,|W|\} \tag{20}$$

$$\sum_{(p,j)\in A^w} \sum_{o\in W} y_{pjpok} \leq |W| \sum_{(i,p)\in A^d} x_{ipk}, \forall k \in K, p \in P \tag{21}$$

$$\sum_{(i,j)\in A^w} \sum_{p\in P} \sum_{o\in W} d_{ij} y_{ijpok} \leq r^{max}, \forall k \in K \tag{22}$$

$$u_l \leq Q^w - q_l \sum_{k\in K} \sum_{i\in N_l} \sum_{p\in P} \sum_{o\in W} y_{pipok}, \forall l \in C \tag{23}$$

$$u_{\hat{l}} \leq u_l - q_{\hat{l}} + Q^w \left(1 - \sum_{k\in K} \sum_{i\in N_l} \sum_{j\in N_{\hat{l}}} \sum_{p\in P} \sum_{o\in W} y_{ijpok}\right), \forall l, \hat{l} \in C, l \neq \hat{l} \tag{24}$$

$$\sum_{l\in C} q_l \left(\sum_{\substack{(i,j)\in A^d \\ |j\in N_l}} x_{ijk} + \sum_{\substack{(i,j)\in A^w \\ |j\in N_l}} \sum_{p\in P} \sum_{o\in W} y_{ijpok} + \sum_{j\in L} z_{ljk} \right) \leq Q^d, \forall k \in K \tag{25}$$

$$\tau''_i \geq \tau'_i, \forall i \in C \cup L \cup P \tag{26}$$

$$\tau'_j \geq \tau''_i + t^d_{ij} - M\left(1 - \sum_{k\in K} x_{ijk}\right), \forall i \in 0 \cup L \cup P, j \in L \cup P \cup 0', i \neq j \tag{27}$$

$$\tau'_l \geq \tau''_i + \sum_{j\in N_l} t^d_{ij} \sum_{k\in K} x_{ijk} - M\left(1 - \sum_{k\in K} \sum_{j\in N_l} x_{ijk}\right), \forall l \in C, i \in 0 \cup L \cup P \tag{28}$$

$$\tau'_j \geq \tau''_l + \sum_{i\in N_l} t^d_{ij} \sum_{k\in K} x_{ijk} - M\left(1 - \sum_{k\in K} \sum_{i\in N_l} x_{ijk}\right), \forall l \in C, j \in L \cup P \cup 0' \tag{29}$$

$$\tau'_{\hat{l}} \geq \tau''_l + \sum_{i\in N_l} \sum_{j\in N_{\hat{l}}} t^d_{ij} \sum_{k\in K} x_{ijk} - M\left(1 - \sum_{k\in K} \sum_{i\in N_l} \sum_{j\in N_{\hat{l}}} x_{ijk}\right), \forall l \in C, \hat{l} \in C, l \neq \hat{l} \tag{30}$$

$$w_l \geq \tau'_p + \sum_{i\in N_l} t^w_{pi} \sum_{k\in K} y_{pip1k} - M\left(1 - \sum_{k\in K} \sum_{i\in N_l} y_{pip1k}\right), \forall l \in C, p \in P \tag{31}$$

$$w_l \geq v_{po-1} + \sum_{i\in N_l} t^w_{pi} \sum_{k\in K} y_{pipok} - M\left(1 - \sum_{k\in K} \sum_{i\in N_l} y_{pipok}\right), \forall l \in C, p \in P,$$
$$o \in \{2,\dots,|W|\} \tag{32}$$

$$w_{\hat{l}} \geq w_l + \sum_{i\in N_l} \sum_{j\in N_{\hat{l}}} \sum_{p\in P} \sum_{o\in W} t^w_{ij} \sum_{k\in K} y_{ijpok} - M\left(1 - \sum_{k\in K} \sum_{i\in N_l} \sum_{j\in N_{\hat{l}}} \sum_{p\in P} \sum_{o\in W} y_{ijpok}\right),$$
$$\forall l, \hat{l} \in C, l \neq \hat{l} \tag{33}$$

$$v_{po} \geq w_l + \sum_{i \in N_l} t^w_{ip} \sum_{k \in K} y_{ippok} - M\left(1 - \sum_{k \in K}\sum_{i \in N_l} y_{ippo}\right), \forall l \in C, p \in P, o \in W \tag{34}$$

$$v_{po} \leq M \sum_{k \in K} \sum_{(p,j) \in A^w} y_{pjpok}, \forall p \in P, o \in W \tag{35}$$

$$v_{po} \geq v_{po-1} - M\left(1 - \sum_{k \in K} \sum_{(p,j) \in A^w} y_{pjpok}\right), \forall p \in P, o \in \{2, \ldots, |W|\} \tag{36}$$

