Logistics Research (2022) 15:8
DOI_10.23773/2022_8

The Multi-Period Location Routing Problem with Locker Boxes

J. Grabenschweiger · K. F. Doerner* · R. F. Hartl

Received: 28 November 2021 / Accepted: 20 July 2022 / Published online: 29 July 2022

ABSTRACT

When optimal parcel delivery plans have to be made, a logistic system with locker boxes as alternative delivery locations can greatly increase efficiency. An important decision in such a system is where to locate the locker box stations. Hence, we extend the vehicle routing problem with locker boxes to a location routing problem with multiple planning periods. In every period, decisions have to be made about which customers are served at home and which are served at locker boxes as well as how the used delivery locations are routed. On the other hand, the decision of which stations are opened is made only once. The objective function is minimizing total cost, which comprises travel costs, costs for compensating locker box customers, and site costs for realized stations. We provide a mixed integer programming formulation of the problem and propose a metaheuristic solution method. We use self-generated instances to compare the performance of the two approaches and can show that the metaheuristic method yields, on average, very good solutions – in most cases, the optimal solution – in a short computational time. We present and test different strategies for distributing the locker box stations, including based on customer clusters obtained by k-means, which works very well for a variety of geographic customer settings. Moreover, we analyze the utilization of locker box delivery in relation to the length of customer time windows and observe that tight time windows lead to a higher utilization.

KEYWORDS: Locker boxes · Location routing problem · Metaheuristics

Jasmin Grabenschweiger
Josef Ressel Center for Adaptive Optimization in Dynamic Environments, Department of Business Decisions and Analytics, University of Vienna, Austria
E-mail: jasmin.grabenschweiger@univie.ac.at

Karl F. Doerner (*Corresponding author)
Josef Ressel Center for Adaptive Optimization in Dynamic Environments, Department of Business Decisions and Analytics, University of Vienna, Austria
E-mail: karl.doerner@univie.ac.at
Phone: +43-1-4277-37951

Richard F. Hartl
Department of Business Decisions and Analytics, University of Vienna, Austria
E-mail: richard.hartl@univie.ac.at

1. INTRODUCTION

Numerous customers have taken to shopping online everyday in the past years. For delivery companies, this trend is reflected in the large trade volume of parcels. In contrast to mail service, there is usually no infrastructure for receiving the parcel at the customer's home address, such that the parcel can be dropped off in a locked spot when the customer is not at home at the time of delivery. Thus, to guarantee that the customer can receive the parcel in person, the delivery has to occur within a given time window, when the customer is at home. This can lead to costly routing plans with respect to the drivers' working time and the driven kilometers. The last point is especially problematic in urban areas, where the governmental regulations aim to reducing traffic and provide these areas some relief. Moreover, the delivery staff's sometimes extensive workload and bad working conditions are known problems. Their time pressure is often so high that parcels are dropped off in front of the door without the customer's permission, resulting in cases of stolen parcels. At other times, parcels are delivered to random places without notifying the customer or they even get lost on the way. A 2019 six-month survey in the

United States revealed that online order packages of 57% of consumers were left in an unsecured area after delivery. Furthermore, 16% of the surveyed online shoppers reported that their parcels had been stolen (Clutch Logistic Survey (2019)). Such incidents cause inconveniences for the recipients as well as the delivery company or sender of the parcel, who have to cover the damage. These incidents can at least partly be prevented by using locker boxes to store the package safely.

As a response to the challenge of efficiently designing parcel deliveries, new concepts have found their way into current delivery systems. A well established and well accepted concept is that of alternative delivery options, where a customer can be connected with several different locations for receiving parcels. The home address can remain one of these locations, but there may also be pick-up points like shops, post offices, or parcel lockers. Roaming delivery locations, such as the trunk of the customer's car (Reyes et al. (2017)), can also be considered alternative delivery locations.

In this work we consider the vehicle routing problem with locker boxes (VRPLB), which was introduced in Grabenschweiger et al. (2021), and extend it to a location routing problem with a planning horizon of multiple periods. First, we will briefly recap the existing VRPLB. Then, we will motivate and introduce the new model – the multi-period location routing problem with locker boxes (MPLRPLB).

In the underlying model, the VRPLB, one fundamental assumption is that a customer can be served at home during a given time window or at one of the locker box stations they have accepted. A significant advantage of such locker boxes is that the parcel delivery person is not restricted to a specific time window when bringing the parcels there. In general, the alternative delivery options provide great potential to increase the efficiency of route planning. The flexibility in location and time allows for the selection of the most favorable option. In addition, the parcels are stored safely, and customers can usually access the locker box station at any time during the day. However, walking to the station to pick up the parcels may mean extra time and effort; thus, customers receive a compensation payment when they are served at locker boxes. It is assumed that customers accept locker box stations that are within a certain radius of the home address. Another characteristic of the VRPLB is that customers can request more than one parcel of different sizes. Additionally, the locker boxes themselves are assumed to be of different sizes. The capacity of a locker box station is restricted. Thus, only a limited number of each size category is available. To allow for better utilization of the capacity, one customer's parcels can be packed together into one slot.

In the VRPLB, the locker box station locations are fixed. The idea of the new problem, the MPLRPLB, is to allow for optimization with respect to these locations. For this purpose, a set of potential locations is given as input and a decision has to be made about which of these stations should be realized to achieve favorable delivery solutions.

To bring together the operational day-to-day decisions of delivery planning and the strategic decisions of location planning, the problem is considered over multiple periods. We have a complete set of customers for the whole planning period, but not every customer has a request in every period. Some customers may have a demand on multiple days and others on only one. Since the present work focuses on decisions of locate locker box stations rather than future routing, one can assume that the data for the customer patterns and demand scenarios is derived from historical data.

Opening a station comes at some cost, denoted as site cost, which is added to the total cost of the problem. We relate the value of the site cost parameter to the total routing cost of the solution without locker delivery.

The objective function of the MPLRPLB is minimizing total travel cost plus total compensation cost plus total site cost. The costs for routing and compensation are summed up over the number of periods. The costs for realizing locations are charged only once within the planning horizon.

Concerning the capacity at opened locker box stations, we assume that the number of free slots is the same in every period, based on the assumption that the per-period rate of incoming parcels (parcels dropped by the delivery person) is, on average, equal to the rate of outgoing parcels (parcels picked up by the customers). This assumption is based on the fact that customers have a pick-up deadline; within this deadline, lockers are emptied while new parcels arrive to occupy free slots. Parcels that are not picked up within the deadline are sent back to the sender.

A solution to the MPLRPLB is provided by the set of realized stations, the delivery plan containing the information of which customers are served at home and which are assigned to which station, and the routes to visit the used delivery locations. Time windows for home delivery customers and capacity restrictions at stations have to be respected.

For the location planning part of our problem, we can make decisions about the infrastructure of a locker box system. Every established station incurs costs. First, a site cost has to be accounted for setting up the station, which covers the cost of acquiring or renting the required space, buying the parcel locker modules, etc. However, after setting it up, the locker box station's maintenance and operating costs also have to be paid. Thus, running an unnecessary station may mean unnecessary costs. Therefore, the goal is to only realize stations that are necessary to guarantee the system's high efficiency level. The system's degree of efficiency is considerably influenced by the actual locations of the locker box stations. Depending on how they are distributed in relation to the customer locations, different delivery solutions can be obtained. A crucial characteristic that influences this behavior is that customers are assumed to accept locker box stations within a certain radius from their home address. Given a fixed value for the radius, the location of the locker boxes defines how many and which are accepted by a customer. This in turn contributes to the decision of which customers are served at locker boxes and at which station. If locations are realized that are favorable in relation to the customers' geographic setting, bigger improvements may be possible through locker box delivery than with poorly chosen stations. Besides the suitability for customer acceptance, it can also play a role how suitable the stations are located when it comes to integrating them efficiently into the driven routes. That is, visiting remote locker box stations incurs a higher additional distance than visiting those which are closer.

Since the set of potential locations is given as input in our problem, optimization with respect to location decisions can only happen within this scope. Thus, the choice of the input location data is important. Of course, one could simply generate a high number of potential locations to guarantee good coverage of the customer area. However, this comes at high computational cost, since every additional potential location increases the complexity of the model. Hence, we generate only a reasonable number of stations by following certain strategies. With the first strategy, the locations of the potential stations are distributed randomly over the customer area. In the second and third strategies, a grid is laid over the area and locker box locations are put in the center of a grid cell or in a random position within the cell. For the fourth strategy, clusters are built based on the customer locations using variations of the k-means clustering algorithm. The centers calculated within the algorithm are then used as locker box locations. Having introduced the different strategies for distributing stations, we aim to find out which one enables the best solution.

The contribution of this work is three-fold. First, an extension to the VRPLB is introduced – the problem now also contains the strategic decision of optimally choosing the locker box station locations. A mixed integer programming (MIP) formulation is presented for this extended problem. Second, a metaheuristic method is proposed that integrates several operators and heuristics, in addition to the routing operators, for opening and closing stations. Third, different strategies are provided for distributing the potential locations of locker box stations. These aim to generate a good input scenario.

This article is structured as follows: Section 2 reviews the related literature. In Section 3, a detailed description and MIP formulation of the MPLRPLB is given. In Section 4, we introduce a metaheuristic method for the problem. In Section 5, we propose several data generation strategies for distributing the potential locations of the locker stations. In Section 6, we conduct an experimental study where we compare the performance of the metaheuristic method with that of the MIP using self-generated instances. In the experimental study, we also analyze the different data generation strategies and extend standard instances from the literature for these experiments.

2. LITERATURE REVIEW

In this article, we extend the VRPLB to a problem where decisions about the locations of locker box stations also have to be taken.

The integration of alternative delivery options into logistic systems for customer service is still a relatively new branch in vehicle routing. One of the first contributions in this field was given by Reyes et al. (2017). They consider a vehicle routing problem where customers are served at so-called roaming delivery locations. These locations represent the customer's car, which has different parking locations over the day. Thus, the itinerary and possible delivery locations are defined by the customer and optimization of locations is not possible. Ozbaygin et al. (2017) use a set covering approach to formulate the vehicle routing problem with roaming delivery locations. Besides roaming delivery, home delivery is also possible. To solve the model, they came up with a branch-and-price algorithm that allows them to solve instances with up to 60 customers to optimality. In Zhou et al. (2018), the alternative delivery locations in the form of pickup points are modeled as the first level in a two-echelon vehicle routing problem. The second echelon represents delivery to the customer's home address. Thus, a customer can be served indirectly via the first echelon or directly via the second echelon. They do not consider capacity restrictions at the pickup points. In the work of Orenstein et al. (2019), customers are served solely at locker box stations. Service at the home address is not considered. Hence, time windows are also not part of the model. The set of available locker box stations is fixed and using them comes without any additional cost. Thus, no decisions with respect to realizing stations have to be taken. The study of Sitek and Wikarek (2019) provides a vehicle routing problem where service at the home address is one delivery option and alternative delivery options in the form of post offices and locker boxes are also available. Time windows are not considered in their model. In Enthoven et al. (2020), a two-echelon vehicle routing problem with alternative delivery options is studied. In the first echelon, the goods are shipped by a truck either to so-called covering locations or shared delivery locations, such as parcel lockers, or to satellites from where zero-emission vehicles, such as cargo bikes, deliver the parcels to the customers. The customers can specify which delivery option(s) they prefer. Service at a covering location can only be provided if the customer is within a certain range from the location. Capacity restrictions appear for the delivery vehicles (trucks and cargo bikes). A recent work on alternative delivery locations was submitted by Mancini and Gansterer

(2021). They define the vehicle routing problem with private and shared delivery locations. Private locations are locations where only one customer can be serviced, for example the customer's home address. The so-called shared locations are pickup points, such as parcel lockers, where more than one customer can be served. Time windows were considered at the private locations and capacity restrictions at the shared ones. However, in their model, no optimization with respect to the locations of the shared delivery points is done. Another recent contribution to the field of alternative delivery options comes from Grabenschweiger et al. (2021). They introduce the vehicle routing problem with heterogeneous locker boxes. Service can be provided at a customer's private address (but only within a predefined time window) or at parcel lockers. Different sizes are considered for the parcels and the parcel lockers, assuming that one customer's parcels can be stored together in a locker box. This requires integrating a packing problem into the VRPLB. The locker boxes are restricted in size and the locker box stations are capacity restricted. Decisions regarding the locations of the locker box stations are not part of the problem. Dumez et al. (2021) present a problem where each customer has a set of acceptable delivery options, each of which is associated with a preference level and time window. The delivery options can be the home address or shared delivery locations, such as locker box stations or post offices. The shared delivery locations are capacity restricted. In Dragomir et al. (2022) a pickup and delivery problem with alternative locations is studied. The alternative locations come in form of roaming delivery locations in the background of consumer to consumer marketplaces, where the seller's as well as the buyer's itinerary change over the day. Time windows are given for the different locations. Capacity restrictions at the delivery locations are not relevant for this problem.

