Flexible Parcel Delivery to Automated Parcel Lockers: Models, Solution Methods and Analysis

Ido Orenstein, Tal Raviv, Elad Sadan

Department of Industrial Engineering, Tel Aviv University

Email: talraviv@tauex.tau.ac.il

June 2019

Keywords: Last Mile Delivery, Vehicle Routing, Automated Parcel Lockers, Mixed-Integer Programming, Heuristics

Abstract

In this study, we introduce a logistic model for the delivery of small parcels to a set of service points (SPs), and we present effective methods for solving it. In the traditional delivery model, each recipient specifies a single location at which they wish to receive the parcel; however, when SPs are used, many recipients may have no strong preference among several locations, e.g., near the recipient's home address, near the recipient's office, or in the recipient's favorite shopping mall. If some recipients are flexible and willing to provide the sender with more than one delivery location, it is possible to perform the delivery task at lower cost and within a shorter amount of time. Our solution methods are based on the concepts of the savings heuristic, the petal method and tabu search with a large neighborhood. An extensive numerical study is conducted to evaluate our solution methods and demonstrate the benefits of our model compared to the traditional nonflexible one. We also present a simulation study to demonstrate that our model can be adapted to a stochastic and dynamic environment.

1 Introduction and literature review

This paper addresses the last leg of the delivery process for small parcels, i.e., from a regional depot to the recipient. This leg is responsible for a significant share of the costs in the parcel delivery industry (Goodman, 2005). One method to reduce these costs is by delivering parcels to recipients through local service points (SPs) located near the recipients instead of bringing each parcel directly to the recipient's address. These SPs may be either staffed facilities, such as post offices and grocery stores, or self-service facilities, such as automated parcel lockers. Cost savings are achieved through the consolidation of shipments to fewer locations, avoiding the need for time synchronization between the couriers and recipients and eliminating the time-consuming task of locating the recipients' addresses (Faugere and Montreuil, 2017). From the recipient perspective, receiving parcels at SPs rather than at home may be less convenient, but the cost savings may translate into lower shipment tariffs. Moreover, for some recipients, avoiding the need to synchronize with the courier may be desirable.

Using the shipment data of a courier company operating in West Sussex in the United Kingdom, Song et al. (2009) found that the use of staffed SPs instead of home delivery significantly reduced travel costs and the average delivery time. The policy of this company is to call recipients to ask them to collect their parcels from the depot if they are not available to receive their parcels at home. Under this policy, the mean travel

distance of the recipients is also reduced since the SPs are generally located much closer than the depot.

The use of SPs for parcel delivery is gaining popularity in Europe and the US. For example, in France, as of 2014, more than 20% of parcels were already being delivered to SPs in stores (Morganti et al., 2014). It is probable that the share of automated and staffed SPs has increased since then.

An automated parcel locker system (APLS) is a parcel collection service that allows customers to have their parcels delivered to SPs and pick them up at any time of day using digital pickup codes. The use of automated service points may further economize the delivery process since automated SPs are available 24/7 and lines are less likely to form at automated SPs than at attended facilities. Such services are provided by many mail and courier companies, e.g., Amazon Locker in the US, BoxIt in Israel and DHL PackStation in Germany. Each SP hosts lockers of various sizes and a terminal that is used by the courier and recipients to deposit and collect parcels. The SPs are typically located on public premises, such as at gas stations or public transit stations. The increasing popularity of APLSs is creating opportunities for more efficient distribution models for small parcels.

In this paper, we introduce a logistic model for parcel distribution that is well suited for APLSs and present effective methods for solving it. An extensive numerical study is conducted to evaluate these methods and demonstrate the benefits of the proposed logistic model compared to traditional methods.

In traditional delivery models, each recipient specifies a single location at which to receive his or her parcel. However, when an APLS is used, many recipients may have no strong preference among several delivery locations; for example, a recipient may have equally convenient access to three different SPs along his or her commuting route from work to home if the parcel is delivered during the day or to another SP at walking distance from his or her home if the parcel is delivered in the evening. If some recipients are flexible and willing to provide the sender with more than one possible delivery location, then the delivery task can be completed at lower cost and within a shorter time.

The goal of this study is to formulate and solve a parcel delivery model for determining the number of vehicles and their routes and assigning parcels to vehicles and destinations. We refer to this model as the flexible parcel delivery (FPD) problem, which is defined as follows. We are given a set of parcels, initially located at a central facility (depot), and a set of SPs, each with a specific capacity. Each parcel is characterized by a set of possible destination SPs, a size and a penalty for failing to deliver it during the next shift. Such penalties can be updated over time to represent the urgency of each parcel. The parcels are distributed using an unlimited fleet of vehicles with identical capacity. The travel time and cost of travel between each pair of locations are given. In addition, there is a fixed handling time per parcel, which represents the time that is required to unload a parcel from a vehicle at an SP. A solution to the problem consists of a set of tours for the vehicles that visit each SP at most once, and a set of assignments of parcels to vehicles and destination SPs. A feasible solution must satisfy the capacity constraints of the vehicles and SPs as well as a shift length constraint that considers both the travel and handling times. The objective is to minimize the total travel cost, the total vehicle cost, and the sum of the penalties due to undelivered parcels.

In this paper, we focus on an extension of the problem in which the SPs and vehicles are divided into lockers and cells, respectively, of specific sizes. Each locker or cell may contain at most one parcel at a time. The solution specifies a set of

assignments of parcels to lockers and cells, in which each parcel can be assigned only to a locker and cell of compatible size (i.e., one that is at least as large as the parcel).

The rest of the paper is organized as follows. In Section 2 we review the relevant literature and identify the gap closed by the current study. In Section 3, a detailed definition and mathematical formulation of the FPD problem are presented. In Section 4, our solution methods for this problem are introduced. In Section 5, a numerical experiment conducted to test our solution methods is reported. It is shown that flexibility makes the delivery process more efficient. In Section 6, to strengthen our conclusions from Section 5, we demonstrate how our model can be used in a multiperiod, dynamic and stochastic setting by applying it in a rolling horizon scenario and allowing the users to redefine their sets of possible destinations if their parcels were not delivered during the current period (shift). In Section 7, some concluding remarks and directions for further research are presented.

2 Literature review

The FPD problem is a vehicle routing problem (VRP). Vehicle routing is a fundamental task for many private and public organizations. It is crucial for shipping goods in a cost-effective manner and for local transport within a factory or warehouse building. Effective and efficient vehicle routing may also have economic and environmental effects: shorter routes for vehicles of higher capacity reduce pressure on the road infrastructure, improve traffic flow, and contribute to decreasing the negative externalities of transportation.

VRPs involve optimizing routes for a fleet of vehicles that need to transport goods, passengers, etc. For extensive reviews and classification of VRPs, see Golden et al. (2008), Drexl (2012), and Toth and Vigo (2014). There are several VRP variants that share certain characteristics with the FPD problem. The classic variant has a single objective and is concerned only with minimizing the total cost or length of all routes when visiting all customers subject to vehicle capacity or route length constraints. This capacitated vehicle routing problem (CVRP) was first introduced by Dantzig and Ramser (1959). An overview of the CVRP can be found in Laporte et al. (2000). The addition of distance constraints to the CVRP yields the distance-constrained CVRP (DCVRP, Laporte et al., 1984). Constraints can be used to limit the distance, duration or cost of the routes.

A scheduling-routing-loading model with customer capacity constraints was addressed by Reyes et al. (2007). These constraints are the basis of the customer capacity vehicle routing problem (CCVRP), and they limit the number of vehicles that can be at a given location at the same time.

The multivehicle covering tour problem (m-CTP) introduced by Hachicha et al. (2000) is defined by a set of locations V that the vehicles can visit and a set of locations W that should be served. Each $w \in W$ is associated with one or more $v \in V$, and the goal is to find a set of m minimum-length tours through elements of V that cover all elements of W. The flexibility aspect of the FPD problem is captured by this model when V is the set of SPs and W is the set of parcels that should be "covered".

Ghiani and Improta (2000) introduced the generalized VRP (GVRP), in which a fleet of vehicles serves a set of customers who are divided into clusters. Each cluster is visited exactly once by only one of the vehicles. A customer can be served when a vehicle visits any of the customers in that customer's cluster. The vehicles are capacitated, and each cluster has its own demand. The objective is to find a minimum-distance set of routes that allows all customers to be served. In this model, the

destination flexibility stems from the fact that the planner needs to choose only one location in each cluster to visit. Biesinger et al. (2018) introduced a genetic algorithm combined with a solution archive for solving the generalized VRP with stochastic demand at the customers. Miranda et al. (2018) extended the generalized VRP to a bi-objective problem that also considers the costs of delivering the goods to their destinations within each cluster. Their model considered only a single vehicle.

Another variation is the vehicle routing problem with profits (VRPP), which shares components of the objective function with our FPD problem. Unlike the CVRP, the VRPP is characterized by a profit value associated with each customer; the objective is to maximize the total net profit from the visited customers after travel costs (Archetti et al., 2014). Equivalently, it is possible to associate a penalty with each customer who is not visited and to formulate the objective as the minimization of the sum of the travel costs and penalties.

Most VRPP studies have examined variations of the single-vehicle case, also known as the traveling salesman problem (TSP) with profits (Feillet et al., 2005), the orienteering problem (Golden et al., 1987; Vansteen et al., 2011), the selective TSP (Laporte and Martello, 1990) and the prize-collecting TSP (Balas, 1989).