$$\tau''_p \geq v_{po}, \forall p \in P, o \in W \tag{37}$$

$$\sum_{i \in N_l} \hat{a}_i \sum_{k \in K} \sum_{(i,j) \in A^d} x_{ijk} \leq \tau'_l \leq \sum_{i \in N_l} \hat{b}_i \sum_{k \in K} \sum_{(i,j) \in A^d} x_{ijk}, \forall l \in C \tag{38}$$

$$\sum_{i \in N_l} \hat{a}_i \sum_{k \in K} \sum_{(i,j) \in A^w} \sum_{p \in P} \sum_{o \in W} y_{ijpok} \leq w_l \leq \sum_{i \in N_l} \hat{b}_i \sum_{k \in K} \sum_{(i,j) \in A^w} \sum_{p \in P} \sum_{o \in W} y_{ijpok}, \forall l \in C \tag{39}$$

$$\rho'_l \geq \sum_{i \in N_l} a_i \sum_{k \in K} \sum_{(i,j) \in A^d} x_{ijk} - \tau'_l, \forall l \in C \tag{40}$$

$$\rho'_l \geq \sum_{i \in N_l} a_i \sum_{k \in K} \sum_{(i,j) \in A^w} \sum_{p \in P} \sum_{o \in W} y_{ijpok} - w_l, \forall l \in C \tag{41}$$

$$\rho''_l \geq \tau'_l - \sum_{i \in N_l} b_i \sum_{k \in K} \sum_{(i,j) \in A^d} x_{ijk}, \forall l \in C \tag{42}$$

$$\rho''_l \geq w_l - \sum_{i \in N_l} b_i \sum_{k \in K} \sum_{(i,j) \in A^w} \sum_{p \in P} \sum_{o \in W} y_{ijpok}, \forall l \in C \tag{43}$$

$$\sum_{h \in H_p} s_{ph} = \sum_{k \in K} \sum_{(i,p) \in A^d} x_{ipk}, p \in P \tag{44}$$

$$\tau'_p \geq \tilde{a}_{ph} s_{ph}, \forall h \in H_p, p \in P \tag{45}$$

$$\tau''_p \leq \tilde{b}_{ph} s_{ph} + M(1 - s_{ph}), \forall h \in H_p, p \in P \tag{46}$$

$$y_{ijpok} \in \{0,1\}, \forall k \in K, (i,j) \in A^w, p \in P, o \in W \tag{47}$$

$$x_{ijk} \in \{0,1\}, \forall k \in K, (i,j) \in A^d \tag{48}$$

$$s_{ph} \in \{0,1\}, \forall h \in H_p, p \in P \tag{49}$$

$$z_{ijk} \in \{0,1\}, \forall k \in K, i \in C, j \in L \tag{50}$$

$$u_i, w_i, \rho'_i, \rho''_i \geq 0, \forall i \in C \tag{51}$$

$$\tau'_i \geq 0, \forall i \in C \cup P \cup L \cup 0' \tag{52}$$

$$\tau''_i \geq 0, \forall i \in 0 \cup C \cup P \cup L \tag{53}$$

$$v_{po} \geq 0, \forall p \in P, o \in W \tag{54}$$

The objective function (10) minimizes the total operational costs. Constraints (11) and (12) ensure that each customer is served. Constraints (13) ensure that a locker is visited if it receives the package of a customer. Constraints (14) state the flow conservation when a vehicle is in the driving mode. Constraints (15) ensure that each vehicle starts and ends at the depot. Constraints (16) and (17) guarantee that each parking spot is visited at most once. Constraints (18)-(22) define the characteristics of the walking mode. In particular, constraints (18) ensure the flow conservation of a driver while walking. Constraints (19) forbid flows originating from different parking spots. Constraints (20) guarantee the precedence relationship among walking subtours of a parking spot. Constraints (21) limit the maximum number of walking subtours that can originate from a parking spot and constraints (22) limit the maximum distance that can be travelled by each driver on foot.

Constraints (23)-(25) limit capacity (both in walking and driving modes) of each driver: (23) and (24) consecutively track the remaining walking capacity upon the first location and following locations visited by a driver on foot from a parking spot. Constraints (25) limit the vehicle capacity.

Constraints (26)-(37) update the arrival and departure times of drivers in both driving and walking modes. Constraints (26) ensure the precedence relationship between the departure and the arrival time variables of a location. Constraints (27)-(30) update the arrival time at a location based on the location previously visited by a vehicle. Constraints (31)-(36) handle the arrival and departure times of drivers in the walking mode. In particular, constraints (31) ensure the time synchronization from driving mode into walking mode. Constraints (32) guarantee the arrival time at the first customer location visited by a driver who has just completed a walking subtour and started a subsequent walking subtour. Constraints (33) ensure the arrival time connectivity between customers visited in a walking mode by a driver. Constraints (34) track the arrival time at a parking spot after completing a walking subtour. Constraints (35) and (36) limit the value of arrival time at a parking spot and define the precedence relationship between two consecutive walking subtours, respectively. Constraints (37) ensure time synchronization when a vehicle departs from a parking spot.