There is a field of location routing problems, where decisions about the locations of logistic facilities have to be taken jointly with decisions about the routing of customers/delivery locations. Surveys about this problem class can be found in Nagy and Salhi (2007) and Prodhon and Prins (2014).

Concerning location routing problems in the field of alternative delivery locations, the following studies are available. Deutsch and Golany (2018) contribute a work about where to locate locker box stations and how many stations should be realized. However, they do not account for capacity restrictions at the locker box stations, and routing decisions are not part of the problem. Hence, this work can be classified as an uncapacitated facility location problem. They use an objective function that accounts for revenue from customers who use locker box delivery, costs for setting up the stations, and costs for compensating customers who have to travel to the locker box. In Veenstra et al. (2018) a problem is investigated where medical products are delivered to patients either directly to the home location or via locker boxes. Only patients within a certain range from a locker can be served at that locker. Decisions about which locker stations should be opened have to be taken with decisions about how to route the open locker stations and home-serviced patients. When a station is opened, all patients within a certain radius around it have to be served there. Home delivery is then not an option for these patients. The routes for visiting the home service patients are separated from the routes for visiting the used locker stations. Time windows at the delivery locations are neglected. Opening a locker station is related to an opening cost. However, capacity restrictions at the stations are not considered. Schwerdfeger and Boysen (2020) study a problem with mobile parcel lockers, where the locations of the lockers change throughout the planning horizon. The parcel lockers are moved around such that customers have their preferred locker box station close by at least once during the day. The number of required locker box stations should be minimized while guaranteeing service to all customers. Locker box delivery is the only delivery option in this problem. Routing decisions with respect to customers or used locker stations are not needed. Kahr (2022) proposes the stochastic multi-compartment locker location problem where the locations of the locker stations are to be chosen such that expected utility of covered customer demand is maximized (demand is assumed to be uncertain). Furthermore, the configuration of a station has to be decided by combining different locker box modules such that the given budget and space constraints are respected. The modules differ with respect to the number of compartments in each size category; some compartments may have special features, for example storage of frozen products. Routing decisions are not part of this locker location problem and it is assumed that customers are served at locker boxes when possible. Boysen et al. (2021) present a survey about novel last-mile delivery concepts, including cargo bikes, drone delivery, autonomous delivery robots, stationary and mobile parcel lockers, crowdshipping, trunk delivery, etc. They evaluate existing work from the literature, but also see a big need for further research in this area.

To the best of our knowledge, the problem we introduce in this work has not been considered in the literature thus far. We combine the VRPLB with a location problem. Multiple periods are used to connect the operational single-period problems of routing with the strategic location decision. We take time windows for customer home delivery into account as well as capacity restrictions at locker box stations. Customers served at locker boxes receive a compensation payment. The location problem is considered as selecting the best locations from a set of potential locations and the site cost for realized stations is added to the objective function of total cost.

3. PROBLEM DEFINITION AND MATHEMATICAL MODEL

In this part we give a mathematical formulation of the MPLRPLB. The formulation is based on the one given in Grabenschweiger et al. (2021) for the relaxed model with unit-size parcels and locker boxes. In that model, the heterogeneous parcel and locker box sizes are transformed into unit-size values. To check the feasibility of a locker box delivery with respect to station capacity, instead of solving a packing problem, one only needs to check whether the aggregated demand fits the aggregated capacity of a station.

Here we extend Grabenschweiger et al. (2021) to a multi-period setting. There are m customers $C = \{1, \cdots, m\}$ and a customer $i \in C$ is identified via the node $\{i\}$, which refers to the home address. The number of unit-size parcels a customer i demands in period t is denoted by q_{it} , where q_{it} can also be 0 for certain periods, when the customer has no demand in that period. However, we assume that each customer has a demand greater than 0 in at least one period.

We consider a planning horizon of $t = 1, \cdots, T$ periods. Periods can be interpreted as days, weeks, and so on.

An unlimited number of homogeneous vehicles is available to perform the trips. We assume that the vehicle capacity is not restrictive based on the assumption that parcels are small enough for the vehicle to transport the required amount. In the absence of a vehicle capacity restriction, we can give a problem formulation that does not ask for a vehicle index. Thus, it can be omitted as in Mancini and Gansterer (2021).

The set of potential locations for the locker box stations is given by $B = \{m + 1, \cdots, n\}$. The site cost of a locker box station $k \in B$ is denoted by f_k. When station k is opened, it provides a capacity of Q_k. We can assume that the rate of incoming parcels (parcels dropped at a station) per period is approximately equal to the rate of outgoing parcels (parcels picked up from a station). Thus, the number of available slots can be assumed to remain unchanged over the periods.

A single depot exists from where the vehicles leave at the beginning of a trip and where they return to at the end. Furthermore, by C_t , we define the customers that have a positive demand in period t, that is, who have to be served in this period. The home locations of the customers, the potential sites for locker box stations, and the depot nodes form the complete set of nodes for period t, denoted by $N_t = \{0\} \cup C_t \cup B$. Travel distance from node i to node j is given by d_{ij} and the corresponding travel time by t_{ij}. Without loss of generality, we assume that $d_{ij} = t_{ij}$ for all nodes. Travel costs are assumed to be proportional to travel distance. For simplicity, we assume that the cost per distance unit is 1.

For each customer i, there is a possibility of serving him/her at the home address, but only during a predefined time window $[E_i, L_i]$. The service time for the home address is given by s_i. For the depot, we set the time window such that every tour has to be completed within a given maximal tour duration, denoted by D_{max}. The same is done for the delivery time windows of the locker box stations. Thus, the time window of the depot and the stations is assumed to be $[0, D_{max}]$.

In addition to home delivery, a customer can be served at a locker box station. We assume that a customer i accepts all locker box stations located within a certain coverage radius ρ from their home address. The resulting set of accepted locker box stations for customer i are denoted by $B_i \subseteq B$. A compensation cost c has to be paid to a customer who receives locker box delivery.

The routing decision variable x^t_{ij} is binary and is 1 if node j is visited directly after node i in period t; it is 0 otherwise. The binary decision variable y^t_{ik} indicates whether customer i is served at locker box station k in period t. The binary decision variable z^t_i t takes value 1 if customer i is served at home in period t and 0 if this customer is served at a locker box station in period t. The continuous decision variable S^t_i gives the service start time at node i in period t. For the depot node, we define S^t_0 to be the earliest departure time. The binary decision variable w_k is 1 when locker box station k is opened and 0 if it is not opened. When a locker box station is opened, it remains open for the whole planning horizon.

The total cost in this problem is total traveled distance plus total compensation costs for serving customers at locker boxes plus total site cost for the open locker box stations. Traveled distance and compensation costs are operational costs and are summed up over all periods $t = 1, \cdots, T$ to get the total cost for the whole planning horizon. The site cost of an open station is charged only once for the whole planning horizon.

$$\min \sum_{t\in T}\sum_{i\in N_t}\sum_{j\in N_t} d_{ij}x^t_{ij} + \sum_{t\in T}\sum_{i\in N_t}\sum_{k\in B_i} c y^t_{ik} + \sum_{k\in B} f_k w_k \tag{1}$$

subject to

$$\sum_{j\in N_t\setminus\{i\}} x^t_{ij} = \sum_{j\in N_t\setminus\{i\}} x^t_{ji} \qquad \forall t\in T, \forall i\in N_t \tag{2}$$

$$y^t_{ik} \leq \sum_{j\in N_t\setminus\{k\}} x^t_{kj} \qquad \forall t\in T, \forall i\in C_t, \forall k\in B_i \tag{3}$$

$$z_i^t \leq \sum_{j \in N_t \setminus \{i\}} x_{ij}^t \qquad \forall t \in T, \forall i \in C_t \tag{4}$$

$$\sum_{j \in N_t \setminus \{k\}} x_{kj}^t \leq w_k \qquad \forall t \in T, \forall k \in B \tag{5}$$

$$\sum_{k \in B_i} y_{ik}^t + z_i^t = 1 \qquad \forall t \in T, \forall i \in C_t \tag{6}$$

$$\sum_{i \in C_t} q_{it} y_{ik}^t \leq Q_k w_k \qquad \forall t \in T, \forall k \in B \tag{7}$$

$$E_i \sum_{j \in N_t \setminus \{i\}} x_{ij}^t \leq S_i^t \leq L_i \sum_{j \in N_t \setminus \{i\}} x_{ij}^t \qquad \forall t \in T, \forall i \in N_t \tag{8}$$

$$S_i^t + s_i + d_{ij} - M(1 - x_{ij}^t) \leq S_j^t \qquad \forall t \in T, \forall i \in N_t, \forall j \in N_t \setminus \{0\} \tag{9}$$

$$S_i^t + s_i + d_{i0} \leq D_{max} \qquad \forall t \in T, \forall i \in N_t \tag{10}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall t \in T, \forall i, j \in N_t \tag{11}$$

$$y_{ik}^t \in \{0,1\} \qquad \forall t \in T, \forall i \in C_t, \forall k \in B \tag{12}$$

$$z_i^t \in \{0,1\} \qquad \forall t \in T, \forall i \in C_t \tag{13}$$

$$w_k \in \{0,1\} \qquad \forall k \in B \tag{14}$$

$$S_i^t \geq 0 \qquad \forall t \in T, \forall i \in C_t \tag{15}$$

Constraints (2) assure route continuity. Constraints (3) connect the routing decision variables x with the locker box delivery decision variables y. To explain, if a customer i is served at station k (i.e., y_{ik}^t is 1), then this station k has to be included in the routing solution, which is guaranteed by requiring that at least one of the respective arc variables x_{kj}^t has to be 1. Constraints (4) connect the routing decision variables x with the home delivery decision variables z. Similar to before, if customer i is served at the home location (i.e. z_i^t is 1), then the home location has to be visited within one of the routes. Constraints (5) establish the connection between the routing variables x and location variables w. Only opened stations (i.e. w_k is 1) can be used for locker box delivery. Constraints (6) ensure that the demand of a customer is fulfilled by serving him/her either at the home address or at an accepted locker box station. Constraints (7) guarantee that capacity at the locker box stations is not exceeded. Constraints (8) ensure that the time windows at visited locations are respected. Furthermore, they connect the service time variables with the routing variables and are needed for the correct assignment of visited locations to vehicles. Constraints (9) maintain the continuity of service time variables. Through constraints (10 we capture the requirement that vehicles return to the depot within the predefined maximal tour duration. Constraints (11)–(15) define the scope of the decision variables.