The team orienteering problem (TOP) is a well-studied version of the multivehicle generalization of the VRPP. The TOP includes a set of geographically scattered customers, each assigned a profit value. Each vehicle must visit a subset of customers within a given time limit. The objective is to maximize the collected profit while satisfying the time limit for each vehicle (Archetti et al., 2007).

In the well-studied traveling purchaser problem (TPP), given a list specifying the products and quantities required, a purchaser must find a purchasing plan that exactly satisfies the product demand by visiting a subset of suppliers on a unique tour. The model contains flexibility in the selection of the supplier for each product. The objective of the purchaser is to minimize the combined travel and purchase costs. The problem combines supplier selection, route construction and product purchase planning. This problem dates back to the 1960s, and more recent multivehicle variations also exist. For a review of the state of the art in TPP research, see Manerba et al. (2017).

Raviv et al. (2013) modeled a VRP variant in a bike-sharing system - the static bicycle repositioning problem (SBRP). In their model, they included the time needed to load and unload bicycles on and off vehicles (handling time), vehicle capacity, station capacity, and route time limits. This problem is a type of inventory routing problem in which the decisions are which customers to visit and when as well as how many goods to deliver to each. The goods are not identified by specific destinations. See Moin and Salhi (2007) for an overview.

Reyes et al. (2017) studied the vehicle routing problem with roaming delivery locations (VRPRDL). This problem is motivated by a new technology that enables the delivery of parcels to the trunk of the recipient's car. The location of the car varies over time, with known, nonoverlapping time windows for each location. The goal is to deliver all parcels to the correct cars.

Lang et al. (2014) considered a variation of the VRP with time windows in which the goal is to minimize the total fuel consumption, which is affected by the vehicle load. There are several alternative stop points for each customer; this scenario is motivated by the routing of a fleet of couriers in an urban environment. Each courier can decide to stop his or her vehicle either on the same side of the street as the customer or on the opposite side. The time window for each possible stop point is adjusted to reflect the walking time from the stop point to the customer location.

The FPD problem is different from all variants of the VRP that have been studied to date in that each item to be delivered is identified and characterized by a set of optional destinations (SPs) and a penalty for not delivering it at all. In a feasible solution, not all items must be delivered, and an item may not be delivered even if its destination is visited. More generally, vehicle routing models that combine the flexible delivery of unique goods, time constraints, handling time, vehicle loading considerations, and customer capacity constraints have not been studied. Table 1 lists some characteristics of the VRP variants discussed above as well as the FPD problem. Indeed, the FPD problem stands out as a unique and rich vehicle routing model. The first four characteristics considered in the table are the existence of vehicle capacity constraints, route length constraints, customer capacity constraints and handling times. The next column concerns the identifiability of particular items. Items may be identified by their locations, urgency, dimensions and time windows. Next, we characterize the destination flexibility if applicable and then list the characteristics of the objective function.

Table 1. Comparison of VRP variants

Variant	**Source**	**Vehicle capacity**	**Route length**	**Customer capacity**	**Handling time**	**Items identified by**	**Destination flexibility**	**Objective function**
CVRP	Dantzig and Ramser (1959)	Limited						Distance, fleet size
CCVRP	Reyes, et al. (2007)			Limited				Distance
DCVRP	Laporte et al. (1984)	Limited	Yes					Distance
m-CTP	Hachicha et al. (2000)		Yes			Location	Flexible locations but all items are delivered	Distance
TOP	Archetti, et al. (2007)		Yes					Distance, penalty
TPP	Reviewed by Manerba et al. (2017)	In multivehicle versions		In some versions			Flexible locations but all products are purchased	Distance, prices

SBRP	Raviv, Tzur & Forma (2013)	Limited	Yes	Limited	Depends on the delivered quantity			Distance, penalty
VRPRDL	Reyes (2017)	Limited				Location Time window	Yes, for different time windows	Distance
GVRP	Ghiani and Improta (2000)	Limited					One location in each cluster	Distance
VRP- alt. stop points	Lang et al. (2014)	Limited		At the alternative points only	Depends on the stop point		Only nearby alternatives	Fuel
FPD	This study	Limited per product size	Yes	Limited per product size	Depends on the delivered quantity	Location Urgency Dimension	Several alternatives per item and items can be skipped	Distance, penalty, fleet size

3 Problem definition and formulation

In this section, we present a formal definition and a mixed-integer linear programming (MILP) formulation of the FPD problem. The context here is an optimal planning problem for a single shift with the possibility of postponing the delivery of some parcels to subsequent shifts. The application of the static single-shift model presented here in a dynamic multishift environment using a rolling horizon framework will be discussed in Section 6. The FPD problem is defined by the following inputs:

A set of SPs, where each SP is characterized by a set of available lockers of different sizes. Each locker may contain at most one parcel at a time. The number of different locker sizes in the system is assumed to be small (e.g., three or four). The assortment of available lockers defines the effective capacity of the SP. The actual capacity of the SP may be larger, but some of the lockers may be occupied by parcels that were dropped off in previous shifts and have not yet been collected by their recipients.

A matrix of the travel times between the SPs and between the depot and the SPs, where the travel cost per time is also given.

A fleet with an unlimited number of identical vehicles. The cost of operating each vehicle during a shift is given. Each vehicle is divided into cells of different sizes. These sizes are assumed to be identical to the sizes of the lockers in the SPs. Each cell may contain at most one parcel at a time. The assortment of cells defines the capacity of the vehicle.

A set of parcels, where each parcel is associated with a set of SPs to which it can be delivered, a penalty for failing to deliver that parcel and a set of locker/cell sizes with which it is compatible. The degree of flexibility is defined by the number of different potential destinations for the parcels. The penalty represents the urgency class

and possibly the seniority of the parcel in the system. Thus, in a multishift setting, the operator may raise the penalty for a parcel after each shift in which that parcel remains undelivered.

A fixed time associated with each operation of **unloading a parcel** from a vehicle and depositing it in its locker. Our model determines which parcels should be loaded onto each vehicle, but it is assumed that the vehicle loading operation commences before the beginning of the planning horizon.

The length of the shift that constitutes the planning horizon. All utilized vehicles depart from the depot at the beginning of the planning horizon and must return by the end of this period.

A solution to the problem consists of a set of routes traveled by the vehicles, the identities of the parcels loaded on each vehicle, their destination SPs and the sizes of the cells and lockers to which they are assigned. A feasible solution satisfies the shift length constraint and the capacity constraints of the vehicles and SPs. The **objective** is to minimize the sum of the following three cost components: the total travel cost for all vehicles, the fixed cost for each utilized vehicle, and the total penalty for all parcels that are not delivered.

Our model is based on the following simplifying assumptions:

1. Each SP can be visited only once per shift by a single vehicle, i.e., there is no "split delivery". This assumption is typically not very restrictive in our application since the number of parcels that should be delivered to each SP is small compared to the vehicle capacity.
2. The availability of lockers at the SPs is known before the beginning of the planning horizon. The generated plan ignores the possibility that parcels may be collected from the SPs during the shift.
3. The sizes of the lockers/cells are nested, i.e., a larger locker can contain any parcel that can also be contained in a smaller one. This assumption is well aligned with the automated parcel locker equipment that is available on the market.

Next, we introduce the following notation to define our MILP model:

Sets	
$\boldsymbol{S}$	SPs; $S = \{1, \dots, n\}$.
$\boldsymbol{S_0}$	Locations, including the depot and SPs; $S_0 = S \cup \{0\}$.
$\boldsymbol{Q}$	Parcels to be potentially delivered; $Q = \{1, \dots, p\}$.
$\boldsymbol{S_q}$	SPs to which parcel $q \in Q$ can be delivered; $S_q \subset S$.
$\boldsymbol{J}$	Indices of the cell/locker sizes (types), in decreasing order of size.

Parameters	
$\boldsymbol{T}$	Maximum route duration (the length of the planning horizon).
$\boldsymbol{V_{ik}}$	Total driving time from SP i to SP k (travel time matrix).
$\boldsymbol{C_j}$	Number of cells of size j in each vehicle.
$\boldsymbol{B_{i,j}}$	Number of available lockers of size j at SP i.

$\boldsymbol{P_q}$	Penalty for not delivering parcel q.
$\boldsymbol{D_q}$	Minimal size of a cell/locker in which parcel q can fit.
$\boldsymbol{\alpha}$	Travel cost per unit of driving time.
$\boldsymbol{\beta}$	Unloading time for a parcel.
$\boldsymbol{\gamma}$	Fixed vehicle cost.