Constraints (38) and (39) ensure that the arrival time of a driver at a customer's location is within the acceptable time windows. Constraints (40)-(43) define the compensation given to customers who receive their packages outside their preferable time windows. Constraints (44)-(46) ensure the time window selection at a visited parking spot and the entire parking activity must occur within the selected time window. Lastly, (47)-(54) define the ranges of the decision variables.

B Label extension functions for Dynamic Programming in Section 4.3

$$\mathcal{T}^{\text{cur}}(\theta') = j \tag{55}$$

$$\mathcal{T}^{\text{cost}}(\theta') = \mathcal{T}^{\text{cost}}(\theta) + \mathcal{T}^{\text{pen}}(\theta') + \begin{cases} c''t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + c't'_{\mathcal{T}^{\text{cPark}}(\theta)j}, & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w} \\ c't'_{ij}, & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d} \\ c''t''_{ij}, & \text{if } e_{ij} \in E2 \\ c''t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + c''t''_{\mathcal{T}^{\text{cPark}}(\theta)j}, & \text{if } e_{ij} \in E3 \\ c''t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + c't'_{\mathcal{T}^{\text{cPark}}(\theta)p} + c''t''_{pj}, & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{mode}(\theta) = \mathscr{w} \\ c't'_{ip} + c''t''_{pj}, & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{mode}(\theta) \in \mathscr{d} \end{cases} \tag{56}$$

$$\mathcal{T}^{\text{time}}(\theta') = \begin{cases} X_j(\theta, t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + t'_{\mathcal{T}^{\text{cPark}}(\theta)j}) & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w} \\ X_j(\theta, t'_{ij}) & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d} \\ X_j(\theta, t''_{ij}) & \text{if } e_{ij} \in E2 \\ X_j(\theta, t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + t''_{\mathcal{T}^{\text{cPark}}(\theta)j}) & \text{if } e_{ij} \in E3 \\ X_j(\theta, \max(t''_{i\mathcal{T}^{\text{cPark}}(\theta)} + t'_{\mathcal{T}^{\text{cPark}}(\theta)p}, \tilde{a}_{ph} - \mathcal{T}^{\text{time}}(\theta)) + t''_{pj}) & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w} \\ X_j(\theta, \max(t'_{ip}, \tilde{a}_{ph} - \mathcal{T}^{\text{time}}(\theta)) + t''_{pj}) & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d} \end{cases} \tag{57}$$

$$\mathcal{T}^{\text{vParks}}(\theta') = \mathcal{T}^{\text{vParks}}(\theta') \cup \begin{cases} \emptyset & \text{if } e_{ij} \in E1 \cup E2 \cup E3 \\ p & \text{if } e_{ij} \in E4 \end{cases} \tag{58}$$

$$\mathcal{T}^{\text{mode}}(\theta') = \begin{cases} \mathscr{d} & \text{if } e_{ij} \in E1 \\ \mathscr{w} & \text{otherwise} \end{cases} \tag{59}$$

$$\mathcal{T}^{\text{cPark}}(\theta') = \begin{cases} -1 & \text{if } e_{ij} \in E1 \\ \mathcal{T}^{\text{cPark}}(\theta) & \text{if } e_{ij} \in E1 \cup E2 \\ p & \text{if } e_{ij} \in E4 \end{cases} \tag{60}$$

$$\mathcal{T}^{\text{dist}}(\theta') = \mathcal{T}^{\text{dist}}(\theta) + \begin{cases} d_{i\mathcal{T}^{\text{cPark}}(\theta)} & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w} \\ 0 & \text{if } e_{ij} \in E1 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d} \\ d_{ij} & \text{if } e_{ij} \in E2 \\ d_{i\mathcal{T}^{\text{cPark}}(\theta)} + d_{\mathcal{T}^{\text{cPark}}(\theta)j} & \text{if } e_{ij} \in E3 \\ d_{i\mathcal{T}^{\text{cPark}}(\theta)} + d_{pj} & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{w} \\ d_{pj} & \text{if } e_{ij} \in E4 \wedge \mathcal{T}^{\text{mode}}(\theta) = \mathscr{d} \end{cases} \tag{61}$$

$$\mathcal{T}^{\text{wd}}(\theta') = \begin{cases} 0 & \text{if } e_{ij} \in E1 \\ \mathcal{T}^{\text{wd}}(\theta) + q_{N^{-1}(j)} & \text{if } e_{ij} \in E2 \\ q_{N^{-1}(j)} & \text{if } e_{ij} \in E3 \cup E4 \end{cases} \tag{62}$$

$$\mathcal{T}^{\text{ParkTW}}(\theta') = \begin{cases} -1 & \text{if } e_{ij} \in E1 \\ \mathcal{T}^{\text{ParkTW}}(\theta) & \text{if } e_{ij} \in E2 \cup E3 \\ h & \text{if } e_{ij} \in E4 \end{cases} \tag{63}$$

, where $X_j(\theta, t) = \max\{A_j, \mathcal{T}^{\text{time}}(\theta) + t\},\ A_j = 0, \forall j \in L; A_j = \hat{a}_j$ otherwise. The customer's demand associated with location j is represented by $q_{N^{-1}(j)}$. Lastly, $\mathcal{T}^{\text{pen}}(\theta')$ representing total penalty up to location j is calculated using the method in Dumas et al. [1990].