4. METAHEURISTIC SOLUTION METHOD

In addition to the presented MIP formulation of the problem, we propose a metaheuristic solution method for solving larger instances of the MPLRPLB, since in Section 6.2 the computational experiments will show that the MIP can only solve instances with up to 25 customers. Here, an outline of the method's building blocks can be seen.

- Construction phase (see Subsections 4.2 and 4.3):
  - Construct a starting location solution with the ADD construction heuristic (Kuehn and Hamburger (1963)). This means, iteratively add an open station until the solution cannot be improved anymore. To evaluate a scenario of open/closed stations in the ADD algorithm, the MPLRPLB is solved by solving a number of single-period problems using the metaheuristic presented in Grabenschweiger et al. (2021). The single-period costs (consisting of routing and compensation cost) are added up over the periods and site cost is added to obtain the solution's total cost.
  - Try to find a better starting selection of open stations through a modified ADD construction heuristic. This heuristic considers in the first iteration not only the best station but also ones that are not the best. The different scenarios of selected stations are evaluated as in ADD.
- Improvement phase (see Subsection 4.4):
  - Starting with the location solution obtained in the construction phase, improvement operators are iteratively applied to the current best location solution. The operators open a closed station, close an open station, and swap an open and closed station. By doing so, a new set of open stations can be analyzed. Therefore, the

single-period locker box routing problems are solved as in the construction algorithms.

The complete algorithm will be summarized in Subsection 4.5. However, first, let us describe the building blocks separately. We start with the single period problem in Subsection 4.1.

4.1. Solving the single-period VRPLB problem

Within the location heuristics and operators described in this section, a number of single-period problems have to be solved. These single-period problems correspond to the VRPLB introduced in Grabenschweiger et al. (2021). The solution method developed and presented within that paper is used to solve the single-period VRPLB. A short summary of the method is given below.

We initialize a solution by solving the problem as a classical vehicle routing problem with time windows (VRPTW), where all customers are served at their home address such that the given time windows are met and total distance is minimized. Adaptive large neighborhood search (ALNS) is used to solve the routing problems within the method.

Starting with this "pure home delivery solution" we apply several operators to evaluate which customers should be served at which locker box stations such that total cost (i.e., traveled distance plus compensation cost) improves.

- The first operator, "reduce distance by assigning customers to locker boxes", simply checks which customers reduce distance the most when they are served at a locker box station instead of at home.
- The second operator, "fill up locker box stations", focuses more on maintaining a low number of used stations and checking which locker box stations are beneficial for use. Therefore, a station is always filled up until no more new customers can be assigned to it. A new station is not used until that criterion is met.
- The third operator, "remove tour", tries removing one complete tour from the routing solution by assigning all customers of that tour to locker boxes or other tours.

The solution obtained after applying one of the described operators then undergoes an improvement phase, where the selection of home delivery and locker box customers is re-optimized. Several improvement operators are used:

- Swapping home delivery and locker box customers.
- Moving one or more locker box customers back to home delivery.
- Moving one or more customers from home delivery to locker boxes.
- Closing a locker box station and reassigning the "free" customers to other stations.

For more details, see in Grabenschweiger et al. (2021).

4.2. Construction heuristic ADD

The ADD construction heuristic is a greedy heuristic and was first presented by Kuehn and Hamburger (1963). It starts with a scenario where all stations are closed (in our problem, this means that all customers are served at their home address). We subsequently open a station until no improvement in total cost is possible. Recall that, besides routing cost, total cost also includes a compensation cost for serving customers at locker box stations and an opening cost for stations. Thus, when we open a station to make additional locker boxes available, we may reduce traveled distance but incur compensation and opening costs.

The first station to be opened is the one that provides the biggest improvement in total cost when exactly one station is opened. Then, we try to add another open station to the already opened one. Therefore, we evaluate the change in total cost when one of the not yet open stations is realized. This is done for each of the not yet open stations, and the best one is opened in addition to the one already open. We add new open stations until the solution cannot be improved further. Note, total cost is always evaluated by solving multiple single-period problems for the scenario-specific set of open locker box stations.

At the end of the ADD construction heuristic, we obtain a set of open locker box stations and a solution for the MPLRPLB. This solution contains the information about how the home delivery customers and used stations are routed and which customers are served at which locker box station in each period. Note that not every open station must be used in every period.

Algorithm 1 provides an outline of the ADD construction heuristic.

Algorithm 1 Construction heuristic ADD

$I = \{1, \cdots, m\}$...set of all locker box stations
$I_1 = \{\}$...set of opened stations
s...solution, s_{best}...best solution
Generate a starting solution s_0, where all stations are closed
$s \leftarrow s_0$, $s_{best} \leftarrow s_0$
while Improvement **do**
  for $k \in I$ **do**
    Evaluate solution with k opened. This yields s.
    if $f(s) < f(s_{best})$ **then**
      $s_{best} \leftarrow s$, $I_1 \cup \{k\}$
    end if
  end for
end while
return s_{best}, I_1

4.3. Modified ADD

The ADD heuristic in its standard form as described in Section 4.2 is a greedy heuristic that takes the best location to be opened in each iteration based on the already selected station(s). In doing so, combinations of stations that perform well when opened together may not be found.

The idea now is to consider not only the best station to be opened in the first iteration of ADD, but the second-, third-, m^{th}-best one and take it as a starting point for the next iteration. This idea can be seen as a best-first search, which is also used in the beam search algorithm (Lowerre (1976)).

If we want to find out all the possible combinations of open stations, we would end up in an enumeration procedure. With this, we would find the best choice of realized stations (best with respect to the heuristic solutions obtained for the single-period routing problems); however, the computational effort would be considerably high, particularly for an increasing number of potential locations.

To keep the computational effort of the modified ADD heuristic on a reasonable level, we make two simplifications. First, we allow a non-best station to be opened only in the first iteration of the algorithm and second, we focus on promising locations when evaluating which combinations should be opened.

We classify a location as promising when it takes a good ranking with respect to solution quality in scenarios where exactly one station is open. These scenarios have already been evaluated when the standard ADD heuristic is executed. In order to avoid redundant calculations, we save the ranking that each station takes in a single station scenario in the standard ADD. The station that leads to the best total cost is the first one in the sorted list of stations. The station that leads to the worst total cost ranks m, where m is the number of potential locations, and goes last in the list.

The idea is then to work with a restricted candidate list. This means we define a list length L that determines how many solutions we consider as promising for being opened in an iteration of the modified ADD heuristic.

In contrast to the standard ADD, in the first iteration of the modified ADD algorithm, we not only allow the best station to be opened, but we also try every location from the set of promising locations as a first open station. For each of these single location scenarios, further open stations are added iteratively in a greedy manner (as in the standard ADD), until new open stations do not lead to a better solution anymore.

We could also execute the iterations after the first one in a non-greedy manner. However, this would increase the computational time considerably. Restricting it only to the first iteration keeps the complexity of the modified ADD manageable. Moreover, we think that this tackles particularly those cases where two or more stations are opened, since a poorly chosen first station may have a stronger effect here.

The best solution is then compared to the solution obtained with the standard ADD heuristic. The better of these two is taken as the incumbent solution and serves as a starting point for the improvement phase, with the operators described in Section 4.4.

The modified ADD algorithm is summed up by Algorithm 2.

Algorithm 2 Modified ADD

$I = \{1, \cdots, m\}$...set of all locker box stations
I_p...set of promising locker box stations
$I_1' = \{\}$...set of opened stations in incumbent solution
$I_1 = \{\}$...set of opened stations in best solution
s...solution, s'...incumbent solution, s_{best}...best solution
Generate a starting solution s_0, where all stations are closed
$s \leftarrow s_0$, $s_{best} \leftarrow s_0$
for $k \in I_p$ **do**
    $s \leftarrow s_0$, $I_1 = \{\}$
    while Improvement **do**
        for $k \in I$ **do**
            Evaluate solution with k opened. This yields s'.
            if $f(s') < f(s)$ **then**
                $s \leftarrow s'$, $I_1 \cup \{k\}$
            end if
        end for
    end while
    if $f(s) < f(s_{best})$ **then**
        $s_{best} \leftarrow s$, $I_1 = I_1'$
    end if
end for
return s_{best}, I_1

4.4. Improvement phase to re-optimize selection of locker box stations

The improvement phase of the complete metaheuristic starts from the best location routing solution found in the construction phase either by the ADD or modified ADD algorithm. Different improvement operators that aim at re-optimizing the current selection of open stations are used. We propose an operator that opens a new station, one that closes an open station, and one that swaps an open station with a closed one. They are all applied in an iterative manner. In each iteration, the used improvement operator is selected randomly and applied to the current best solution. In case the improvement operator gives a selection of open stations that leads to a better solution, this solution is considered the new best solution. To evaluate if the new set of open stations is better, we solve the locker box routing problem for each period – given the changed location scenario – and compare the total cost of the current best solution and the new one. We set an iterator that counts the number of unsuccessful iterations, that is, iterations where the improvement operator did not yield a new best solution. The overall improvement phase ends when the predefined threshold of unsuccessful iterations or a predefined time limit is reached.

The used improvement operators are described below.

4.4.1. Add location

With this operator, we go through the set of not yet open stations and check whether one could be opened such that a better solution to the MPLRPLB is obtained.

4.4.2 Drop location

This operator evaluates whether it is beneficial to close one of the currently open stations. It can only be applied if more than one station is open in the current best solution.

4.4.3. Swap

With the swap operator, we open a currently closed station and close a currently open one. Here, we work with promising solutions again to reduce the number of swaps to be evaluated. Thus, locations currently not in use and identified as the least promising in a single station scenario are not considered for a swap. This can be considered a restricted candidate list, often used in local search procedures.

The swap operator is applied following a best-improvement strategy, that is, we evaluate all potential swaps and take the best solution (provided there is an improving solution at all).

4.5. The complete metaheuristic method

Algorithm 3 provides a pseudocode for the complete metaheuristic.

Algorithm 3 Metaheuristic solution method for the MPLRPLB

$I = \{1, \cdots, m\}$...set of all locker box stations
I_p...set of promising locker box stations
$I'_1 = \{\}$...set of opened stations in incumbent solution
$I_1 = \{\}$...set of opened stations in best solution
s...solution, s'...incumbent solution, s_{best}...best solution
Generate a starting solution s_0, where no locker box delivery is used; that is, solve the VRPTW for each period by ALNS; sum up period routing costs
$s_{best} \leftarrow s_0$
Initialize a set of open stations by ADD, yielding solution s.
Given a set of open stations in the different iterations of ADD, the single-period problems are solved by the corresponding algorithm and the resulting routing, locker box compensation, and site costs are added up.
if $f(s) \leftarrow f(s_{best})$ **then**
  $s_{best} \leftarrow s$
end if
Starting with s_0, try to find a better set of initial open stations by modified ADD, yielding solution s.
For the iterations of modified ADD, the different location scenarios are evaluated by solving the single-period VRPLB and comparing total cost.
if $f(s) \leftarrow f(s_{best})$ **then**
  $s_{best} \leftarrow s$
end if
while limit on number of iterations without improvement or time limit not reached **do**
  Choose an improvement operator randomly.
  The changed selection of open stations obtained with the improvement operator is evaluated by solving the single-period problems. This yields solution s.
  if $f(s) \leftarrow f(s_{best})$ **then**
    $s_{best} \leftarrow s$
  end if
end while
return s_{best}

5. DATA GENERATION STRATEGIES

In the following, we propose different strategies to be used when possible locations for locker box stations have to be generated. The way the possible locations are distributed may influence solution quality considerably. We want to determine which strategy should be used to generate data such that the optimization potential with respect to locker box delivery and location selection can be exploited well.