Decision variables

$\boldsymbol{x_{ik}}$	Binary variable; equals 1 if a vehicle travels from SP i to SP k for all $i, k \in S$.
$\boldsymbol{y_{qik}}$	Binary variable; equals 1 if parcel q travels on a vehicle from SP i to SP k for all $q \in Q$ and $i, k \in S$
$\boldsymbol{z_{qi}}$	Binary variable; equals 1 if parcel q is delivered to SP i for all $q \in Q, i \in S_q$.
$\boldsymbol{u_i}$	Arrival time of a vehicle at location i (u_0 is assumed to be 0).

$$\min \alpha \sum_{i \in S_0} \sum_{k \in S_0} V_{ik} x_{ik} + \sum_{q} \left(1 - \sum_{i \in S_q} z_{qi}\right) P_q + \gamma \cdot \sum_{k \in S} x_{0k} \quad (1)$$

s.t.

$$y_{qik} \leq x_{ik} \quad \forall q \in Q, i, k \in S_0 \quad (2)$$

$$\sum_{q: D_q \leq j} y_{q0i} \leq x_{0i} \cdot \sum_{j'=1}^{j} C_{j'} \quad \forall j \in J, i \in S \quad (3)$$

$$\sum_{q: D_q \leq j \wedge i \in S_q} z_{qi} \leq \sum_{j'=1}^{j} B_{ij'} \quad \forall i \in S, j \in J \quad (4)$$

$$\sum_{k \in S_0} x_{ik} \leq 1 \quad \forall i \in S \quad (5)$$

$$\sum_{k \in S_0} x_{ik} = \sum_{k \in S_0} x_{ki} \quad \forall i \in S_0 \quad (6)$$

$$\sum_{i \in S_0} y_{qik} = \sum_{i \in S_0} y_{qki} \quad \forall q \in Q, k \in S \setminus S_q \quad (7)$$

$$\sum_{i \in S_0} y_{qik} = \sum_{i \in S_0} y_{qki} + z_{qk} \quad \forall q \in Q, k \in S_q \quad (8)$$

$$u_k \geq u_i + \beta \sum_{q: i \in S_q} z_{qi} + V_{ik} - (1 - x_{ik})T \quad \forall i \in S_0, k \in S \quad (9)$$

$$u_0 = 0 \tag{10}$$

$$u_k + \beta \sum_{q:k \in S_q} z_{qk} + V_{k0} \leq T \quad \forall k \in S \tag{11}$$

$$\sum_{i \in S_q} z_{qi} \leq 1 \qquad \forall q \in Q \tag{12}$$

$$\sum_{k \in S} y_{q0k} \leq \sum_{i \in S_q} z_{qi} \qquad \forall q \in Q \tag{13}$$

$$x_{ik} \in \{0,1\} \quad i,k \in S_0 \tag{14}$$

$$y_{qik} \in \{0,1\} \quad \forall q \in Q, i,k \in S_0 \tag{15}$$

$$z_{qi} \in \{0,1\} \, \forall q \in Q, i \in S_q \tag{16}$$

$$u_i \geq 0 \quad \forall \, i \in S_0 \tag{17}$$

The model can be described as follows:

(1) The objective function minimizes the sum of the three cost components: the travel costs for the vehicles, the penalties for parcels not delivered, and a fixed cost for each vehicle used. The first and third components are jointly referred to as the *transportation cost.*

(2) The decision variables x and y are associated with each other. For each parcel carried on a vehicle in a section (between two locations), the corresponding section must be a part of a vehicle route.

(3) The total number of parcels of a certain size or larger that are sent from the depot to a specific SP must be at most equal to the total capacity of the vehicle for parcels of that size or larger. This inequality guarantees that each vehicle has sufficient capacity for the parcels of each size that it is to deliver, independent of the assignment of parcels to particular cells.

(4) The total number of parcels of size j or larger that are delivered to SP i must be no greater than the number of available lockers of this size or larger at SP i. This inequality guarantees that each SP has sufficient available capacity for the parcels of each size that are to be delivered to it, independent of the assignment of parcels to particular lockers.

(5) At most one vehicle may depart from any SP since we assume that split deliveries are not allowed.

(6) The number of vehicles that arrive at a location is equal to the number of vehicles that depart from it (vehicle flow conservation equation). Note that according to (5), at each SP, this number is either zero or one.

(7) Each parcel that travels to an SP that is not one of its possible destinations must leave that SP.

(8) Each parcel that travels to one of its possible destination SPs either leaves that SP or is delivered to it. Together, (7) and (8) stipulate the conservation of parcel flow.

(9) If a vehicle travels from SP i to SP k, then its arrival time at SP k is at least its arrival time at SP i (or 0) plus the time required to unload all parcels delivered to SP i and the travel time from SP i to SP k. This inequality eliminates subtours that do not contain the depot.
(10) The arrival time at the depot is zero.
(11) The total time for each vehicle's tour is limited to at most T.
(12) Each parcel can be delivered to at most one SP.
(13) Only delivered parcels can leave the depot, each on at most one vehicle.
(14)-(17) The domains of the decision variables are defined.

This problem is intractable because it is a generalization of various NP-hard problems, such as the CVRP. Obtaining a reasonably approximated solution to (1)-(17) using a commercial solver is not practical for most real-life instances, as we demonstrate in Section 5. In the next section, we present heuristic algorithms designed to generate good solutions for large problem instances.

4 Methodology

In this section, we present two mathematical construction heuristics for solving the FPD problem: a savings heuristic based on the idea proposed by Clarke and Wright (1964) and a heuristic based on the petal heuristic of Foster and Ryan (1976) and Ryan et al. (1993). In both cases, the subproblems are solved to optimality using a commercial MILP solver. In addition, we design a tabu search heuristic that can be used to improve the solutions obtained with these construction heuristics. Via the numerical experiment described in Section 5, we will show that these heuristics reach good solutions in a relatively short time.

4.1 Savings heuristic

The savings heuristic for the classic CVRP was introduced by Clarke and Wright (1964). Since that time, this heuristic has been adapted to many VRP variants. For examples of its application, see Toth and Vigo (2014), Chapters 4, 8, and 12. The fundamental idea of the algorithm is to repeatedly unify existing routes to reduce the total cost. The algorithm starts with a set of simple routes – tours consisting of the depot and one customer. In each iteration, the algorithm checks for the potential total cost savings that can be obtained by unifying each pair of routes. The pair that yields the feasible tour with the largest savings is unified, and the algorithm is repeated until no feasible unification can yield a positive savings. Altinkemer and Gavish (1991) introduced an improvement of the savings heuristic by optimizing each candidate pair of routes for unification by solving the TSP.

In the FPD problem, the calculation of the savings obtained by unifying two routes should consider all three cost components, i.e., the travel times, penalties for undelivered parcels and vehicle costs. The value of each potential unification is evaluated in two stages: the routes are determined first, followed by the delivery plan.

In the first stage of evaluating a candidate solution, the unification of two routes reduces the vehicle cost component by the fixed cost of a single vehicle and the associated travel cost. The travel cost savings value is calculated as the difference between the sum of the travel times of the two routes and the travel time of the unified route, as calculated by solving the TSP using CPLEX. Since the routes are rather short,

these subproblems can be solved quickly. If the travel time of the unified route exceeds the time limit T, then the unified route is infeasible.

In the second stage of the evaluation, the effect of route unification on the total penalty is obtained by comparing the total penalty for the current solution with the optimal penalty that can be obtained with the new set of routes. Note that due to the flexibility of the parcel destinations, route unification may affect the delivery of parcels to any SP in the system, not only those that are visited by the unified route. Given the set of routes (for each candidate unification), the algorithm finds the optimal assignment of parcels to vehicles and SPs by solving a streamlined version of the MILP problem defined by (1)-(17), in which the vehicle route variables are fixed. The objective function serves to minimize the total penalty for the parcels that are not delivered on these routes. We refer to this subproblem as the *loading problem*. Note that route unification can only increase the total penalty since the capacity and route length constraints are tightened, while the two other cost components are reduced.

The loading problem for a given set of routes is formulated using some additional notation as follows: R denotes the set of the routes under consideration, the set G_r consists of all SPs visited by a route $r \in R$, and $TSP(G_r)$ is the shortest route that visits all of the SPs in G_r plus the depot. Let $U = \bigcup_{r=1}^{|R|} G_r$ be the set of all visited SPs, and let $\mathrm{T}_r = \mathrm{T} - \mathrm{TSP}(\mathrm{G}_r)$ be the remaining time available for unloading parcels on route r. Next, we redefine the decision variables as shown below.

Decision variables	
$\boldsymbol{y_{q,r}}$	Binary variable; equals 1 if parcel q is assigned to a vehicle that is traveling on route r. This variable is defined for each tuple $(q \in Q, r \in R: S_q \cap G_r \neq \emptyset)$.
$\boldsymbol{z_{q,i}}$	Binary variable; equals 1 if parcel q is delivered to SP i. This variable is defined for each tuple $(q \in Q, i \in S_q \cap U)$.

The remaining notation is the same as that used in (1)-(17).

$$\min \sum_{q \in Q} P_q \left(1 - \sum_{i \in S_q \cap U} z_{qi}\right) \tag{18}$$

s.t.

$$\sum_{q: D_q \leq j \wedge (S_q \cap G_r \neq \emptyset)} y_{qr} \leq \sum_{j'=1}^{j} C_{j'} \qquad \forall j \in J; r \in R \tag{19}$$

$$\sum_{q: D_q \leq j \wedge i \in S_q} z_{qi} \leq \sum_{j'=1}^{j} B_{ij'} \qquad \forall i \in U, j \in J \tag{20}$$

$$\beta \sum_{q: (S_q \cap G_r \neq \emptyset)} y_{qr} \leq T_r \qquad \forall r \in R \tag{21}$$

$$\sum_{i \in S_q \cap G_r} z_{qi} \leq y_{qr} \qquad \forall q \in Q, r \in R: S_q \cap G_r \neq \emptyset \qquad (22)$$

$$\sum_{i \in S_q \cap U} z_{qi} \leq 1 \qquad \forall q \in Q: S_q \cap U \neq \emptyset \qquad (23)$$

$$\sum_{r:(S_q \cap G_r \neq \emptyset)} y_{qr} \leq 1 \qquad \forall q \in Q: S_q \cap U \neq \emptyset \qquad (24)$$

$$y_{qr} \in \{0,1\} \qquad \forall q \in Q, r \in R: S_q \cap G_r \neq \emptyset \qquad (25)$$

$$z_{qi} \in \{0,1\} \qquad \forall q \in Q, i \in S_q \cap U \qquad (26)$$

The objective function (18) minimizes the total penalty for all parcels that are not delivered along the fixed routes. Constraints (19) and (20) are the capacity constraints for the vehicles and SPs, similar to (3) and (4). Constraints (21) limit the time available for unloading parcels along each route. Constraints (22) state that if a parcel is delivered to an SP on a given route, then it is carried by the vehicle that serves that route. Constraints (23) and (24) state that each parcel can be delivered to only a single SP via only one route. Constraints (25) and (26) define the domains of the decision variables.