C Results for Section 5

Table 11: Detail results of parameter tuning

Notation	Values			
$(\eta^{\text{min}}, \eta^{\text{max}})$	(0.1, 0.3)	(0.1, 0.4)*	(0.2, 0.3)	(0.2, 0.4)
$\Delta(\%)$	0.19	0.00	0.55	0.26
$(\alpha^{\text{dp}}_{\text{init}}, \alpha^{\text{dp}}_{\text{min}})$	(1.05, 1.01)*	(1.05, 1.025)	(1.025, 1.01)	(1.025, 1.005)
$\Delta(\%)$	0.15	0.04	0.00	0.53
η^{dp}	5*	10	15	20
$\Delta(\%)$	0.00	0.44	0.08	0.21
ξ^{dp}	0.5*	0.6	0.7	0.8
$\Delta(\%)$	0.00	0.04	0.27	0.02
$(\alpha^{\text{acc}}_{\text{init}}, \alpha^{\text{acc}}_{\text{final}})$	(1.05, 1.001)	(1.025, 1.001)*	(1.05, 1.005)	(1.025, 1.005)
$\Delta(\%)$	0.86	0.00	5.41	0.44
η^{SP}	2000*	3000	4000	5000
$\Delta(\%)$	0.00	0.35	0.81	0.45
η^{iter}	15000	20000*	25000	30000
$\Delta(\%)$	0.08	0.00	0.23	0.12

$\Delta = \frac{(Obj_{\text{eval}} - Obj_{\text{min}}) * 100\%}{Obj_{\text{min}}}$ where Obj_{eval} and Obj_{min} represent the average of overall best objective values of the evaluated parameter(s) and the parameter(s) resulting in the lowest average of overall best objective values, respectively.

* denotes the initial values during the execution of parameter tuning.

Table 12: Detail results of HLNS on solving VRPRDL instances (set B1)

Instance	BKS	f^b	f^{ave}	CT(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
Instance_0	901	901	901	3.45	0.00	0.00
Instance_1	1286	1286	1286	3.27	0.00	0.00
Instance_2	991	991	991	2.76	0.00	0.00
Instance_3	1062	1062	1062	2.94	0.00	0.00
Instance_4	1832	1832	1832	2.89	0.00	0.00
Instance_5	1294	1294	1294	4.38	0.00	0.00
Instance_6	1155	1155	1155	3.84	0.00	0.00
Instance_7	1455	1455	1455	4.60	0.00	0.00
Instance_8	1260	1260	1260	5.01	0.00	0.00
Instance_9	1684	1684	1684	4.91	0.00	0.00
Instance_10	1922	1922	1922	8.16	0.00	0.00
Instance_11	2324	2324	2324	9.93	0.00	0.00
Instance_12	1747	1747	1747	10.00	0.00	0.00
Instance_13	1273	1273	1273	8.27	0.00	0.00
Instance_14	1694	1694	1694	10.43	0.00	0.00
Instance_15	1938	1938	1938	10.40	0.00	0.00
Instance_16	1965	1965	1965	12.53	0.00	0.00
Instance_17	1827	1827	1827	8.33	0.00	0.00
Instance_18	2083	2083	2083	7.60	0.00	0.00
Instance_19	1822	1822	1822	7.42	0.00	0.00
Instance_20	3761	3761	3761	31.91	0.00	0.00
Instance_21	2828	2828	2828	33.93	0.00	0.00
Instance_22	4440	4440	4440	31.10	0.00	0.00
Instance_23	3378	3378	3378	31.95	0.00	0.00
Instance_24	3161	3161	3161	98.60	0.00	0.00
Instance_25	4536	4536	4536	33.05	0.00	0.00
Instance_26	2865	2865	2865	26.17	0.00	0.00
Instance_27	4173	4173	4173	35.06	0.00	0.00
Instance_28	3964	3964	3964	37.88	0.00	0.00
Instance_29	4107	4107	4107	31.38	0.00	0.00
Instance_30	4935	4935	4935	134.82	0.00	0.00
Instance_31	5258	5258	5258	161.59	0.00	0.00
Instance_32	5061	5061	5061	176.42	0.00	0.00
Instance_33	5218	5218	5218	163.13	0.00	0.00
Instance_34	5498	5498	5498	150.02	0.00	0.00
Instance_35	6498	6498	6498	97.93	0.00	0.00
Instance_36	4830	4830	4830	137.04	0.00	0.00
Instance_37	5604	5604	5604	149.72	0.00	0.00
Instance_38	5841	5841	5841	120.54	0.00	0.00
Instance_39	4995	4995	4996.6	158.01	0.00	0.03
Average	3061.65	3061.65	3061.69	49.28	0.00	0.00