This is related to the work by Grabenschweiger et al. (2018), where different strategies for locating the so-called optional nodes were tested to determine which one leads to good solutions. The "random" and "grid" strategies are considered here in a slightly adapted form. In addition, we consider k-means clustering algorithms for locating the nodes of locker box stations.

When the proposed data generation strategies are applied to real-world instances where the customer nodes come from a real map, the outcome may suggest spots for the lockers that do not fit the street layout. In this case, one could look for the closest feasible location for a locker station. There might also be more sophisticated approaches that model the problem of finding good locker station locations, in which you could incorporate additional attributes such as the availability of nearby parking lots. However, this is beyond the scope of our article. We aim to provide concepts for generating locations. In Subsections 5.1 to 5.3, we propose three such general strategies, a purely random distribution of locker boxes, locating them based on clustering of the customers, and a grid structure to ensure a rather geometric distribution of locker locations. These concepts can also be adapted to real world situations.

5.1. Random distribution of locations

With this strategy, the potential locations are chosen randomly within the relevant area. Implementing it in the process of instance generation is simple. However, the potential savings generated by using locker boxes may be restricted when the locations are distributed poorly. This may happen if there is no locker box available in a region that would be favorable in terms of locker box delivery.

5.2. Distribution of locations on a grid structure

Another strategy is to put a geometric grid structure over the customer area. The area is divided into grid cells and locations are distributed such that a potential

Algorithm 4 k-means algorithm Lloyd (1982)

```
1: x_1, ..., x_m...data points; customer locations
2: k...number of locker box locations
3: Initialization: Randomly select centroids μ_1, ..., μ_k
4: Initialization: Clusters S_1, ..., S_k → ∅
5: repeat
6:     Assign each data point to closest centroid
7:     for i = 1, ..., m do
8:         j ← argmin_{j*} ||x_i − μ_{j*}||^2
9:         S_j ← S_j ∪ {i}
10:    end for
11:    Update centroids
12:    for j = 1, ..., k do
13:        μ_j = 1/|S_j| Σ_{i∈S_j} x_i
14:    end for
15: until Clusters do not change or maximum number of iterations is reached
16: return Clusters S_1, ..., S_k and centroids μ_1, ..., μ_k
```

location is available in each cell. For "random grid", the location is put on a random spot within a cell. For "fixed grid", the location is inserted in the center of a cell. This gives the most uniform distribution of locations across the map.

5.3. Distributing locations by k-means

Here, the locations of the locker box stations are chosen based on k-means clustering. In general, k-means clustering aims at dividing given data points into k clusters, such that each data point is assigned to the cluster that has the closest mean. The mean of a cluster also called a centroid.

In our problem, the data points are the coordinates of the customer locations. The parameter k is the number of potential locker box stations we want to generate. Then, k clusters are computed and a potential locker box is located in the centroid of the corresponding cluster.

In general, k-means clustering belongs to the class of NP-hard optimization problems. However, there are several approximation algorithms that work simply and efficiently.

The k-means algorithm of Lloyd (1982) is often considered the standard version of k-means algorithms. An outline can be seen in Algorithm 4.

In the k-means algorithm of Lloyd (1982), the centroids are initialized randomly, which is easy to implement. However, studies have shown that the initialization of the centroids is crucial, since the iterative computation of the clusters starts from there, and the clustering result may be different depending on how the seeds are chosen. In a modified version, called k-means++ (Arthur and Vassilvitskii (2007)), the idea is that the starting centroids are chosen with respect to some dissimilarity measure. The rest of the algorithm works as stated in Algorithm 4. Hence, only line 3 changes.

To explain, the initialization in k-means++ is done as follows: From the input data points, one is chosen randomly to serve as the first centroid. Then, new centroids are added iteratively until the number k is reached. In each iteration, a data point is chosen that is dissimilar to any of the existing centroids (where dissimilarity is measured as being far away). This can be done in two ways. First, the data point farthest away from its closest centroid is taken. Second, some more randomness is included by defining the probability that a data point is chosen to be proportional to its distance to the closest existing centroid. In particular, this means that the higher the minimum distance of a data point to its closest centroid, the more likely it is that this data point is chosen.

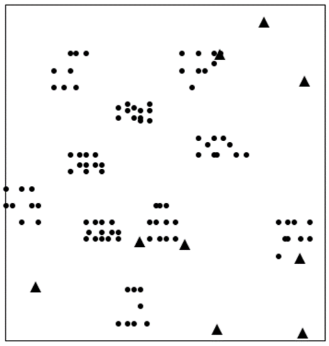
So, we have in total three variants for initializing the centroids. The first corresponds to the standard k-means and randomly selects the initial centroids. The second and third correspond to k-means++ and work with a dissimilarity measure based on the chosen centroids.

Independent of the initialization strategy used, the rest of the clustering algorithm is the same. That is, after initialization of the centroids, we assign points to clusters and update centroids as described in Algorithm 4.

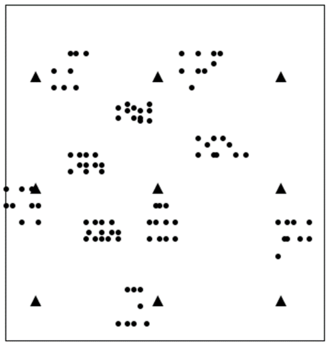
Since some randomness is included in each of the three centroid initialization mechanisms, we are likely to obtain different clustering solutions in different runs. Thus, the idea is to run each mechanism for 50 iterations and take over all mechanisms and iterations, the centroids of the clustering solution with the best objective value as potential locations for locker box stations. The objective function for the clustering problem is the minimization of the sum of squared distances from the data points to their closest centroid.

5.4. Graphical illustration of the data generation strategies

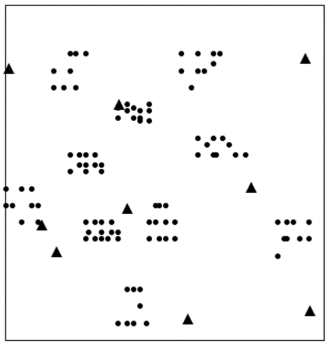
Figure 1 shows a possible result when the presented strategies are applied to an example instance of Solomon's instance set C1 (Solomon (1987)). The nodes are depicted in a square of size 100 × 100.



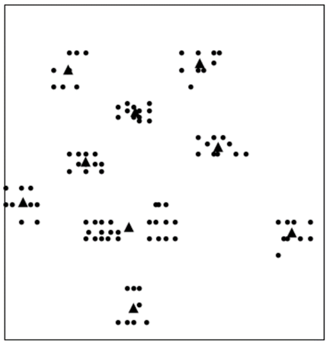
(a) random



(b) fixed grid



(c) random grid



(d) k-means

Fig. 1: Different strategies for generating locations of potential locker box stations. Circles represent customer nodes and triangles represent locker box locations.

One can see in Figure 1a that, with "random", no locker box station is located in the top left corner, while two locations are generated in the top right and bottom right corners, although no customers live there. With "fixed grid", a very uniform, geometric distribution of locker box locations is obtained that covers the whole area. With "random grid", we have one station in every grid cell; however, two stations could be generated very close together when they are located next to the same cell border. With the two grid strategies, we have a better coverage of the whole area. However, the actual customer locations are not considered, which also leads to station locations quite far away from the customer clusters. With the "k-means" strategy, the geographic structure of the customer distribution flows in through the algorithm itself. This can be clearly seen in the corresponding Figure 1d, where the locker box locations are within or between clusters of customers.

6. COMPUTATIONAL STUDY

The models and methods are coded in C++. The exact models are solved by CPLEX 12.9 (multithreading is switched off). All tests are executed on an Intel Xeon Processor E5-2670 v2 (25M Cache, 2.50GHz) with a 3GB RAM limit. Linux is the operating system.

6.1. Compare different strategies for generating locations of locker boxes

We first compare the outcome of the various strategies for distributing locker box stations described in Section 5.

6.1.1. Test instances

To test the different strategies, we use a set of instances generated based on the instances from Solomon (1987).

Therefore, the spatial, time, and demand information of the customer nodes is taken as in the original set from Solomon (1987). Regarding the number of demand periods, we face a single-period problem, where all given customers require service. The reason we work with the Solomon instances here is that they feature a special distribution structure of the customer nodes and with respect to the time windows. A complete instance contains 1 depot and 100 customers. The customer locations are distributed within a [100 × 100] square. The instances are grouped according to their characteristics, and the groups are named C1, C2, R1, R2, RC1, and RC2. Each group comprises 8–12 instances. In sets C1 and C2, the customers' locations show a clustered structure. In R1 and R2, the customers are distributed randomly. In RC1 and RC2, half the customers are located in clusters and the other half are located randomly. In C1, R1, and RC1, the time windows are rather narrow, while in C2, R2, and RC2 they are broader.

Within one instance set, the geographic data of the nodes remains unchanged. Thus, for example, in instance C101 to C109, the customers' coordinates are the same. This holds true for the demand data. The only difference in customer information comes from the time windows.

In order to create the locations of the potential locker box stations for our instances, we use the data generation strategies introduced in Section 5. We consider a set of 9 potential locker box sites. This is because 9 is obtained when using a 3x3 fixed grid as described in Section 5.2. In order to have a fair comparison, we choose the same number of locker box stations also when using the random and clustering strategies as described in Sections 5.1 and 5.3.

Another decision made in the application of the data generation strategies to the set of Solomon instances, is not to take the full [100 × 100] square, but to narrow down the potential distribution area for locker stations for each instance to the area where customers of this instance are actually located. This guarantees that the grid spans only the relevant area, thus bringing the center points of the cells closer to the customers. Furthermore, for the strategy "random", it makes sense to avoid locating locker sites in areas where no customer can be found. When the locations are distributed by k-means, the algorithm itself drags the cluster centroids to the relevant customer area.

Note, when we extend the Solomon instances by the locker box information, the locations obtained by "fixed grid" and "k-means" are the same within the instances corresponding to the same instance set (e.g., within all instances belonging to C1) because the customer locations are the same. For "random" and "random grid", the locker box locations are different in every instance.

For each instance, the distance matrix is obtained by calculating the distances between two nodes, which is calculated based on Euclidean metrics. The time window of the depot represents the maximal duration of a route. The time window of a locker station is assumed to be the same as the time window from the depot. For the service time of a locker box station, we take twice the service time given for the customers. A customer accepts service at the home address or at locker box stations located within a predefined radius ρ from the home address. We choose this parameter to be 30, since it has proven to provide a good measure in relation to the overall geographic setting.