Since the savings calculation for each pair of existing routes involves solving two optimization problems (the TSP and the loading problem), we have devised a method that allows the calculations for many dominated pairs, i.e., pairs whose unification cannot yield the largest savings in the current iteration, to be skipped.

Recall that each unification of two routes yields some savings in terms of the vehicle and travel costs and some additional cost due to the increased penalty. We refer to the former as the *transportation savings* and the latter as the *added penalty* for each pair of routes that can be unified. The net transportation savings after the added penalty is called the *total savings* for a pair.

While the savings heuristic runs, we store a list, L, of feasible route pairs along with their potential transportation savings. In each iteration of the savings heuristic, the procedure loops through L in nonascending order of the transportation savings. For each route pair with a transportation savings greater than the best total savings encountered so far, we calculate the added penalty by solving the loading problem. If the total savings value is greater than the best total savings found so far, we store this pair as the best candidate for unification and update the value of the best total savings found. Once we encounter a pair with smaller transportation savings than the best total savings found so far, we exit the loop. Note that all remaining pairs in the sorted list will have smaller total savings since their transportation savings are smaller even without considering the added penalty. The best route pair is unified. The list L is updated by removing each pair that contains a member of the unified pair, and new pairs that contain the newly created route are added. The transportation savings of the new pairs are calculated by solving TSPs, and the savings heuristic proceeds to the next iteration. In Pseudocode 1, we present the details of a single iteration of the savings heuristic.

Pseudocode 1 – Select the best pair of routes for unification in a single iteration:

L = list of feasible pairs of routes in descending order of transportation savings

```
BestSavings = 0
For each pair in L
    If pair.transportationSavings >BestSavings
        newLoadingCost = optimal solution to the loading problem with the unified pair
        CostSavings = currentLoadingCost –newLoadingCost+ pair.transportationSavings
        If CostSavings > BestSavings
            BestSavings = CostSavings
            bestPair = pair
            bestLoading = newLoadingCost
    Else

break
remove bestPair from L
Insert into L new pairs that contain the unified pair and each other route in L
Calculate the travel cost savings values for the new unified routes (by solving TSP problems)
currentLoadingCost = bestLoading
```

To reduce the time needed to solve the loading problems in each iteration, we add a constraint to the model that bounds the value of the objective function such that the total savings cannot be less than the best savings found so far in this iteration. In many cases, this results in an infeasible loading problem, and the solver terminates faster.

After the savings heuristic terminates, the algorithm checks the profitability of each of the obtained routes. Routes for which the total penalty for the parcels delivered on those routes is smaller than the travel and vehicle costs are not profitable. Using an iterative procedure, we select the worst of the nonprofitable routes, remove it from the solution, and re-solve the loading problem with the remaining routes. This process is repeated until all routes are profitable. Note that the removal of a nonprofitable route in one iteration may cause other nonprofitable routes to become profitable ones in the new solution to the loading problem. Therefore, we remove the nonprofitable routes one by one.

4.2 Petal heuristic

The petal heuristic for the CVRP was introduced by Foster and Ryan (1976) and was improved by Ryan et al. (1993). In the first step of the petal heuristic, a TSP solution for all customers (but not the depot) is found using either some heuristic or an exact method. This TSP solution is referred to as the *grand tour*. In the second step, *petal routes* are created. The petals are contiguous subsequences of customers along the grand tour, and each consists of a set of customers that can be served by a single vehicle, meaning that their total demand does not exceed the vehicle capacity. A *petal route* is constructed by solving a TSP for the customers on a petal in addition to the depot. In the third step, a set of petal routes that covers all customers while minimizing the total cost is selected by solving a set-covering problem. Note that the number of considered routes is limited to quadratic order in the number of customers, whereas the number of all potential routes is exponential in the number of customers. Petal routes are attractive since they consist of sets of customers that are geographically close together.

In this paper, we adapt the petal heuristic to the FPD problem. Only petal routes that satisfy the shift length constraint are considered, and heuristic domination criteria are used to further reduce the number of candidate routes. Note that if the triangle inequality holds, there is no need to solve the TSP for all possible petals. Once a tour that visits a subset of SPs is found to exceed the shift time limit, all subsets that contain it can be eliminated from consideration. We use CPLEX to find the optimal grand tour and to solve the TSPs for the petals. In the second step, we only create potential tours for the vehicles; we do not assign parcels to vehicles and SPs yet.

In an actual distance matrix obtained from geographic information systems (GIS), one may encounter some minor violations of the triangle inequality, due to some rounding errors and noise in the data collection process. In these cases, adding an SP to a route may result in shortening its travel time. Therefore, there may be some pathological cases in which an infeasible route, with a total travel time that is slightly larger than T, may be a subset of a feasible route that is slightly shorter than T. However, these routes are unlikely to be in the optimal solution since routes that exploit nearly all the planning horizon (T) for travelling (leaving very short time for parcel unloading operations) are anyway unattractive for our petal heuristic.

In the third step, we simultaneously select an optimal subset of the routes to be served and assign parcels to vehicles and SPs. That is, we produce a plan that minimizes the total vehicle, travel and penalty costs. Note that in our heuristic, not all SPs have to be covered. This problem is formulated as an MILP model, with the same notation used in the savings heuristic formulation defined in (18)-(26). However, the set of routes R now represents all of the candidate petal routes rather than a fixed set in each iteration of the savings heuristic. The meanings of G_r and T_r are also changed accordingly. The decision variables and the problem formulation, given by (27)-(37), are presented below.

Decision variables

$\boldsymbol{x_r}$	Binary variable; equals 1 if route r is served by a vehicle.
$\boldsymbol{y_{qr}}$	Binary variable; equals 1 if parcel q is assigned to a vehicle that is traveling on route r. This variable is defined for each tuple $(q\epsilon Q, r\epsilon R_q)$.
$\boldsymbol{z_{qi}}$	Binary variable; equals 1 if parcel q is delivered to SP i. This variable is defined for each tuple $(q\epsilon Q, i\epsilon S_q)$.

$$\min\ \alpha \sum_{r\in R} T_r x_r + \sum_{q\in Q}\left(1 - \sum_{i\in S} z_{qi}\right) P_q + \gamma \sum_{r\in R} x_r \tag{27}$$

s.t.

$$\sum_{q:(D_q\le j)\wedge(G_r\cap S_q\neq\emptyset)} y_{qr} \le x_r \sum_{j'=1}^{j} C_{j'} \qquad \forall r\in R, j\in J \tag{28}$$

$$\sum_{q:(D_q\le j)\wedge(i\in S_q)} z_{qi} \le \sum_{j'=1}^{j} B_{ij'} \qquad \forall i\in S, j\in J \tag{29}$$

$$z_{qi} \le \sum_{r:i\in G_r} y_{qr} \qquad \forall q\in Q, i\in S_q \tag{30}$$

$$\beta \cdot \sum_{q\in Q:G_r\cap S_q\neq\emptyset} y_{qr} \le x_r \cdot T_r \qquad \forall r\in R \tag{31}$$

$$\sum_{r:i\in G_r} x_r \leq 1 \qquad \forall i \in S \qquad (32)$$

$$\sum_{r\epsilon R: G_r \cap S_q \neq \emptyset} y_{qr} \leq 1 \qquad \forall q \in Q \qquad (33)$$

$$\sum_{i\in S_q} z_{qi} \leq 1 \qquad \forall q \in Q \qquad (34)$$

$$x_r \in \{0,1\} \qquad \forall r \in R \qquad (35)$$

$$y_{qr} \in \{0,1\} \qquad \forall q\epsilon Q, r\epsilon R: G_r \cap S_q \neq \emptyset \qquad (36)$$

$$z_{qi} \in \{0,1\} \qquad \forall q \in Q, i \in S_q \qquad (37)$$

The objective function (27) minimizes the sum of the three cost components: the travel cost, the penalty cost and the fixed cost per vehicle. Constraints (28) and (29) are the capacity constraints for the vehicles and SPs, formulated similarly to (3) and (4). Constraint (30) states that each parcel that is delivered to a specific SP will be delivered on a route that serves that SP. Constraint (31) limits the unloading time along each selected route. Constraint (32) states that each SP can be served by only one vehicle. Constraints (33) and (34) state that each parcel can be delivered via only one route to only a single SP. Constraints (35)-(37) define the domains of the decision variables.

We apply a heuristic consideration to further reduce the number of considered routes. To this end, we define the *profitability bound* of a petal route as an upper bound on the penalty saved by delivering parcels to the route's SPs minus the travel and fixed vehicle costs. The upper bound on the saved penalty is calculated by greedily adding parcels to the route in nonincreasing order of their penalties while maintaining the shift length constraint. A petal route is a *promising route* if its profitability bound is positive and is no lower than the profitability bound of any shorter petal route contained in it. For example, if the profitability bound of the route constructed from petal {3,1} is 100 and the profitability bound of the route constructed from {3,1,7} is 90, then the latter is not considered a promising route. Nonpromising routes can be excluded from the set *R* in (27)-(37), thereby significantly reducing the solution time for the model. Note that the longer a petal route is, the weaker its profitability bound is likely to be because it can be assigned more parcels that could be delivered on other routes in the solution. Therefore, shorter routes with higher profitability bounds are likely to be more profitable.