Table 13: Detail results of HLNS on solving VRPRDL instances (set B2)

Instance	BKS	f^b	f^{ave}	CT(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
Instance_0	773	773	773	2.94	0.00	0.00
Instance_1	1065	1065	1065	2.57	0.00	0.00
Instance_2	988	988	988	2.53	0.00	0.00
Instance_3	914	914	914	75.18	0.00	0.00
Instance_4	1710	1710	1710	3.17	0.00	0.00
Instance_5	1099	1099	1099	3.34	0.00	0.00
Instance_6	996	996	996	3.54	0.00	0.00
Instance_7	1346	1346	1346	69.07	0.00	0.00
Instance_8	997	997	997	148.28	0.00	0.00
Instance_9	1166	1166	1166	3.95	0.00	0.00
Instance_10	1587	1587	1587	6.62	0.00	0.00
Instance_11	1808	1808	1808	8.08	0.00	0.00
Instance_12	1563	1563	1563	8.46	0.00	0.00
Instance_13	1058	1058	1058	6.91	0.00	0.00
Instance_14	1347	1347	1347	8.59	0.00	0.00
Instance_15	1517	1517	1517	8.85	0.00	0.00
Instance_16	1445	1445	1445	8.75	0.00	0.00
Instance_17	1627	1627	1627	6.64	0.00	0.00
Instance_18	1461	1461	1461	6.60	0.00	0.00
Instance_19	1715	1715	1715	5.89	0.00	0.00
Instance_20	2580	2580	2580	25.57	0.00	0.00
Instance_21	2206	2206	2206	28.24	0.00	0.00
Instance_22	3363	3363	3363	202.26	0.00	0.00
Instance_23	2569	2569	2569	29.60	0.00	0.00
Instance_24	2378	2378	2378	34.58	0.00	0.00
Instance_25	2845	2845	2845	28.13	0.00	0.00
Instance_26	2518	2518	2518	22.84	0.00	0.00
Instance_27	2758	2758	2758	27.30	0.00	0.00
Instance_28	2892	2892	2892	31.52	0.00	0.00
Instance_29	2691	2691	2691	25.56	0.00	0.00
Instance_30	3666	3666	3666	107.47	0.00	0.00
Instance_31	3885	3885	3885	127.30	0.00	0.00
Instance_32	3543	3544	3544.6	163.45	0.03	0.05
Instance_33	3694	3694	3694.2	138.02	0.00	0.01
Instance_34	3127	3184	3184	206.91	1.82	1.82
Instance_35	4251	4273	4280.4	248.21	0.52	0.69
Instance_36	3217	3217	3217	104.36	0.00	0.00
Instance_37	3935	3935	3935.2	182.75	0.00	0.01
Instance_38	4300	4300	4300	113.20	0.00	0.00
Instance_39	3534	3555	3555.8	255.35	0.59	0.62
Average	2253.35	2255.88	2256.11	62.31	0.07	0.08

Table 14: Detail results of HLNS on solving VRPRDL instances (set B3)

Instance	BKS	f^b	f^{ave}	CT(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
41.v1	3203	3203	3203	17.47	0.00	0.00
42.v1	2799	2799	2799	16.47	0.00	0.00
43.v1	2603	2603	2603	24.65	0.00	0.00
44.v1	2261	2261	2261	13.97	0.00	0.00
45.v1	3217	3217	3217	15.40	0.00	0.00
46.v1	2805	2805	2805	19.04	0.00	0.00
47.v1	3339	3339	3339	18.39	0.00	0.00
48.v1	3325	3325	3325	16.23	0.00	0.00
49.v1	3534	3534	3534	15.44	0.00	0.00
50.v1	2752	2752	2752	18.52	0.00	0.00
41.v2	2133	2133	2133	17.54	0.00	0.00
42.v2	1946	1946	1946	17.45	0.00	0.00
43.v2	1966	1966	1966	18.48	0.00	0.00
44.v2	1610	1610	1610	14.76	0.00	0.00
45.v2	2478	2478	2478	14.90	0.00	0.00
46.v2	2469	2469	2469	19.43	0.00	0.00
47.v2	1946	1946	1946	18.53	0.00	0.00
48.v2	2380	2380	2380	17.72	0.00	0.00
49.v2	2492	2492	2492	19.79	0.00	0.00
50.v2	2443	2443	2443	18.55	0.00	0.00
Average	2585.05	2585.05	2585.05	17.64	0.00	0.00

Table 15: Detail results of HLNS on solving VRPRDL instances (set B4)