For the locker box stations, we assume that there is a unit-size capacity of 250 slots available in each period. Within one instance, the site costs are the same for each station. Over the whole set of instances, this parameter is not fixed, but is chosen dependent on the solution cost without using locker boxes. During the computational experiments in this section, we observed that the site cost parameter has a strong influence on locker box utilization, or in particular, on how many stations are opened. If this parameter is too high, in the extreme case, no locker box stations are opened; if it is too low, too many stations are opened, making the location selection problem not really a problem of selection. We could not find a fixed site cost value such that a reasonable number of stations is used for all instances. The reason for this lies essentially in the fact that there are significant differences in the objective function values of the instances. Consequently, we decided to choose the site cost for an instance in relation to the solution cost without locker box utilization. For the instance sets R1, R2, RC1, and RC2, we came up with 5% as a proper ratio with respect to solution cost. Thus, for an instance with solution cost 1000 in absence of locker boxes, the site cost value would be 50. For the instances of sets C1 and C2, we have to reduce the site cost percentage to 4% in order to achieve at least one open locker box station in all cases.

The parameter c that represents the compensation cost for serving a customer at a locker box is set to 5, as suggested in Grabenschweiger et al. (2021). A change for this value is made for instance set C2, where we set $c = 3$. This is necessary because – even with an already reduced site cost percentage – we do not achieve a meaningful utilization of locker boxes. For the instances with clustered customer regions (i.e., C1 and C2), it seems that using locker boxes does not provide such strong efficiency effects as it does for the other instances. Thus, the cost parameters have to be lower to obtain a comparable use rate of locker boxes. This may be traced to the geographical structure of the C1 and C2 instances, since customers in clusters are easier to serve in efficient routes, and taking only some customers out of a route to serve them at locker boxes does not lead to high enough cost reductions. Another aspect that obviously influences the attractiveness of locker box delivery is the length of the time windows. This conclusion emerged in the parameter tuning phase, as we can see in the choice of a lower compensation cost value for the clustered instances with wider time windows (C2) compared to the clustered instances with tighter time windows (C1). This observation is also plausible, since home delivery is possible only during the given time windows and the tighter time windows are, the harder it is to pack customers feasibly into delivery routes. With the possibility of serving customers at locker boxes, significant savings with respect to routing costs may be achieved. Conversely, for the C2 instances, locker box delivery may not have as great potential for cost savings, since planning customers in efficient home delivery routes is easier when time windows are wider.

6.1.2. Settings for the construction and improvement phase of the metaheuristic method

The instances in this part of the computational study contain 100 customers and 9 locker box stations. Comparing the MIP and metaheuristic methods in the next section will validate that we can not solve instances of this size by MIP. Hence, we use the proposed metaheuristic. However, instances of this size mean a high computational effort even for the metaheuristic. One characteristic that drives this behavior is, for example, the number of locker box stations – since an increasing number of stations makes parts of the algorithm more and more complex. The more stations we have, the more scenarios of open/closed stations have to be evaluated in the construction phase as well as in the improvement phase. Moreover, every change made in the scenario of open stations requires a call to the algorithm for the one-period problem. Thus, first, we restrict the algorithm for the one-period problem to a time limit of 10 minutes. All parameters used in the the algorithm for the one-period problem are set as in Grabenschweiger et al. (2021). For the ADD heuristic in the construction phase of the location planning algorithm, we do not impose any runtime restriction. For the modified ADD heuristic, which works with a restricted candidate list, the scenarios with a single open station are sorted in a list by increasing order of total cost. The number of promising stations, that is the position in the list up to which we consider stations for evaluation in the first iteration of the ADD algorithm (length L of the restricted candidate list), is determined by multiplying the list length by a random factor from the range [0.4, 0.6]. For the stations that may be added to the already open station, we take a list length L that is determined by multiplying the length of the whole sorted list by a random factor from the range [0.7, 1.0]. For the improvement phase of the location planning algorithm, we set the iterator that counts the number of unsuccessful iterations to 10. Additionally, we set a time limit of 1 hour for the improvement phase, in case the iterator was not hit within that time. With these configurations of the overall algorithm, we achieve, on average, a runtime of about 6.5 hours.

6.1.3. Computational results

The results are given in Table 1 to Table 6, each table belonging to one of the instance classes C1, C2, R1, R2, RC1, and RC2.

The first column shows the instance name, with the first part of the name referring to the original instance name from Solomon (1987). The middle part of the name refers to the number of customers, and the last part references the number of potential locker box stations. Then, the average total cost values over 5 runs are reported for each strategy and instance. A value in italics means that this strategy performed, on average, the worst for the respective instance. A bold value means that this strategy performed best, on average, over the 5 runs. The last three rows of each table give performance measures for each strategy aggregated over the whole instance set. The row "#best" reports how often a strategy was best in the respective instance set and the row "#worst" shows how often a strategy led to the worst result. The row "avg." states the average values over all instances of the set for each strategy.

We can see that for the clustered instances C1, the k-means clustering strategy clearly dominates the other strategies. It yields the lowest average total cost of 800.3 and is never the worst strategy, but the best strategy in 6 out of 9 cases. "Fixed grid" works very poorly, having a significantly higher total cost of 819.6, never the best strategy, but the worst one in 6 out of 9 cases. For the clustered instances C2, "k-means" performs very well. It yields the lowest average total cost of 568.5 and is the best strategy for half the instances and only once the worst.

instance	random	fixed grid	random grid	k-means
C101_100_9	*821.5*	820.8	818.7	**799.7**
C102_100_9	816.5	*819.3*	805.8	**790.4**
C103_100_9	**784.5**	*817.0*	803.3	807.9
C104_100_9	807.6	*810.7*	797.6	**795.7**
C105_100_9	*823.1*	820.0	**785.5**	803.2
C106_100_9	806.2	823.2	*829.8*	**801.6**
C107_100_9	793.5	*817.5*	**786.7**	796.4
C108_100_9	809.4	*828.4*	813.1	**807.7**
C109_100_9	804.8	*818.7*	779.3	**770.3**
#best	1	0	2	6
#worst	2	6	1	0
avg.	807.8	819.6	805.1	**800.3**

Table 1: Comparison of location generation strategies for the instance set C1

instance	random	fixed grid	random grid	k-means
C201_100_9	**572.8**	579.0	580.0	*580.7*
C202_100_9	**546.4**	*577.9*	558.0	575.8
C203_100_9	569.7	580.5	*597.7*	**545.9**
C204_100_9	*583.6*	580.1	569.2	**563.0**
C205_100_9	582.0	581.2	*584.7*	**568.8**
C206_100_9	*589.5*	580.7	**570.3**	571.5
C207_100_9	583.3	579.2	*587.1*	**566.5**
C208_100_9	**560.7**	*580.1*	574.6	575.6
#best	3	0	1	4
#worst	2	2	3	1
avg.	573.5	579.8	577.7	**568.5**

Table 2: Comparison of location generation strategies for the instance set C2

"Fixed grid" is also the worst strategy for the C2 set, since it leads to the highest objective value of 579.8 and is the best strategy for none of the instances in this set. For the C1 instances, we have an average runtime of 5.6 hours. For the C2 instances, it is 5.4 hours.

For the random instances R1, the random grid strategy works best on average (total cost 938.8), with only a small gap to the clustering strategy (939.2). Moreover, the clustering strategy is never the worst strategy in the set. Distributing the locker box location by means of fixed grid gives very poor results for R1: average total cost of 979.2 and never the best strategy but the worst in 9 out of 12 cases. For the random instances R2, the clustering strategy works best on average (total cost 780.1), but "random grid" is second (781.3) only with a small gap to the leading strategy "k-means". The reason the random grid strategy performs quite well for a randomly distributed customer set may lie in the number of potential stations. The more stations we have to generate, the finer the grid structure becomes and, together with the random part, this may yield on average a quite good coverage of the customer area. The average runtime of the R1 instance set is 7.2 hours; for the R2 set, it is 6.4 hours.

instance	random	fixed grid	random grid	k-means
R101_100_9	**1029.3**	1107.6	*1125.0*	1079.1
R102_100_9	1003.0	*1066.3*	**979.0**	1020.4
R103_100_9	*1048.6*	976.9	**920.5**	963.9
R104_100_9	906.9	*909.3*	**851.1**	858.5
R105_100_9	**937.2**	*1061.3*	1000.0	1015.0
R106_100_9	1005.7	*1014.1*	985.7	**973.4**
R107_100_9	893.2	*946.8*	**877.9**	923.4
R108_100_9	**812.4**	*897.6*	896.2	844.6
R109_100_9	981.4	965.1	*1004.1*	**929.7**
R110_100_9	944.8	*954.9*	**894.2**	908.7
R111_100_9	908.3	*951.9*	896.7	**896.3**
R112_100_9	839.2	*898.5*	**834.6**	857.4
#best	3	0	6	3
#worst	1	9	2	0
avg.	942.5	979.2	**938.8**	939.2

Table 3: Comparison of location generation strategies for the instance set R1

instance	random	fixed grid	random grid	k-means
R201_100_9	929.0	*948.0*	**897.5**	916.1
R202_100_9	879.4	*880.3*	**825.7**	827.5
R203_100_9	775.6	*785.2*	**764.7**	779.7
R204_100_9	701.6	*720.5*	701.2	**694.3**
R205_100_9	857.3	*871.8*	**812.4**	839.8
R206_100_9	**769.7**	805.6	*831.5*	784.4
R207_100_9	**728.1**	752.2	*766.3*	749.7
R208_100_9	**679.1**	696.3	*698.7*	679.7
R209_100_9	**745.3**	*810.3*	777.9	787.2
R210_100_9	*861.1*	821.6	**793.2**	798.2
R211_100_9	725.2	*747.9*	724.7	**724.2**
#best	4	0	5	2
#worst	1	7	3	0
avg.	786.5	803.6	781.3	**780.1**

Table 4: Comparison of location generation strategies for the instance set R2

For the random-clustered instances RC1, the fixed grid strategy leads to the best average total cost value of 1132.5. However, "k-means" is close to it with an average total cost value of 1135.1 and also regarding the counts of how often it was best and worst. Distributing the potential locker box stations purely randomly gives very bad results for the RC1 instances. For the RC2 instances, the clustering strategy dominates the others – it shows the lowest average total cost of 926.8. Further, it is never the worst strategy and in half of the cases, it is the best one. The instances of set RC1 need on average 7.2 hours computational time and the instances of RC2 on average 6.2 hours. Concerning runtime, we can observe that over the instance sets, the ones with wider time windows (C2, R2, RC2) have a lower runtime than those with tighter time windows (C1, R1, RC1), which can be justified by the fact that problems with tight time windows are, in general, harder to solve.

instance	random	fixed grid	random grid	k-means
RC101_100_9	*1404.7*	**1237.7**	1257.1	1292.6
RC102_100_9	*1221.1*	1183.6	**1160.6**	1202.0
RC103_100_9	*1149.2*	1129.6	1134.3	**1092.1**
RC104_100_9	1022.2	1002.4	*1048.0*	**995.5**
RC105_100_9	*1447.4*	**1205.4**	1330.1	1227.9
RC106_100_9	*1278.1*	1168.7	**1063.9**	1184.9
RC107_100_9	*1184.4*	1093.0	1113.0	**1081.7**
RC108_100_9	1034.2	*1039.8*	**993.0**	1004.1
#best	0	2	3	3
#worst	6	1	1	0
avg.	1217.6	**1132.5**	1137.5	1135.1

Table 5: Comparison of location generation strategies for the instance set RC1

Overall, we conclude that locating potential locker box stations by clustering through the k-means algorithm is the most suitable and stable strategy, since it is clearly the best strategy for 3 of the 6 instance sets, and not much worse than the best strategy or slightly the best strategy in the other three 3 cases. So, it seems that with the clustering approach for locker box nodes, a variety of geographical characteristics at the side of customer nodes can be captured.