4.3 Tabu search

The tabu search framework is a framework for designing neighborhood search heuristics. To implement it, one needs to define an algorithm for constructing an initial solution, a neighborhood to be examined in each iteration, a tabu mechanism, and a stopping criterion.

In our implementation, the initial solution is obtained through either the savings heuristic or the petal heuristic, as described above. In Section 5, we present the results

of applying a tabu search after each of these methods. The neighborhood is defined by the set of solutions that can be obtained by moving an SP from one route to another, swapping two SPs between two routes, inserting an unserved SP into a new or existing route, or turning a served SP into an unserved one. If one of these operations results in an empty route, that route is removed from the solution.

Each entry in the tabu list forbids the insertion of an SP into a particular route and consists of the corresponding SP and route indices. A tabu list entry is created after each operation that removes an SP from a route. For example, after SP i is moved from route r to route s, an entry (i, r) is inserted into the tabu list. Any operation that tries to reinsert SP i into route r is disallowed until this tabu entry expires. The swapping operation removes two SPs from their routes and thus creates two new tabu list entries. For the purposes of the tabu mechanism, the set of unserved SPs is also treated as a route, meaning that an SP that was removed from the set of unserved routes cannot be reinserted into this set until the corresponding tabu entry expires. The length of the tabu list is a parameter of this algorithm. In our numerical experiment, we set this parameter equal to one quarter of the number of SPs. The algorithm is stopped after a predefined time (or a number of iterations), and the best-found solution is returned.

Calculating the objective function for each neighbor requires the following steps: 1. reoptimizing the affected route(s) by solving the corresponding TSP(s) and 2. solving the loading problem for the entire system with the new routes. Each operation performed to generate a neighbor may result in higher (or lower) vehicle and travel costs as calculated in step 1, but step 2 may compensate for these costs by means of a lower (or higher) penalty. To reduce the calculation time, we use several algorithmic enhancements. First, we store the value of the optimal TSP solution obtained for each subset of SPs considered throughout the process in a hash table. Since the tabu search procedure requires the same TSPs to be solved multiple times, this caching mechanism eliminates most of the computational effort that is required for solving TSPs during the tabu search procedure.

Second, we solve (or retrieve from the cache) the TSPs for the routes of all neighbors and sort the neighbors in ascending order of their transportation costs (vehicle and travel costs). The loading problems for the neighbors are then solved in this order. For each instance of the loading problem, we add a constraint that limits the total cost for the current neighbor to be lower than the cost for the best neighbor found so far. Since the neighbors are sorted in ascending order of their transportation costs, this additional constraint renders many instances of the loading problem infeasible, and this infeasibility is quickly detected by the solver. Thus, we reduce the number of computations required to solve each such instance to optimality.

Third, we can save computational effort by skipping the process of solving loading problems for neighbors that satisfy the following conditions:

1) The neighbor is obtained by removing one or two SPs from one or two unsaturated routes, i.e., routes where their travel time and capacity constraints are not binding.
2) The transportation cost improvement relative to the current solution is lower than the best improvement found so far in the neighborhood.

Neighbors that satisfy these conditions cannot be better than the best one found so far. Indeed, removing an SP from an unsaturated route cannot decrease the total penalty cost in the solution because if we could benefit from delivering additional parcels to any SPs on the original route, then the optimal loading solution would saturate the route. Therefore, improvement can stem only from a reduction in transportation cost. Note that the above argument holds regardless of the new route to which the SP is moved.

5 Numerical experiment

In this section, we present the results of a numerical experiment conducted to compare the various solution methods for the FPD problem, and we examine the effect of the degree of flexibility on the delivery cost.

5.1 Experimental settings

All of the proposed heuristic methods were coded in Python 2.7 with CPLEX 12.7 as the MILP solver. The experiments were performed on a system with an Intel i7-6700 4.0 GHz processor with 64 GB of RAM running 64-bit Windows 10.

We generated 27 instances corresponding to nine system configurations (as defined by the numbers of SPs and parcels) and three degrees of flexibility, as specified below:

- Number of SPs (depot included): 20, 40, and 50.
- Average number of parcels per SP: 10, 20, and 30. For example, among the 50-SP instances, there were instances with 500, 1000 and 1500 parcels.
- Level of flexibility: none, low, and high, as described below.

In the nonflexible instances, each parcel had only one desired destination. In the low-flexibility instances, two-thirds of the parcels had only one destination each, while the rest each had two. In the high-flexibility instances, one-third of the parcels had one destination each, another third had two destinations each, and the rest each had three possible destinations. The destinations of the parcels were uniformly selected from the set of SPs. To create some similarity between the instances, the parcels were generated jointly for all three levels of flexibility; three possible destinations were generated for each instance, but only the first one or two destinations were used when applicable.

In all instances, each parcel was characterized as being one of three sizes: large, medium, or small. The parcels were generated such that 20% were large, 40% were medium, and 40% were small. The late delivery penalties were drawn from a geometric distribution with a positive support and a parameter $p = 0.1$ (i.e., mean 10).

The total numbers of lockers in each SP were set to 16 large lockers, 32 medium lockers, and 32 small ones. The fraction of available lockers of each size was drawn from the triangle distribution $TRIA(0.1,0.5,0.75)$, and the result was rounded to the nearest integer. The numbers of large, medium and small truck cells were set to 32, 64 and 64, respectively. The shift length constraint was set to $T = 480$ minutes (eight hours). The travel cost per minute was set to $\alpha = 1$, and the cost for using each additional vehicle was set to $\gamma = 60$. The handling time was set to $\beta = 1$ minute per parcel.

The three sets of SP locations were randomly selected from a list of gas stations in central Israel. Note that automated lockers are commonly located in gas stations. The 20- and 40-SP instances were subsets of the 50-SP instances. The depot was located in Airport City, in close proximity to the main distribution centers of several courier companies. The travel times in minutes between the locations were determined using Google Maps. We ensured that the data approximately satisfied the triangle inequality, except for some rare and minor violations due to rounding errors.

The dataset used in our experiment is available in electronic appendix A.

5.2 Experimental results

In this section, we compare the various solution methods presented in Section 4. We applied the complete MILP model of (1)-(17), the petal heuristic, and the savings heuristic to the 27 test instances. The solutions obtained with both the petal and savings heuristics were improved via tabu search. The time limit for the complete MILP model was set to three hours, which seems practical for a daily operation. Generation of the petals was completed in several minutes, and the solution time for the MILP formulation of (27)-(37) was limited to one hour. The savings heuristic terminated in no more than 32 minutes in the largest instances. Three hours were allocated for the tabu search method, from which the actual time taken in the construction phase (petal or savings) was subtracted.

In Table 2, we report the results of this experiment. In the first column, we present the characteristics of the problem instance in the following format: number of SPs/number of parcels/degree of flexibility. Under "MILP obj.", we present the best integer objective value of a solution to the complete model obtained after 3 hours (and, in parentheses, the result obtained with a 10-hour time limit for the smaller instances). Under "Petal", we present the objective value obtained with the petal heuristic and the corresponding run time. Under "Petal+Tabu", we present the objective value of the solution obtained by applying the tabu search method to the initial solution obtained with the petal heuristic and the corresponding relative improvement. The relative improvement is calculated by subtracting the objective value after the tabu search from the value before the tabu search and dividing by the value before the improvement. In the remaining columns, we present the same information for the savings heuristic.

For each instance, the best obtained solution value is typeset in bold. For cases in which the MILP model of the petal method could not be solved to optimality within the one-hour time limit, the solution time is marked with an asterisk.

Table 2 shows that even in a significantly shorter time, the petal method and the savings heuristic method each reached better solutions than those obtained with the complete MILP formulation in most of the instances that we tested. The advantage of the heuristic construction methods increases as the size of the instance grows. In fact, for all instances with at least 40 SPs and 800 parcels, the result obtained with the complete model was the trivial solution in which no parcels are delivered at all and all penalties are incurred. By contrast, both construction methods are scalable and can be used to obtain high-quality solutions in a short time.

The tabu search method always improved the solution obtained with either construction heuristic within the time limit. The average relative improvement was 4.2%. Thus, if time is available, it is always worth applying the tabu search method.

For the smaller instances (with 20 SPs), we also ran the complete MILP model with a 10-hour time limit. Such a time budget is inappropriate for practical use during daily operations, but we were interested in finding optimal solutions to serve as a benchmark. However, none of these instances reached optimality, and the average and maximal optimality gaps were still 7.8% and 12.7%, respectively, after 10 hours. We noted some improvement in the obtained results, but in all instances in which the heuristic methods yielded better results than the complete MILP formulation under the three-hour time limit, the heuristic methods were still better or equal when 10 hours were allocated for the complete MILP solution.

It is also apparent from Table 2 that adding flexibility to the destinations always reduces the total cost. When we allowed one-third of the customers to choose two destinations and one-third to choose three destinations (high flexibility), we obtained an average cost savings of 15.2% in all instances compared with the nonflexible case

and a cost savings of at least 12% in eight of the nine configurations. These improvements were calculated based on the best solution obtained with our solution methods. These results indicate that flexibility leads to substantially lower total transportation costs and penalties for undelivered parcels. Even low flexibility had a significant effect on the total objective value in all cases. In the low-flexibility case, the average cost savings due to flexibility was 9.3%.