Instance	BKS	f^b	f^{ave}	CT(s)	$\text{Gap}^b(\%)$	$\text{Gap}^{\mathsf{Ave}}(\%)$
41.v1	2662	2662	2662	12.90	0.00	0.00
42.v1	2610	2610	2610	12.79	0.00	0.00
43.v1	2260	2260	2260	13.67	0.00	0.00
44.v1	2147	2147	2147	11.57	0.00	0.00
45.v1	3172	3172	3172	13.34	0.00	0.00
46.v1	2616	2616	2616	14.19	0.00	0.00
47.v1	3010	3010	3010	14.12	0.00	0.00
48.v1	3278	3278	3278	13.17	0.00	0.00
49.v1	3514	3514	3514	12.80	0.00	0.00
50.v1	2727	2727	2727	15.60	0.00	0.00
41.v2	1998	1998	1998	12.69	0.00	0.00
42.v2	1927	1927	1927	13.12	0.00	0.00
43.v2	1830	1830	1830	14.52	0.00	0.00
44.v2	1478	1478	1478	11.08	0.00	0.00
45.v2	2466	2466	2466	12.09	0.00	0.00
46.v2	2388	2388	2388	14.72	0.00	0.00
47.v2	1848	1848	1848	14.31	0.00	0.00
48.v2	2264	2264	2264	12.30	0.00	0.00
49.v2	2457	2457	2457	11.78	0.00	0.00
50.v2	2302	2302	2302	13.83	0.00	0.00
Average	2447.70	2447.70	2447.70	13.23	0.00	0.00

Table 16: Detail results of HLNS on solving PLRP small and medium instances

Instance	BKS	f^b	f^{ave}	CT(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
20_A_1	30.9482	30.9482	30.9482	2.85	0.00	0.00
20_A_2	41.5547	41.5547	41.5547	2.92	0.00	0.00
20_A_3	36.2344	36.2344	36.3177	2.63	0.00	0.23
20_A_4	36.0251	36.0251	36.0251	3.38	0.00	0.00
20_A_5	35.2747	35.2747	35.2747	3.24	0.00	0.00
20_A_6	42.2866	42.2866	42.2866	2.61	0.00	0.00
20_A_7	38.6881	38.6881	38.6881	2.54	0.00	0.00
20_A_8	36.8504	36.8504	36.8504	2.65	0.00	0.00
20_A_9	29.6105	29.6105	29.6105	3.61	0.00	0.00
20_A_10	39.6656	39.6656	39.7011	2.85	0.00	0.09
30_A_1	40.7372	40.7372	40.7372	6.05	0.00	0.00
30_A_2	45.6465	45.6465	45.6465	6.83	0.00	0.00
30_A_3	49.1828	49.1828	49.1828	5.44	0.00	0.00
30_A_4	43.7556	43.7556	43.7556	7.14	0.00	0.00
30_A_5	47.0619	47.0619	47.3287	5.98	0.00	0.57
30_A_6	49.5880	49.5880	49.5880	6.06	0.00	0.00
30_A_7	45.5340	45.5340	45.7269	5.86	0.00	0.42
30_A_8	43.6799	43.6799	43.6799	7.68	0.00	0.00
30_A_9	41.1609	41.1609	41.1609	6.99	0.00	0.00
30_A_10	46.8936	46.8936	46.8936	6.61	0.00	0.00
40_A_1	59.1695	59.1695	59.6793	7.99	0.00	0.86
40_A_2	58.1191	58.1191	58.1191	11.06	0.00	0.00
40_A_3	62.9222	62.9222	62.9222	8.05	0.00	0.00
40_A_4	50.3730	50.3730	50.4882	13.21	0.00	0.23
40_A_5	51.7278	51.7278	52.2136	12.28	0.00	0.94
40_A_6	61.2621	61.2621	61.2621	10.14	0.00	0.00
40_A_7	54.8554	54.8554	54.8554	13.11	0.00	0.00
40_A_8	55.9774	55.9774	56.0034	10.99	0.00	0.05
40_A_9	56.3013	56.3013	56.3013	9.48	0.00	0.00
40_A_10	57.6954	57.6954	57.6954	11.02	0.00	0.00
50_A_1	57.0235	57.0235	57.0235	19.90	0.00	0.00
50_A_2	60.5906	60.3312	60.9500	20.82	**-0.43**	0.59
50_A_3	63.5529	63.6788	63.6828	17.68	0.20	0.20
50_A_4	56.6342	56.6342	58.2334	20.96	0.00	2.82
50_A_5	64.0909	64.0909	64.0909	16.85	0.00	0.00
50_A_6	64.8116	64.8116	64.8627	22.56	0.00	0.08
50_A_7	63.6350	63.6350	63.8961	26.77	0.00	0.41
50_A_8	68.6323	68.6323	69.3458	15.80	0.00	1.04
50_A_9	58.2212	58.2212	58.8693	20.52	0.00	1.11
50_A_10	60.7796	61.2239	61.2888	25.29	0.73	0.84
Average	50.1688	50.1766	50.3185	10.21	0.01	0.26

bold value indicates a new BKS.