We have previously mentioned that, when we had to decide the cost parameters for the C1 and C2 instances, we observed how tighter or wider time windows influence the attractiveness of locker box service (attractiveness from a routing planner's perspective). This observation is supported by the numbers given in Table 7.

The column "#LB customers" states the number of customers served at a locker box (out of the 100 customers; the remaining ones are served at their home

instance	random	fixed grid	random grid	k-means
RC201_100_9	1076.7	*1100.0*	**1067.4**	1096.2
RC202_100_9	996.1	**972.8**	*1008.1*	977.3
RC203_100_9	871.1	881.0	*901.3*	**868.7**
RC204_100_9	*776.2*	756.0	753.3	**745.4**
RC205_100_9	*1058.4*	1043.7	1052.6	**1026.6**
RC206_100_9	975.8	*997.5*	**955.3**	991.1
RC207_100_9	**896.3**	*940.0*	910.3	921.2
RC208_100_9	800.1	*802.0*	797.9	**787.8**
#best	1	1	2	4
#worst	2	4	2	0
avg.	931.3	936.6	930.8	**926.8**

Table 6: Comparison of location generation strategies for the instance set RC2

	# LB customers	# open stations
C1	27.5	1.4
C2	30.2	1.5
R1	71.9	2.8
R2	50.9	2.1
RC1	59.1	2.7
RC2	42.3	2.1

Table 7: Locker box utilization

address). The value is the average over all instances from the respective set. The column "#open stations" reports the average number of stations to be opened and used for locker box delivery. For the R and RC instances, we can see that for the instance sets with wider time windows (R2 and RC2), on average less customers are served at locker boxes and less stations are opened compared to the instance set with tighter time windows (R1 and RC1). To explain, in the R2 instances, 50.9 customers on average are served at locker boxes and 2.1 stations are used, compared to the R1 instances, where 71.9 customers are served on average at locker boxes and 2.8 stations are used. In the RC2 instances, we have 42.3 locker box customers on average and 2.1 used stations, compared to the RC1 instances with 59.1 locker box customers and 2.7 used stations. An explanation for this outcome has been given previously when justifying a lower compensation cost value for C2. For the instances R1 and R2, the comparison of these two measures is valid, since cost parameters are the same for both sets. The same holds true for sets RC1 and RC2. For C1 and C2, the values are not significantly different, but comparison in this case is difficult since different values for the compensation cost parameter were used.

6.2. Compare methods: MIP and metaheuristic solution method

We now want to use the MIP presented in 3 and the metaheuristic solution method presented in 4 to solve a set of self-generated test instances and to compare their performance with respect to solution quality and computational time.

6.2.1. *Test instances*

In the self-generated instances all nodes are distributed within a square of size [30 × 30]. The depot node is located in the middle of the southern edge. The customer nodes are distributed randomly across the area. Instances with 10, 12, 15, 17, 20, 22, and 25 customers are generated. The experiments in the preceding section have shown that, on average, the k-means strategy works best for distributing the locker box locations. Thus, we use this strategy here. Instances with 5 potential locker box stations are generated. Distances between two nodes are calculated based on Euclidean metrics. The time window for the depot node is set to [0, 720]. This implies that a maximum tour duration of 720 minutes (12 hours) is assumed. The same time window is set for the locker box stations. The length of the customers' time windows is assumed to be 1 hour, and a time window can only start at clock hour. The service time of the customers is fixed to 9 minutes and that of the locker box stations to 20 minutes.

For the customers, demand data also need to be generated. We assume that each customer has a positive demand in at least one period. Service in more than one period or in all periods may also be required. The decision of whether a customer has a demand in a certain period is taken based on a probability distribution, which gives “yes demand” with 70% probability and “no demand” with 30% probability. Consequently, on average, about 70% of the customers in each period show a positive demand and have to be served. When a customer has a demand in a certain period, a demand of 1 parcel appears with 70% probability, a demand of 2 parcels with 20% probability, and a demand of 3 parcels with 10% probability. Service can happen at the home address or at an accepted locker box station. We assume that a customer accepts all stations located within a predefined radius ρ from the home address. We choose this parameter to be 10, since this value is one-third the value 30 from the extended Solomon instances, preserving roughly the relation between the ranges of the nodes coordinates (30 to 100). For the locker box stations, we assume that there is a unit-size capacity of 25 slots available in each period. We set the site cost of each station to 20. The computational experiments have shown that with this value, we obtain a reasonable solution with respect to locker box utilization. The parameter c, which represents the compensation cost for serving a customer at a locker box, is set to 5 as suggested in Grabenschweiger et al. (2021).

We generate instances for a problem with 1,2, and 3 periods, respectively.

Depending on the availability of data or on the purpose of the problem to analyze, one can choose the number of periods properly. Theoretically, the model allows for any number of T periods. However, limitations arise from a computational point of view, since every period adds additional complexity to the MIP or for the metaheuristic. To compare the different solution methods, we came up with 1, 2, and 3 periods to consider in order to explore computational performance and solution quality.

6.2.2. Computational results

The results are given in Table 8, Table 9, and Table 10, respectively. The name of an instance is given in the first column in the form X_Y_Z, where X, Y, and Z denote the number of customers, the number of potential locker box stations, and the instance number. Then, the total cost of the solution found by the MIP within the given time limit of 10 hours is reported. If the time limit is not breached, this is considered the optimal solution; otherwise, it is referred to as the best known solution (BKS). The computational time in seconds is the third column in the table. A breach of the predefined time limit is indicated by “time limit”. For the metaheuristic method, 5 runs were executed for each instance. The columns f^{avg}, f^{best}, and f^{worst} show the average, best, and worst solution over the performed runs, respectively. The information given by gap^{avg}, gap^{best}, and gap^{worst} refers to the percentage gap from the corresponding f to the BKS found by the MIP. t^{avg} reports the average computational time in seconds. The last row in each table gives the average values over all instances for the information belonging to the metaheuristic.

The results for the 1-period problem in Table 8 demonstrate that for 32 out of 35 instances, the MIP yielded the optimal solution within the given time limit of 10 hours. For almost all of these instances, the metaheuristic also found the optimal solution in the best case. On average, we have in the best case a gap of −0.08%, and in the worst case a gap of 0.10%. For the average solution cost, we have on average a gap of −0.04%. Therefore, the metaheuristic performs really well with respect to solution quality for the 1-period problem. With respect to computational time, we can see that for the smaller instances – and in some cases for larger ones – the MIP is even slightly faster than the metaheuristic method. However, for one of the 20 customers’ instances, the MIP exceeded the time limit, while the metaheuristic needed only about 2 minutes to yield an even better solution. Hence, there is no doubt that the metaheuristic clearly outperforms the MIP in terms of runtime for increasing problem size.

This also becomes obvious in Table 9, which shows the results for the 2-period problem. There, none of the 25 customers’ instances could be solved to proven optimality by using the MIP and predefined time limit. For the next smaller size category of 22 customers, only 1 out of 5 instances could be solved to optimality. Conversely, for the metaheuristic method, we still have an average runtime of not even 3 minutes over all results. Concerning solution quality, we can see that the best metaheuristic solution is worse than the BKS from the MIP in only two cases. On average, the best metaheuristic solutions have a gap of −0.07% from the BKS, the worst metaheuristic solutions have a gap of 0.14%, and the average values a gap of 0.01%. This shows that the metaheuristic gives very good results even for a 2-period problem.

The results for the 3-period problem are reported in Table 10. With an increasing number of planning periods, the complexity of the model increases, as reflected in the computational effort of the MIP. For the 3-period problem, one of the 12 customers’ instances breached the time limit. For the 20 customers’ instances, the MIP did not provide a proven optimal solution. For the computational time of the metaheuristic, no significant change could be observed in case of 3 periods. The average value is lower than that of the 2-period problem, but this can be traced to the fact that the 22 and 25 customers’ instances are not in the list any more, and thus their runtime value is not considered in the average value. The good performance of the metaheuristic with respect to solution quality is also confirmed for the 3-period problem. In the best case, we have, on average, a deviation of only 0.03% to the BKS, in the worst case a deviation of 0.35%, and in the average case 0.16%.

6.3. Algorithm design

Our metaheuristic designed for the location routing problem in this work comprises several components as presented in Section 4. Since we face a complex

instance	MIP BKS	t^{MIP} (s)	f^{avg}	f^{best}	f^{worst}	gap^{avg}	gap^{best}	gap^{worst}	t^{avg} (s)
10_5_1	108.48	3.16	108.48	108.48	108.48	0.00%	0.00%	0.00%	14.88
10_5_2	156.7	19.28	156.70	156.70	156.70	0.00%	0.00%	0.00%	32.57
10_5_3	149.38	10.81	149.38	149.38	149.38	0.00%	0.00%	0.00%	19.04
10_5_4	163.38	4.71	163.41	163.38	163.54	0.02%	0.00%	0.10%	20.48
10_5_5	117.66	5.76	117.66	117.66	117.66	0.00%	0.00%	0.00%	15.95
12_5_1	131.11	5.72	131.11	131.11	131.11	0.00%	0.00%	0.00%	24.12
12_5_2	157.53	5.73	158.07	157.53	160.02	0.34%	0.00%	1.58%	35.95
12_5_3	171.26	79.48	171.26	171.26	171.26	0.00%	0.00%	0.00%	16.68
12_5_4	159.62	173.35	159.62	159.62	159.62	0.00%	0.00%	0.00%	26.32
12_5_5	144.29	38.51	144.29	144.29	144.29	0.00%	0.00%	0.00%	23.46
15_5_1	157.25	30.82	157.25	157.25	157.25	0.00%	0.00%	0.00%	57.56
15_5_2	202.04	396.18	202.04	202.04	202.04	0.00%	0.00%	0.00%	40.06
15_5_3	147.84	62.19	147.84	147.84	147.84	0.00%	0.00%	0.00%	51.39
15_5_4	153.58	3079.10	153.58	153.58	153.58	0.00%	0.00%	0.00%	65.68
15_5_5	186.39	473.37	186.39	186.39	186.39	0.00%	0.00%	0.00%	54.14
17_5_1	173.99	30.38	174.07	174.07	174.07	0.05%	0.05%	0.05%	55.06
17_5_2	178.19	53.79	178.19	178.19	178.19	0.00%	0.00%	0.00%	60.66
17_5_3	196.17	152.48	196.17	196.17	196.17	0.00%	0.00%	0.00%	180.93
17_5_4	189.43	204.96	190.07	189.43	192.61	0.34%	0.00%	1.68%	162.08
17_5_5	189.37	489.29	189.41	189.37	189.58	0.02%	0.00%	0.11%	54.16
20_5_1	184.69	2664.26	184.69	184.69	184.69	0.00%	0.00%	0.00%	32.27
20_5_2	193.34	101.38	193.62	193.34	194.74	0.14%	0.00%	0.72%	197.21
20_5_3	188.39	114.99	188.83	188.39	189.48	0.23%	0.00%	0.58%	171.63
20_5_4	202.26	*time limit*	195.55	195.55	195.55	-3.32%	-3.32%	-3.32%	120.10
20_5_5	219.51	5287.48	219.61	219.51	219.63	0.04%	0.00%	0.05%	182.46
22_5_1	200.94	2594.96	200.94	200.94	200.94	0.00%	0.00%	0.00%	169.64
22_5_2	245.17	2130.97	245.17	245.17	245.17	0.00%	0.00%	0.00%	162.84
22_5_3	218.89	2686.64	218.89	218.89	218.89	0.00%	0.00%	0.00%	181.28
22_5_4	201.96	255.91	201.96	201.96	201.96	0.00%	0.00%	0.00%	343.39
22_5_5	212.54	2067.72	212.54	212.54	212.54	0.00%	0.00%	0.00%	199.72
25_5_1	219.44	1361.43	219.55	219.44	220.00	0.05%	0.00%	0.26%	504.05
25_5_2	227.93	*time limit*	228.49	227.93	229.58	0.24%	0.00%	0.72%	207.02
25_5_3	243.58	2937.65	243.58	243.58	243.58	0.00%	0.00%	0.00%	211.92
25_5_4	240.69	*time limit*	241.94	241.77	242.60	0.52%	0.45%	0.79%	436.77
25_5_5	228.19	20876.70	228.19	228.19	228.19	0.00%	0.00%	0.00%	445.72
			184.53	184.45	184.78	**-0.04%**	**-0.08%**	**0.10%**	130.78

Table 8: Comparison of MIP and metaheuristic (1-period problem)

problem and the computational times are high, it is important to check whether all components are indeed useful and necessary. First, we investigate how the improvement phase in the location algorithm contributes to the solution quality. Then, we evaluate different settings for the parameters used in the location heuristics. Finally, we consider a possible method for reducing the computational time in case of large instances.