These benefits of flexibility are not particularly sensitive to our random selection of the destination SPs for each parcel. Indeed, in electronic appendix A, we present the results of a similar experiment, with the difference that the destinations of the flexible parcels were selected to be located only near each other. Even in this case, when high flexibility was allowed, there was an average cost reduction of 13.8%, and with low flexibility, the average reduction was 6.9%. These findings suggest that the benefits of flexibility cannot be explained merely by the opportunity to eliminate some regions from the routes and cover only certain regions.

As seen from a comparison of the petal and savings heuristics, neither of them significantly outperforms the other in terms of the objective value of the obtained solution. Moreover, even after applying the tabu search method to improve the solutions obtained with these heuristics, we still cannot identify a dominating approach. However, note that in the larger instances, the MILP model of (27)-(37) could not be solved to optimality within an hour, while the savings heuristic always terminated quickly. In addition, the solution time of the petal method significantly increased with an increasing degree of flexibility, while the solution time of the savings heuristic was not very sensitive to the flexibility. Therefore, we believe that the savings heuristic is more scalable and better suited for instances with high flexibility than the petal method is.

The number of tabu search iterations that could be performed within the time limit decreased from thousands in the smallest instances to only a few iterations in the largest instances. We observed that most of the run time for these instances was spent in solving numerous instances of the TSP and loading problem. The TSPs are solved repeatedly, shift after shift, for the same set of locations. Thus, similar routes are likely to recur. In Section 6, we show that caching the TSP solutions may eliminate most of this time and allow more iterations to be performed within the allotted time.

Table 2: Results obtained within 3 hours

		Petal		Petal+Tabu		Savings		Savings+Tabu	
SP/parcels/flex	MILP obj.	Obj.	Time (sec.)	Obj.	Improv.	Obj.	Time (sec.)	Obj.	Improv.
20/200/none	**563** (560)	570	35	**563**	1.2%	572	25	**563**	1.6%
20/200/low	**502** (502)	570	142	505	11.4%	572	22	505	11.7%
20/200/high	**463** (463)	560	821	483	13.8%	572	23	483	15.6%
20/400/none	715 (659)	671	33	**659**	1.8%	667	25	**659**	1.2%
20/400/low	656 (650)	656	196	**649**	1.1%	656	25	**649**	1.1%
20/400/high	646 (642)	654	924	645	1.4%	650	27	**642**	1.2%
20/600/none	1125 (896)	900	43	**885**	1.7%	897	34	**885**	1.3%
20/600/low	1112	778	246	**767**	1.4%	782	34	**767**	1.9%

	(819)								
20/600/high	897 (768)	755	879	**747**	1.1%	756	38	**747**	1.2%
40/400/none	953	964	116	946	1.9%	1037	113	**920**	11.3%
40/400/low	881	889	545	**879**	1.1%	945	115	901	4.7%
40/400/high	826	814	3690*	**806**	1.0%	896	117	827	7.7%
40/800/none	7811	1456	187	**1376**	5.5%	1521	158	1517	0.3%
40/800/low	7811	1272	1032	**1202**	5.5%	1311	156	1305	0.5%
40/800/high	7811	1208	3689*	1142	5.5%	1272	178	**1141**	10.3%
40/1200/none	11862	2034	376	1960	3.6%	1978	278	**1959**	1.0%
40/1200/low	11862	1731	1332	1673	3.4%	1862	318	**1660**	10.8%
40/1200/high	11862	1769	3700*	1566	11.5%	1531	430	**1509**	1.4%
50/500/none	5168	1229	749	**1143**	7.0%	1184	255	1168	1.4%
50/500/low	5168	1186	3423	1128	4.9%	1121	269	**1103**	1.6%
50/500/high	5168	1058	4236*	**1006**	4.9%	1084	236	1070	1.3%
50/1000/none	10233	1677	844	**1599**	4.7%	1652	325	1611	2.5%
50/1000/low	10233	1553	4235*	**1491**	4.0%	1518	352	1516	0.1%
50/1000/high	10233	1626	4236*	1424	12.4%	1380	369	**1350**	2.2%
50/1500/none	15237	2340	1074	**2308**	1.4%	2509	528	2322	7.5%
50/1500/low	15237	2001	3781	**1955**	2.3%	2103	582	1959	6.8%
50/1500/high	15237	1954	4244*	1868	4.4%	1830	727	**1829**	0.1%

In Table 3, we present the components of the objective function for the best obtained solution for each instance. In the first column, we present the characteristics of the problem instance in the same format as in Table 2. The "Method" column presents the best solution method(s) for the instance. Under "Objective", we present the objective function value of the best solution for the instance. Under "Travel Time", we present the total length of the routes in the solution. Under "No. of Vehicles", we present the number of vehicles used in the solution. Recall that the cost of each vehicle used is 60. Under "Penalty", we present the total penalty for undelivered parcels in the solution. Under "No. of Parcels Delivered", we present the number of parcels delivered in the solution.

Table 3: Results obtained within 3 hours, divided into individual components

SP/parcels/flex	Method	Objective	Travel Time	No. of Vehicles	Penalty	No. of Parcels Delivered
20/200/none	MILP	563	365	2	78	190
	Petal+Tabu		365	2	78	190
	Savings+Tabu		365	2	78	190
20/200/low	MILP	502	362	2	20	196
20/200/high	MILP	463	313	1	90	160
20/400/none	Petal+Tabu	659	477	3	2	398
	Savings+Tabu		477	3	2	398
20/400/low	Petal+Tabu	649	468	3	1	399
	Savings+Tabu		468	3	1	399
20/400/high	Savings+Tabu	642	462	3	0	400
20/600/none	Petal+Tabu	885	533	4	112	554
	Savings+Tabu		533	4	112	554
20/600/low	Petal+Tabu	767	499	4	28	579
	Savings+Tabu		499	4	28	579
20/600/high	Petal+Tabu	747	499	4	8	592
	Savings+Tabu		499	4	8	592
40/400/none	Savings+Tabu	920	593	3	147	376
40/400/low	Petal+Tabu	879	568	3	131	378

40/400/high	Petal+Tabu	806	516	3	110	379
40/800/none	Petal+Tabu	1376	829	5	247	756
40/800/low	Petal+Tabu	1202	809	5	93	778
40/800/high	Savings+Tabu	1141	678	5	163	772
40/1200/none	Savings+Tabu	1959	919	7	620	1041
40/1200/low	Savings+Tabu	1660	925	7	315	1103
40/1200/high	Savings+Tabu	1509	850	7	239	1108
50/500/none	Petal+Tabu	1143	826	4	77	492
50/500/low	Savings+Tabu	1103	724	3	199	456
50/500/high	Petal+Tabu	1006	627	3	199	467
50/1000/none	Petal+Tabu	1599	994	7	185	963
50/1000/low	Petal+Tabu	1491	966	7	105	979
50/1000/high	Savings+Tabu	1350	814	6	176	937
50/1500/none	Petal+Tabu	2308	1118	9	650	1322
50/1500/low	Petal+Tabu	1955	1121	9	294	1396
50/1500/high	Savings+Tabu	1829	1012	9	277	1387

It is apparent from Table 3 that our test instances span a wide variety of instance types, including instances in which all or almost all the parcels are delivered and the penalty is negligible as well as instances in which it is reasonable to avoid delivering up to 20% of the parcels and incur the corresponding penalties.

6 Simulation study of multiperiod settings

In practical settings, the last leg of parcel delivery service is conducted in several shifts per day, each lasting several hours. Parcels that are not delivered in the first shift after their arrival at the regional depot are typically delivered in one of the following shifts, with increased priority. For our flexible delivery model, we conceive of a scenario in which the recipient is notified by a text message once his or her parcel has arrived at the depot and is asked to select a set of possible SPs. The opportunity to select the possible destinations only a short time before the expected delivery enables the recipient to determine locations that are accessible to him or her at that particular time. If, due to capacity constraints, the parcel cannot be delivered during the first shift after its arrival, the recipient is given the opportunity to redefine the set of possible destinations.

In this section, we present a simulation study that demonstrates how our single-period deterministic routing model and solution methods can be used in such a dynamic and stochastic environment. Moreover, we show that our conclusion regarding the benefits of flexible delivery holds in such a realistic environment.

In this simulation experiment, all of the lockers at the SPs were initially empty and available for parcels. Before the beginning of each shift, new parcels arrived at the depot in accordance with a random process. Each new parcel was initially assigned an identical penalty. After the parcels had arrived at the depot, the FPD problem was solved for all parcels currently at the depot subject to the availability of lockers in the system. Based on this solution, parcels were assigned to lockers at the SPs. Next, some parcels were collected from the SPs by their recipients in accordance with a random process. The penalty for each parcel that was not delivered (was left at the depot) was increased, and the set of possible destinations for each of these parcels was redefined before processing for the next shift was initiated.

We simulated 40 shifts of eight hours each. The number of parcels that arrived before each shift was generated from a Poisson distribution with a mean of either $\lambda = 400$ or $\lambda = 800$. We used the 40-SP instances with the geographic locations described in the previous section. Each SP hosted 6 large, 12 medium and 12 small lockers for

the $\lambda = 400$ case, and these numbers were doubled for the $\lambda = 800$ case. The parcels were collected from the SPs by their recipients within zero, one, two, or three shifts after delivery, with probabilities of 0.2, 0.4, 0.2 and 0.2, respectively. We note that most delivery companies allow recipients a certain amount of time to pick up their parcels from an SP. If a parcel is not collected by its recipient within that time, it is collected by the shipping company and returned to the depot, and the sender is informed. The parameters of the arrival and collection processes were selected such that the capacity constraints of the SPs would be likely to be binding in some but not all periods and locations. The penalty for each parcel was initially set to 10 and was increased by 20% after each shift in which the parcel was not delivered. This increase represents the increasing urgency of parcels that are delayed. The remaining settings were exactly the same as in the previous section.