Table 17: Detail results of HLNS on solving PLRP large instances

Instance	BKS	f^b	f^{ave}	CT(s)	Gap^b(%)	Gap$^{\mathsf{Ave}}$(%)
60_A_1	63.6180	61.7997	61.9830	32.83	**-2.86**	**-2.57**
60_A_2	64.8337	63.5337	63.8563	34.34	**-2.01**	**-1.51**
60_A_3	64.3194	64.0061	64.0371	30.94	**-0.49**	**-0.44**
60_A_4	68.8561	68.8561	68.8561	23.16	0.00	0.00
60_A_5	67.8520	67.4505	67.7206	25.68	**-0.59**	**-0.19**
60_A_6	65.6751	64.8204	65.0034	29.56	**-1.30**	**-1.02**
60_A_7	64.7246	64.7246	64.7246	25.33	0.00	0.00
60_A_8	65.6574	65.6574	65.8816	30.95	0.00	0.34
60_A_9	67.1968	66.0875	66.0875	39.52	**-1.65**	**-1.65**
60_A_10	77.3963	75.5365	75.7463	35.60	**-2.40**	**-2.13**
70_A_1	80.3145	78.2219	78.4339	32.15	**-2.61**	**-2.34**
70_A_2	85.1052	84.0489	84.7349	32.61	**-1.24**	**-0.44**
70_A_3	71.4182	70.8533	70.8533	43.03	**-0.79**	**-0.79**
70_A_4	74.4721	72.8007	73.3267	39.17	**-2.24**	**-1.54**
70_A_5	79.0669	77.9923	78.0909	43.39	**-1.36**	**-1.23**
70_A_6	70.8563	70.3558	70.3798	33.83	**-0.71**	**-0.67**
70_A_7	78.2697	77.1333	77.4180	41.75	**-1.45**	**-1.09**
70_A_8	75.8930	75.8930	75.9117	41.60	0.00	0.02
70_A_9	79.8592	77.0350	77.8463	43.64	**-3.54**	**-2.52**
70_A_10	82.1809	81.2518	81.2609	28.02	**-1.13**	**-1.12**
80_A_1	91.4923	87.3309	88.3495	69.96	**-4.55**	**-3.44**
80_A_2	90.9719	88.7659	89.6554	50.12	**-2.42**	**-1.45**
80_A_3	96.2431	91.8364	92.2849	37.98	**-4.58**	**-4.11**
80_A_4	94.4960	91.3268	91.4289	46.47	**-3.35**	**-3.25**
80_A_5	93.4428	89.3962	89.8281	44.17	**-4.33**	**-3.87**
80_A_6	91.1502	88.2419	89.6365	44.78	**-3.19**	**-1.66**
80_A_7	90.1759	87.3974	87.5084	42.92	**-3.08**	**-2.96**
80_A_8	92.4352	86.9158	87.6682	49.15	**-5.97**	**-5.16**
80_A_9	101.0565	95.3330	95.5215	39.35	**-5.66**	**-5.48**
80_A_10	92.9522	88.5957	88.5995	41.19	**-4.69**	**-4.68**
90_A_1	92.1341	90.0784	90.3748	75.11	**-2.23**	**-1.91**
90_A_2	85.6971	82.8125	83.1454	64.40	**-3.37**	**-2.98**
90_A_3	88.6107	83.2549	83.4454	70.06	**-6.04**	**-5.83**
90_A_4	91.9519	88.7919	89.9237	60.01	**-3.44**	**-2.21**
90_A_5	100.3967	95.8919	96.9145	53.12	**-4.49**	**-3.47**
90_A_6	81.5321	76.5464	76.8275	88.66	**-6.12**	**-5.77**
90_A_7	95.2463	91.3835	91.5899	53.44	**-4.06**	**-3.84**
90_A_8	96.7539	93.9850	94.4316	65.21	**-2.86**	**-2.40**
90_A_9	85.2391	80.4393	80.7930	61.06	**-5.63**	**-5.22**
90_A_10	98.5392	94.1509	94.5841	53.83	**-4.45**	**-4.01**
Average	82.4521	80.0133	80.3666	44.95	**-2.77**	**-2.36**

bold values indicate new BKSs.