6.3.1. Contribution of improvement phase

One component of the metaheuristic is the improvement phase, where a possibly better re-selection of locker box locations should be found. The scope of the improvement phase comprises here also the modified ADD (described in Section 4.3), since this is also a kind of improvement step, together with the other three improvement operators: add, drop, and swap (described in Sections 4.4.1 to 4.4.3). Below, we analyze how the solution quality and runtime change when we turn off the improvement phase. The calculations are done based on the experiments from Section 6.2.

In Section 6.2, we tested instances with 1, 2, and 3 periods. Here, we consolidate the results with respect to the number of periods. For example, the first line gives the average cost and runtime of all 1-period instances. Concerning runtime, we see that the runtime reduces when we do not run the improvement phase.

instance	MIP BKS	t^{MIP} (s)	f^{avg}	f^{best}	f^{worst}	gap^{avg}	gap^{best}	gap^{worst}	t^{avg} (s)
10_5_1	228.59	38.45	228.59	228.59	228.59	0.00%	0.00%	0.00%	13.99
10_5_2	230.33	119.32	230.92	230.33	232.19	0.26%	0.00%	0.81%	24.86
10_5_3	172.84	96.68	173.82	172.84	174.47	0.57%	0.00%	0.94%	21.92
10_5_4	188.67	3.46	188.67	188.67	188.67	0.00%	0.00%	0.00%	22.17
10_5_5	161.78	14.21	161.78	161.78	161.78	0.00%	0.00%	0.00%	13.91
12_5_1	187.70	87.71	187.70	187.70	187.70	0.00%	0.00%	0.00%	26.51
12_5_2	229.84	75.72	229.84	229.84	229.84	0.00%	0.00%	0.00%	21.30
12_5_3	239.54	63.43	239.54	239.54	239.54	0.00%	0.00%	0.00%	35.29
12_5_4	261.53	77.54	262.27	261.53	263.38	0.28%	0.00%	0.71%	43.09
12_5_5	210.72	161.41	210.72	210.72	210.72	0.00%	0.00%	0.00%	33.62
15_5_1	259.60	431.90	259.60	259.60	259.60	0.00%	0.00%	0.00%	44.19
15_5_2	280.16	*time limit*	281.54	281.50	281.56	0.49%	0.48%	0.50%	81.13
15_5_3	246.94	325.02	246.97	246.94	247.11	0.01%	0.00%	0.07%	38.28
15_5_4	264.51	1629.95	264.51	264.51	264.51	0.00%	0.00%	0.00%	85.36
15_5_5	287.01	2435.53	287.01	287.01	287.01	0.00%	0.00%	0.00%	73.62
17_5_1	312.61	28159.30	312.95	312.61	313.18	0.11%	0.00%	0.18%	118.08
17_5_2	254.07	2327.78	254.07	254.07	254.07	0.00%	0.00%	0.00%	73.52
17_5_3	305.85	4330.99	305.85	305.85	305.85	0.00%	0.00%	0.00%	146.16
17_5_4	284.69	1349.41	284.69	284.69	284.69	0.00%	0.00%	0.00%	72.60
17_5_5	281.92	3004.08	281.92	281.92	281.92	0.00%	0.00%	0.00%	55.18
20_5_1	324.44	949.84	324.90	324.44	326.26	0.14%	0.00%	0.56%	361.10
20_5_2	329.40	5780.62	329.40	329.40	329.40	0.00%	0.00%	0.00%	229.73
20_5_3	301.43	23321.10	301.43	301.43	301.43	0.00%	0.00%	0.00%	158.18
20_5_4	300.89	12874.10	300.89	300.89	300.89	0.00%	0.00%	0.00%	315.35
20_5_5	259.02	1580.35	259.61	259.02	261.96	0.23%	0.00%	1.14%	170.73
22_5_1	344.94	*time limit*	347.37	344.94	347.98	0.71%	0.00%	0.88%	195.18
22_5_2	330.33	3567.46	330.38	330.33	330.56	0.01%	0.00%	0.07%	214.75
22_5_3	319.33	*time limit*	319.33	319.33	319.33	0.00%	0.00%	0.00%	350.55
22_5_4	273.44	*time limit*	273.82	273.82	273.82	0.14%	0.14%	0.14%	300.35
22_5_5	359.19	*time limit*	359.19	359.19	359.19	0.00%	0.00%	0.00%	547.41
25_5_1	306.96	*time limit*	306.75	306.44	306.96	-0.07%	-0.17%	0.00%	551.87
25_5_2	334.05	*time limit*	334.05	334.05	334.05	0.00%	0.00%	0.00%	460.14
25_5_3	346.55	*time limit*	346.49	345.72	349.58	-0.02%	-0.24%	0.87%	428.69
25_5_4	340.65	*time limit*	340.65	340.65	340.65	0.00%	0.00%	0.00%	340.87
25_5_5	329.23	*time limit*	321.01	320.27	323.08	-2.50%	-2.72%	-1.87%	334.56
			276.81	276.58	277.19	**0.01%**	**-0.07%**	**0.14%**	171.55

Table 9: Comparison of MIP and metaheuristic (2-period problem)

Concerning solution quality, we observe a deterioration. For the 1-period instances it is, on average, 19% worse, while for the 3-period instances it is already worse by 39%. When we look at the average number of open stations across the respective instance set, we see that with an increasing number of periods, more stations have to be opened (which is logical, since with more periods you face more different scenarios in which to serve customers efficiently). With more open stations, there are more possibilities for choosing correctly or incorrectly. Consequently, the more open stations there are, the more important the improvement phase becomes.

For the purpose of benchmarking the heuristic solutions to the optimal or best known MIP solutions, more importance is given to the solution quality than to runtime. Thus, the improvement phase is important in this case.

6.3.2. Parameter analysis

The metaheuristic solution method presented in Section 4 contains various parameters. Below, we do an analysis on two of them, where the experiments from Section 6.2 are used as a base case (see Tables 8 – 10).

In the modified ADD algorithm (described in Section 4.3), we have to decide about the number of promising stations that we open in the first iteration, what can be

instance	MIP BKS	t^{MIP} (s)	f^{avg}	f^{best}	f^{worst}	gap^{avg}	gap^{best}	gap^{worst}	t^{avg} (s)
10_5_1	340.87	1820.12	340.87	340.87	340.87	0.00%	0.00%	0.00%	27.87
10_5_2	274.60	379.73	276.57	274.60	277.55	0.72%	0.00%	1.07%	23.66
10_5_3	302.73	13784.30	302.73	302.73	302.73	0.00%	0.00%	0.00%	25.24
10_5_4	276.96	6045.97	277.53	276.96	277.95	0.20%	0.00%	0.36%	32.53
10_5_5	344.88	1351.24	344.88	344.88	344.88	0.00%	0.00%	0.00%	33.78
12_5_1	368.43	3045.49	368.80	368.43	370.26	0.10%	0.00%	0.50%	42.92
12_5_2	362.69	3049.87	363.99	363.99	363.99	0.36%	0.36%	0.36%	46.25
12_5_3	356.13	833.23	358.87	356.13	359.56	0.77%	0.00%	0.96%	42.11
12_5_4	268.64	420.63	268.76	268.64	269.23	0.04%	0.00%	0.22%	34.16
12_5_5	340.67	*time limit*	341.26	340.67	341.66	0.17%	0.00%	0.29%	46.59
15_5_1	304.20	1762.58	304.20	304.20	304.20	0.00%	0.00%	0.00%	61.79
15_5_2	372.20	20707.50	375.35	375.35	375.35	0.85%	0.85%	0.85%	271.95
15_5_3	392.81	9285.06	392.81	392.81	392.81	0.00%	0.00%	0.00%	188.60
15_5_4	335.72	*time limit*	332.66	332.66	332.66	-0.91%	-0.91%	-0.91%	99.36
15_5_5	378.47	*time limit*	384.32	383.01	387.16	1.55%	1.20%	2.30%	210.26
17_5_1	398.78	4288.88	399.34	398.78	400.26	0.14%	0.00%	0.37%	223.13
17_5_2	345.07	*time limit*	339.76	339.73	339.86	-1.54%	-1.55%	-1.51%	340.98
17_5_3	409.72	14254.80	414.56	414.53	414.60	1.18%	1.17%	1.19%	151.80
17_5_4	360.47	25023.20	360.47	360.47	360.47	0.00%	0.00%	0.00%	162.15
17_5_5	424.53	*time limit*	425.04	424.53	427.07	0.12%	0.00%	0.60%	104.90
20_5_1	430.05	*time limit*	427.26	427.26	427.26	-0.65%	-0.65%	-0.65%	247.59
20_5_2	372.33	*time limit*	372.07	371.25	375.23	-0.07%	-0.29%	0.78%	105.78
20_5_3	428.53	*time limit*	429.49	428.53	433.35	0.22%	0.00%	1.12%	319.66
20_5_4	395.52	*time limit*	395.52	395.52	395.52	0.00%	0.00%	0.00%	325.22
20_5_5	467.89	*time limit*	471.46	470.52	471.96	0.76%	0.56%	0.87%	260.32
			362.74	362.28	363.46	**0.16%**	**0.03%**	**0.35%**	137.14

Table 10: Comparison of MIP and metaheuristic (3-period problem)

	improve on (base case)			improve off		gap	
instances	f^{avg}	t^{avg} (s)	# open stations	f^{avg}	t^{avg} (s)	f	t
1 period	184.53	130.78	1.49	184.82	32.50	0.19%	-71.08%
2 periods	276.81	171.55	1.97	277.45	46.39	0.26%	-72.96%
3 periods	362.74	137.14	1.88	364.15	29.43	0.39%	-77.34%
avg.	274.69	146.49	1.78	275.47	36.11	0.28%	-73.79%

Table 11: Contribution of improvement phase

interpreted as the list length L of the restricted candidate list.