At the beginning of each shift, the FPD problem was solved using the petal and tabu methods with a total time limit of three hours. Thus, the total run time for each simulation was 120 hours. In addition, each optimal TSP route that was found during this process was *cached* and was then retrieved if needed in future iterations or shifts. Moreover, in the petal method, it was necessary to generate the petals only in the first period. Starting from the second period, the petals could be retrieved from the cache, and only the MILP formulation of (27)-(37) needed to be solved.

We created four problem instances with arrival rates of $\lambda = 400$ and $\lambda = 800$. For each arrival rate, we tested the no- and high-flexibility cases, as described in Section 4. Detailed results of the 40-shift simulation of each of the above instances are described in electronic appendix B. The service quality and cost measures calculated from these results are presented in Table 4. The table presents the percentages of parcels delivered during the shift following their arrival and within one shift (first- and second-shift delivery, respectively). The first-shift (second-shift) delivery measure is the ratio between the number of parcels that were delivered during the first (second) shift after their arrival and the total number of parcels that arrived in the system. For the calculation of the second-shift delivery measure, the last shift in the simulation was omitted. Next, the *delivery proportions* were calculated separately for parcels with one, two, and three destinations. The *delivery proportion for N-dest. recipients* is the ratio between the number of parcels with *N* possible destinations that were delivered and the total number of parcels with *N* possible destinations that were available for delivery (note that parcels that were not delivered during the shift after their arrival are counted more than once in the denominator). When flexibility is not allowed, this measure is relevant only for one destination.

The delivery time is the difference between the delivery shift and the first shift after the arrival of the parcel. The average delivery time was calculated based on the delivery times of all parcels that arrived before the first occurrence of a parcel that was not delivered before the end of the simulation. To eliminate bias, all parcels that arrived during the same shift as this parcel or later were omitted. As a result, we omitted only two to six shifts in each of the simulation runs, which suggests that given enough time, each sent parcel is indeed delivered.

Table 4 also presents the total travel time of the vehicles during the simulation and the number of vehicle working shifts, that is, the sum of the number of vehicles used in each shift over the simulation period. We calculated the average cost of delivering a parcel by dividing the transportation cost by the total number of parcels delivered during the simulation. This measure does not include the penalties for the parcels because these penalties represent the service level and not the delivery cost.

Table 4: General measures for the multiperiod settings

Measure	λ=400		λ=800	
	No flexibility	High flexibility	No flexibility	High flexibility
First-shift delivery	84.1%	95.4%	91.2%	98.2%
Second-shift delivery	95.3%	99.5%	98.3%	99.7%
Delivery proportion for 3-dest. recipients	-	99.5%	-	99.98%
Delivery proportion for 2-dest. recipients	-	98%	-	99.8%
Delivery proportion for 1-dest. recipients	80.2%	87%	90.1%	93.9%
Average time to deliver a parcel (shifts)	0.25	0.055	0.107	0.024
Total travel time	32092	23801	41745	31769
Number of vehicle working shifts	133	121	245	224
Average cost of delivering a parcel	2.52	1.95	1.76	1.41

It is apparent from Table 4 that the introduction of flexibility enables significant improvement in the level of service while reducing the costs incurred by the operator.

It can also be seen that with destination flexibility, the delivery proportion for 3-destination recipients is almost 100%, while these proportions are somewhat lower for the 1- and 2-destination recipients. These findings indicate a personal incentive for recipients to show flexibility because it increases their chances of receiving their parcels earlier. Note that with the introduction of flexibility, even the nonflexible recipients enjoy an improvement in their service level, but the impact is much stronger for the flexible recipients.

As expected, a comparison of the instances with low and high arrival rates reveals economies of scale due to the pooling effect of the SPs and opportunities for better consolidation of the parcels in the vehicles. However, the advantage provided by flexibility is still significant, even when the demand and the capacity of the lockers are both relatively high.

In all instances, TSP caching saved some computation time, as seen from the fact that the number of TSPs solved in the last five shifts was decreased by 56.9% on average compared with the number solved in the first five shifts of the simulation. However, this did not translate into a significant improvement in the performance of the tabu search algorithm since the loading subproblems consumed most of the computation time in each iteration.

There are several factors that may affect the time required for the system to reach a steady state if such a state existed. For example, since all lockers are available at the beginning of our simulated scenario, it may take some time until the locker capacity constraints start to be binding. Similarly, during the early shifts, there are few undelivered parcels at the depot. On the one hand, such parcels compete for the available resources, but on the other hand, they create more opportunities for consolidation. Moreover, our solution method may also take some time to warm up since the TSP cache is built gradually. To verify that our experimental results represent a stable system, we plotted the number of parcels in the depot and the first- and second-shift delivery ratios for each shift. These plots are shown in Figures 1-3 for the instances with $\lambda = 800$. Similar figures are provided for the $\lambda = 400$ instances in electronic appendix B of this paper, and these figures lead to the same conclusions. For high-

flexibility instances, the situation seems to stabilize immediately, while the nonflexible instances take some time to reach a steady state. During this warmup time, the first- and second-shift delivery ratios decrease; thus, the service quality presented in Table 4 for the nonflexible instances is biased upwards. This result only strengthens our argument in favor of introducing flexibility.

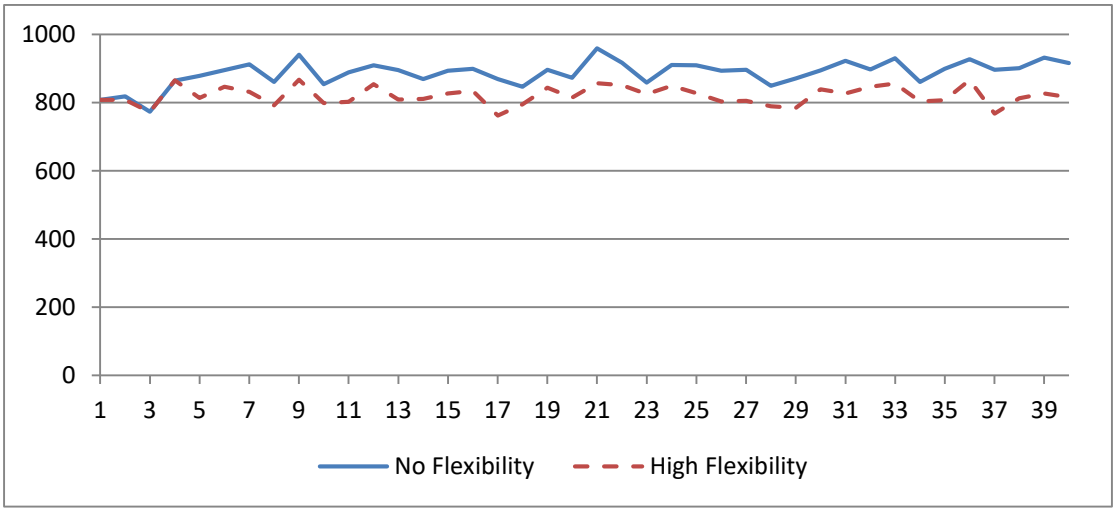

Figure 1: Number of parcels waiting to be delivered at the beginning of each shift for a parcel arrival rate of 800 parcels/shift.

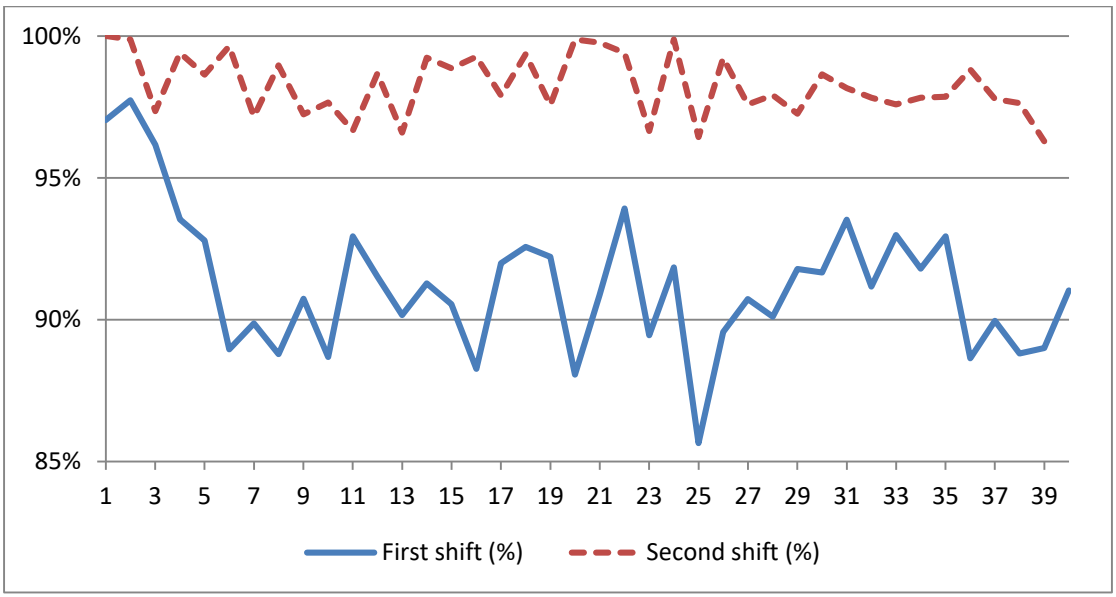

Figure 2: First- and second-shift delivery percentages for a parcel arrival rate of 800 parcels/shift and no flexibility.