Table 18: Detailed results of HLNS on solving FPLRP large instances

Instance	C	N	P	HLNS			LNS_{DP}			LNS_{SP}		
				f^b	f^{ave}	CT(s)	f^b	f^{ave}	CT(s)	f^b	f^{ave}	CT(s)
FPLRP_20_1	20	65	20	247.92	253.03	49.56	247.92	254.11	39.27	247.92	254.60	39.89
FPLRP_20_2	20	95	29	231.56	233.70	162.83	233.45	234.21	150.09	231.56	233.70	131.85
FPLRP_20_3	20	72	22	249.69	251.76	51.21	248.79	251.13	47.23	252.54	252.92	43.04
FPLRP_20_4	20	80	25	230.22	231.01	83.57	230.27	231.83	70.61	230.27	231.68	62.64
FPLRP_20_5	20	85	26	242.24	243.20	90.58	242.24	245.15	71.70	243.94	246.68	65.46
FPLRP_20_6	20	73	23	231.82	233.38	68.82	231.82	236.79	45.99	244.24	246.21	51.23
FPLRP_20_7	20	79	24	251.66	252.07	85.56	251.66	252.07	65.84	251.66	251.86	62.41
FPLRP_20_8	20	80	25	240.73	242.04	62.37	242.18	243.47	54.54	242.92	245.77	56.42
FPLRP_20_9	20	84	26	231.20	236.27	94.84	231.60	237.28	75.27	239.94	241.11	70.36
FPLRP_20_10	20	86	27	241.39	241.39	72.93	241.39	241.39	68.55	241.39	241.39	61.73
FPLRP_40_1	40	122	37	405.73	406.91	235.16	408.04	410.14	180.00	413.94	414.77	181.61
FPLRP_40_2	40	152	46	408.94	412.21	411.15	412.01	414.06	267.71	408.09	413.30	256.40
FPLRP_40_3	40	158	48	393.47	399.35	395.25	400.68	406.55	237.79	410.87	411.79	244.33
FPLRP_40_4	40	167	51	385.79	397.29	388.27	396.13	402.30	265.76	400.59	411.55	281.40
FPLRP_40_5	40	163	50	376.03	378.31	303.23	379.85	382.08	279.58	378.00	380.19	277.97
FPLRP_40_6	40	150	46	386.09	386.09	389.54	386.09	387.39	269.21	389.18	389.18	273.36
FPLRP_40_7	40	161	49	407.54	414.03	439.49	413.39	416.95	282.82	413.81	417.57	274.99
FPLRP_40_8	40	159	48	396.04	398.36	514.64	396.04	396.04	330.41	401.26	401.26	386.99
FPLRP_40_9	40	155	47	398.26	403.48	431.94	406.43	408.55	280.66	411.60	416.81	310.90
FPLRP_40_10	40	155	47	401.62	406.50	333.45	406.21	409.37	200.47	414.78	418.15	209.05
FPLRP_60_1	60	207	63	552.41	558.12	933.20	571.38	573.19	545.80	585.24	586.30	586.22
FPLRP_60_2	60	245	74	495.93	511.72	1384.88	528.72	532.31	657.47	509.50	523.33	1013.12
FPLRP_60_3	60	243	74	508.85	515.48	1125.77	527.07	535.03	744.33	529.96	536.94	817.39
FPLRP_60_4	60	221	67	550.51	555.39	917.82	548.12	562.96	612.84	562.68	568.25	624.31
FPLRP_60_5	60	209	64	524.47	544.40	874.92	547.58	569.42	587.90	574.79	580.81	611.30
FPLRP_60_6	60	224	68	535.49	542.57	1026.03	544.71	547.35	625.30	538.50	544.85	664.05
FPLRP_60_7	60	215	65	536.79	538.89	1013.25	536.63	545.17	627.94	555.46	556.40	665.32
FPLRP_60_8	60	249	75	542.79	546.28	1260.84	530.56	540.57	847.51	537.57	541.05	862.36
FPLRP_60_9	60	231	70	543.59	546.61	1024.22	543.73	552.28	658.21	548.70	549.00	692.17
FPLRP_60_10	60	216	66	534.23	543.60	979.21	535.99	547.01	577.84	548.89	554.02	635.89
FPLRP_80_1	80	281	85	676.26	683.88	2048.03	679.56	692.02	1316.17	686.61	688.91	1359.53
FPLRP_80_2	80	300	91	659.20	676.85	2160.98	678.45	692.91	1501.39	684.89	694.20	1493.94
FPLRP_80_3	80	325	98	618.43	627.12	2991.69	624.11	644.90	1910.71	644.07	650.05	1930.44
FPLRP_80_4	80	299	90	655.93	663.50	2246.68	669.94	676.46	1467.85	659.53	668.45	1500.51
FPLRP_80_5	80	319	96	654.59	666.78	2405.07	664.45	684.83	1696.04	677.77	683.37	1765.41
FPLRP_80_6	80	266	81	709.93	714.98	1918.19	698.77	712.36	1321.45	711.95	721.34	1388.50
FPLRP_80_7	80	274	83	648.15	664.17	1847.01	672.79	679.45	1242.31	682.08	685.03	1333.12
FPLRP_80_8	80	301	91	634.25	639.43	2032.68	635.30	653.26	1447.87	639.38	651.99	1489.14
FPLRP_80_9	80	316	96	635.84	660.39	2361.71	667.42	674.62	1587.28	664.76	672.58	1729.57
FPLRP_80_10	80	308	93	642.74	653.82	2376.83	659.30	667.53	1549.75	658.67	662.43	1646.01