The length of the restricted candidate list L is obtained by multiplying the length of the sorted station list (sorted by increasing objective value) by a random factor from a certain interval. In the base case of Section 6.2, this interval was chosen to be [0.4, 0.6]. Furthermore, when we evaluate which stations should be added to the already open stations, we consider only a subset of promising stations, giving another restricted candidate list. The length of the list is obtained as before. In the base case the interval for the random factor is [0.7, 1.0]. For the experiments in this part, the idea is to shorten the restricted candidate lists by using lower values for the intervals. This restricts the modified ADD in the sense that fewer possibilities for opening and combining stations are available.

To analyze the impact of these parameter changes, we run the metaheuristic method once by using [0.0, 0.2] and [0.2, 0.5] as range intervals, and once with [0.2, 0.4] and [0.4, 0.7].

In Table 12, we can see that the solutions are up to 0.18% worse compared to the base case. One could assume that the computational time should decrease, when the solution space for the modified ADD is reduced and less effort is made in this step. However, we observe an increase in total average runtime. A possible reason for this could be that more effort has

instances	base case		[0.0,0.2] and [0.2,0.5]				[0.2,0.4] and [0.4,0.7]			
	f^{avg}	t^{avg} (s)	f^{avg}	t^{avg} (s)	f gap	t gap	f^{avg}	t^{avg} (s)	f gap	t gap
1 period	184.53	130.78	184.86	171.55	0.18%	31.17%	184.79	182.86	0.14%	39.83%
2 periods	276.81	171.55	277.30	231.37	0.18%	34.87%	277.17	212.20	0.13%	23.70%
3 periods	362.74	137.14	362.95	184.74	0.06%	34.70%	362.98	160.47	0.07%	17.01%
avg.	274.69	146.49	275.04	195.88	0.14%	33.58%	274.98	185.18	0.11%	26.84%

Table 12: Parameter analysis for the restricted candidate list in the modified ADD

instances	base case		7 iterations				5 iterations				3 iterations			
	f^{avg}	t^{avg} (s)	f^{avg}	t^{avg} (s)	f gap	t gap	f^{avg}	t^{avg} (s)	f gap	t gap	f^{avg}	t^{avg} (s)	f gap	t gap
1 period	184.53	130.78	184.69	127.78	0.09%	-2.29%	184.81	110.98	0.15%	-15.14%	184.90	86.64	0.20%	-33.75%
2 periods	276.81	171.55	276.82	166.84	0.00%	-2.75%	276.86	145.78	0.02%	-15.02%	276.88	107.57	0.03%	-37.29%
3 periods	362.74	137.14	362.98	130.92	0.07%	-4.54%	363.16	99.38	0.12%	-27.53%	363.20	82.68	0.13%	-39.71%
avg.	274.69	146.49	274.83	141.85	0.05%	-3.19%	274.95	118.71	0.10%	-19.23%	274.99	92.30	0.12%	-36.92%

Table 13: Parameter analysis for the iterator in the improvement phase

instances	base case f^{avg}	t^{avg} (h)	estimator for routing f^{avg}	t^{avg} (h)	gap f	t
C1	796.99	5.60	811.13	2.03	1.77%	-63.70%
C2	568.46	5.40	593.34	0.79	4.38%	-85.29%
R1	939.20	7.20	971.62	2.19	3.45%	-69.63%
R2	780.06	6.40	825.37	1.77	5.81%	-72.31%
RC1	1135.11	7.20	1160.45	2.15	2.23%	-70.19%
RC2	926.79	6.20	967.55	1.65	4.40%	-73.37%
avg.	857.77	6.33	888.24	1.76	3.67%	-72.42%

Table 14: Use an estimator as speed-up for routing part in location heuristics

to be spent on the improvement phase (where the add, drop and swap operators are executed), when less effort is spent on the modified ADD.

Another parameter, that has to be determined in the metaheuristic method for the MPLRPLB, is the iterator in the improvement phase, that is used as a termination criterion by counting the number of unsuccessful iterations. We did experiments with varying values for this iterator.

Based on Table 13, we can conclude that the solution quality decreases with a lower value for the iterator. For a value of 7 the solutions across all instances are, on average, 0.05% worse compared to the base case (iterator value 10). For 5 iterations the average deterioration is 0.10% and for 3 iterations it is 0.12%. Moreover, we can observe that the runtime decreases, since the improvement phase terminates earlier when the iterator is set to a lower value. For the lowest value, which is 3 iterations, the runtime is on average decreased by 36.92%.

As a result of this parameter analysis, we can conclude that for practical purposes, computational time could be saved by choosing a small iterator at rather low cost. However, when evaluating different location strategies, a more precise estimation of the costs is important so that these results were based on 10 iterations.

6.3.3 Runtime analysis

In Section 6.1, we see that for the 100 customer Solomon instances, the metaheuristic method takes more than 5 hours on average. To keep the computational time within boundaries, we levied some runtime restrictions: the algorithm for the 1-period problem was performed with a runtime time limit of 10 minutes and the improvement phase of the location selection was performed with a runtime limit of 1 hour.

To further reduce the computational time of the instances, the idea now is to run the routing part in the location heuristics faster. ALNS is used to find a good routing solution. Depending on how many destroy-repair iterations there are, there is a trade-off between solution quality and runtime. In the location heuristics of our overall method, it is maybe sufficient to have only an estimation of a routing solution when evaluating whether a locker location is beneficial. Hence, we reduce the number of iterations in the ALNS of the location heuristics to reduce total runtime. When we find a new best selection of locker locations, the routing is optimized with the usual ALNS iteration settings.

We then analyze how the solution quality changes in relation to the results we obtained in Section 6.1 (see Tables 1 – 6). Note: We use those instances for comparison, where k-means strategy was used for distributing locker station, since this strategy has proven to perform overall very well.

The results in Table 14 show that the runtime can be reduced substantially when a faster procedure for the routing part is used in the location heuristics. On average, the runtime is now 1.76 hours compared to 6.33 from the base case of Section 6.1. However, this comes at the cost of solution quality, where an average deterioration of 3.76% can be observed.

With the experiments in Section 6.1 we compared the four different data generation concepts based on the 100 customers Solomon instances (extended for our model). For this, we put more emphasis on solution quality, to have a rather meaningful comparison, while an average runtime of 6.33 hours was considered acceptable. However, if you want to solve bigger instances or apply the method in a real-world setting, a lower runtime might be of more importance.

7. CONCLUSION

We presented the MPLRPLB as a relevant problem for research as well as for practice. A description of the MPLRPLB was given, followed by a complete MIP formulation. In the MPLRPLB, a set of potential locations for locker box stations is given as input. Within the model, the locations to be realized must be determined. Realizing a station comes at a cost that is added to the total cost of routing and compensation. Routing and delivery decisions have to be made in every period, while the decision of which stations to open is made only once within the multiple periods.

To solve the problem, we developed a metaheuristic solution method that basically evaluates different configurations of open and closed stations. For a certain setting of open/closed stations, the solution of the resulting locker box delivery problem was computed by solving the single-period problems for each period. Total costs were computed and compared to other configurations of open/closed stations. Stations were opened and closed by specific heuristics and operators. In the experimental study, we solved small self-generated instances exactly once by using the proposed MIP and once by using the proposed metaheuristic. Problems with 1, 2, and 3 periods were considered. The results obtained with the two methods were then compared with respect to solution quality (i.e., objective function value) and computational time. For the 1-period problem, the biggest instances had 25 customers and 5 potential stations. We could see that for some of the instances of this complexity, the MIP did hit the given time limit of 10 hours. For the 2 and 3-period problems, the MIP found its limits at instances with a smaller number of customers. The computational effort of the metaheuristic did not show over-proportional increases with increasing problem size. With respect to solution quality, the metaheuristic performed very well for the 1-period problems, with an average gap of −0.04% over all instances and over 5 runs for each instance compared to the BKS of the MIP. For the 2-period problems, the average gap was 0.01% and for the 3-period problems it was 0.16%. For the 3-period problem, the gap of the worst metaheuristic solution was 0.35% (over all instances of the 3-period problem). Thus, we could see that the metaheuristic gave very good results with respect to solution quality.

Furthermore, we designed different ways to generate the input data for the set of locker box locations. That is, we had an instance with customer information, in particular about the customers' home location and evaluated different input scenarios for the potential locations of the locker box stations. The strategies to distribute the locations are random distribution, distribution based on a grid structure, and based on clustering the customer locations. For the numerical experiments, we used the Solomon instances since they are classified according to specific characteristics of the distribution of customer nodes and the length of the customers' time windows. The results showed that for the clustered instances C1 and C2, the clustering strategy worked best on average. For the random instances R1, the random grid strategy worked best on average, but the clustering strategy only maintained a small gap at second place. For the random instances R2, the clustering strategy worked best. For the random-clustered instances RC1, the fixed grid strategy was on average the best strategy; however, the clustering strategy was again close to it. For the random-clustered instances RC2, the clustering strategy again performed best on average. Thus, we conclude that the clustering strategy based on the k-means algorithm fits very well to several geographic customer settings. We used the experimental results from this study to derive some conclusions about the relation between the length of the customers' time windows and the utilization of locker box delivery. For this, we used the number of locker box customers and of used stations to assess locker box utilization. The outcome was that locker box delivery is used more when time windows are tighter. The reason for that may lie in the fact that efficient routing of customers is difficult when they should be served at home during the time window and the time windows are short. Hence, in this case, the possibility of locker box delivery provides more potential to increase efficiency.

ACKNOWLEDGEMENTS

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged.

REFERENCES

David Arthur and Sergei Vassilvitskii. K-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

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.

Clutch Logistic Survey. Package theft statistics: How to prevent lost or stolen packages, 2019. Available online at *https://clutch.co/logistics/resources/package-theft-statistics-prevent-lost-stolen-packages*, accessed April 19, 2022.

Yael Deutsch and Boaz Golany. A parcel locker network as a solution to the logistics last mile problem. *International Journal of Production Research*, 56(1-2):251–261, 2018.

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.

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, 2021.

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.

Jasmin Grabenschweiger, Fabien Tricoire, and Karl F Doerner. Finding the trade-off between emissions and disturbance in an urban context. *Flexible Services and Manufacturing Journal*, 30(3):554–591, 2018.

Jasmin Grabenschweiger, Karl F Doerner, Richard F Hartl, and Martin WP Savelsbergh. The vehicle routing problem with heterogeneous locker boxes. *Central European Journal of Operations Research*, 29(1):113–142, 2021.

Michael Kahr. Determining locations and layouts for parcel lockers to support supply chain viability at the last mile. 2022.

Alfred A Kuehn and Michael J Hamburger. A heuristic program for locating warehouses. *Management Science*, 9 (4):643–666, 1963.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Bruce T Lowerre. *The harpy speech recognition system*. Carnegie Mellon University, 1976.

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

Gábor Nagy and Saïd Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

Ido Orenstein, Tal Raviv, and Elad Sadan. Flexible parcel delivery to automated parcel lockers: models, solution methods and analysis. *EURO Journal on Transportation and Logistics*, 8(5):683–711, 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.

Caroline Prodhon and Christian Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

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

Stefan Schwerdfeger and Nils Boysen. Optimizing the changing locations of mobile parcel lockers in last-mile distribution. *European Journal of Operational Research*, 285(3):1077 – 1094, 2020.

Pawel Sitek and Jarosław Wikarek. Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach. *Annals of Operations Research*, 273(1-2): 257–277, 2019.

Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

Marjolein Veenstra, Kees Jan Roodbergen, Leandro C Coelho, and Stuart X Zhu. A simultaneous facility location and vehicle routing problem arising in health care logistics in the netherlands. *European Journal of Operational Research*, 268(2):703–715, 2018.

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.