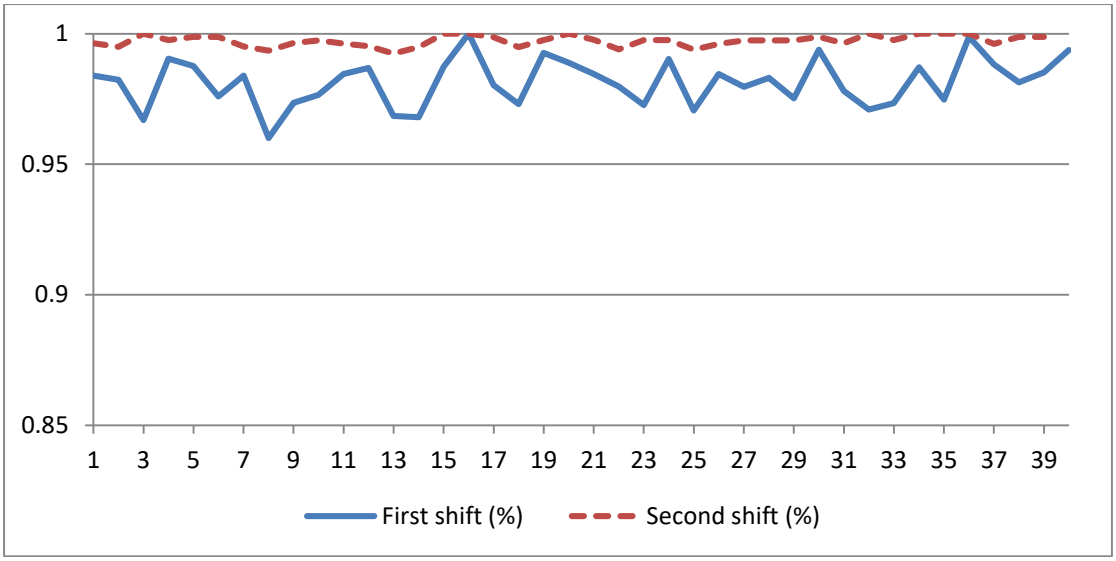

Figure 3: First- and second-shift delivery percentages for a parcel arrival rate of 800 parcels/shift and high flexibility.

7 Conclusions

In this paper, we introduced a logistic model for the delivery of parcels from a single depot to SPs in an APLS in which recipients can choose more than one possible destination SP. We showed that by exploiting this flexibility of the recipients, it is possible to reduce costs and shorten the delivery time significantly.

We formulated the problem as an MILP model and devised effective heuristic solution methods for this model that perform well on large instances, even with high flexibility. Specifically, we introduced two construction heuristics based on the savings and petal methods for the CVRP and an improvement algorithm based on the tabu search framework, in which a very large neighborhood is searched with the aid of an MILP solver.

Both the savings heuristic and the tabu search method are based on repeatedly solving many instances of the TSP and the loading problem. The TSP instances are relatively small, and we can use caching to reduce their solution times. Hence, most of the overall solution time is spent on solving instances of the loading problem. In our experiments, the linear programming relaxation of our formulation always resulted in an integer solution. However, we could not formally prove that the problem is solvable in polynomial time. This limitation raises the theoretical question of whether the loading problem is NP-hard. From a practical perspective, the development of a more efficient solution method for the loading problem will increase the number of tabu search iterations that can be performed within a specified time limit.

Our model was formulated in the context of a single period with deterministic demand. However, through a simulation study, we showed that it could be adapted to a dynamic and stochastic environment. Our experimental results strengthen our conclusion that exploiting the recipients' flexibility makes the delivery process more efficient for the system as a whole and probably also for the individual recipients themselves. Thus, there is a personal incentive for the recipients to show flexibility. Formulating the problem directly in a multiperiod setting would result in a much more intricate model, and it would not be suitable for settings in which the recipients may arbitrarily change their desired destinations if their parcels fail to be delivered during

the current shift. However, such a formulation could create better opportunities for efficient delivery. These will be interesting topics for future research.

Acknowledgment: This research was supported by the Israeli Ministry of Science and Technology.

References

1. Altinkemer, K. and Gavish, B., 1991. Parallel savings based heuristics for the delivery problem. *Operations Research*, 39(3), pp.456-469.
2. Archetti, C., Hertz, A. & Speranza, M. G., 2007. Metaheuristics for the team orienteering problem. *J Heuristics,* Issue 13, pp. 49-76.
3. Archetti, C., Speranza, M.G. and Vigo, D., 2014. Vehicle routing problems with profits. *Vehicle Routing: Problems, Methods, and Applications*, *18*, p.273.
4. Balas, E., 1989. The prize collecting traveling salesman problem. *Networks*, 19(6), pp. 621-636.
5. Biesinger, B., Hu, B. and Raidl, G.R., 2018. A Genetic Algorithm in Combination with a Solution Archive for Solving the Generalized Vehicle Routing Problem with Stochastic Demands. *Transportation Science, 52(3),* pp.673-690.
6. Clarke, G. and Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *12*(4), pp.568-581.
7. Dantzig, G.B. and Ramser, J.H., 1959. The truck dispatching problem. *Management science, 6(1),* pp.80-91.
8. Drexl, M., 2012. Rich vehicle routing in theory and practice. Logistics Research, 5(1-2), pp.47-63.
9. Faugere, L. and Montreuil, B., 2017. Hyperconnected Pickup & Delivery Locker Networks. Proceedings of 4th International Physical Internet Conference, Graz, Austria
10. Feillet, D., Dejax, P. & Gendreau, M., 2005. Traveling Salesman Problems with Profits. *Transportation Science,* 5, 39(2), pp. 188-205.
11. Foster, B.A. and Ryan, D.M., 1976. An integer programming approach to the vehicle scheduling problem. *Journal of the Operational Research Society*, *27*(2), pp.367-384.
12. Ghiani, G. and Improta, G., 2000. An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research, 122(1),* pp.11-17.
13. Golden, B. L., Levy, L. & Vohra, R., 1987. The Orienteering Problem. *Naval Research Logistics,* Issue 34, pp. 307-318.
14. Golden, B.L., Raghavan, S. and Wasil, E.A. eds., 2008. *The vehicle routing problem: latest advances and new challenges* (Vol. 43). Springer Science & Business Media.

15. Goodman, R., 2005. Whatever you call it, just don't think of last-mile logistics, last. *Global Logistics & Supply Chain Strategies, 9(12),* pp.46-51.
16. Hachicha, M., Hodgson, M.J., Laporte, G. and Semet, F., 2000. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research, 27(1),* pp.29-42.
17. Lang, Z., Yao, E., Hu, W. and Pan, Z., 2014. A vehicle routing problem solution considering alternative stop points. *Procedia-Social and Behavioral Sciences, 138,* pp.584-591.
18. Laporte, G., Desrochers, M. and Nobert, Y., 1984. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks, 14(1),* pp.161-172.
19. Laporte, G., Gendreau, M., Potvin, J.-Y. & Semet, F., 2000. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research,* pp. 285-300.
20. Laprote, G. & Martello, S., 1990. The Selective Travelling Salesman Problem. *Discrete Applied Mathematics,* Issue 26, pp. 193-207.
21. Manerba, D., Mansini, R. and Riera-Ledesma, J., 2017. The traveling purchaser problem and its variants. *European Journal of Operational Research, 259(1),* pp.1-18.
22. Miranda, P.A., Blazquez, C.A., Obreque, C., Maturana-Ross, J. and Gutierrez-Jarpa, G., 2018. The bi-objective insular traveling salesman problem with maritime and ground transportation costs. *European Journal of Operational Research, 271(3),* pp.1014-1036.
23. Moin, N.H. and Salhi, S., 2007. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society, 58(9),* pp.1185-1194.
24. Morganti, E., Dablanc, L. and Fortin, F., 2014. Final deliveries for online shopping: The deployment of pickup point networks in urban and suburban areas. *Research in Transportation Business & Management*, *11*, pp.23-31.
25. Raviv, T., Tzur, M. and Forma, I.A., 2013. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, *2*(3), pp.187-229.
26. Reyes, D., Savelsbergh, M. and Toriello, A., 2017. Vehicle routing with roaming delivery locations. Transportation Research Part C: Emerging Technologies, 80, pp.71-91.
27. Reyes, L.C., Barbosa, J.J.G., Vargas, D.R., Huacuja, H.J.F., Valdez, N.R., Ortiz, J.A.H., Cruz, B.A.A. and Orta, J.F.D., 2007, August. A distributed metaheuristic for solving a real-world scheduling-routing-loading problem. In *International Symposium on Parallel and Distributed Processing and Applications* (pp. 68-77). Springer, Berlin, Heidelberg..
28. Ryan, D. M., Hjorring, C. & Glover, F., 1993. Extensions of the Petal Method for Vehicle Routeing. *Journal of the Operational Research Society,* 44(3), pp. 289-296.
29. Song, L., Cherrett, T., McLeod, F. and Guan, W., 2009. Addressing the last mile problem: transport impacts of collection and delivery points. *Transportation Research Record: Journal of the Transportation Research Board*, (2097), pp.9-18.
30. Toth, P. and Vigo, D. eds., 2014. *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

31. Vansteen, P., Souffriau, W. & Van Oudheusden, D., 2011. The orienteering problem: A survery. *European Journal of Operational Research,* Issue 209, pp. 1-10.