INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 30 (2023) 915–945
DOI: 10.1111/itor.13125

The location routing problem with facility sizing decisions

Rafael D. Tordecilla[a,b,*], Jairo R. Montoya-Torres[b], Carlos L. Quintero-Araujo[c], Javier Panadero[a,d] and Angel A. Juan[e]

[a] *IN3 – Computer Science Department, Universitat Oberta de Catalunya, Barcelona, Spain*
[b] *School of Engineering, Universidad de La Sabana, Chia, Colombia*
[c] *International School of Economics and Administrative Sciences, Universidad de La Sabana, Chia, Colombia*
[d] *Euncet Business School, Terrassa, Spain*
[e] *Department of Applied Statistics and Operations Research, Universitat Politècnica de València, Alcoy, Spain*
E-mail: rtordecilla@uoc.edu [Tordecilla]; jairo.montoya@unisabana.edu.co [Montoya-Torres]; carlosqa@unisabana.edu.co [Quintero-Araujo]; jpanaderom@uoc.edu [Panadero]; ajuanp@eio.upv.es [Juan]

Received 6 August 2020; received in revised form 21 September 2021; accepted 25 January 2022

Abstract

The location routing problem (LRP) integrates operational decisions on vehicle routing operations with strategic decisions on the location of the facilities or depots from which the distribution will take place. In other words, it combines the well-known vehicle routing problem (VRP) with the facility location problem (FLP). Hence, the LRP is an *NP-hard* combinatorial optimization problem, which justifies the use of metaheuristic approaches whenever large-scale instances need to be solved. In this paper, we explore a realistic version of the LRP in which facilities of different capacities are considered, i.e., the manager has to consider not only the location but also the size of the facilities to open. In order to tackle this optimization problem, three mixed-integer linear formulations are proposed and compared. As expected, they have been proved to be cost- and time- inefficient. Hence, a biased-randomized iterated local search algorithm is proposed. Classical instances for the LRP with homogeneous facilities are naturally extended to test the performance of our approach.

Keywords: location routing problem; heterogeneous facilities; biased randomization; metaheuristics

1. Introduction

The location routing problem (LRP) is a traditional strategic-tactical-operational problem that considers a set of potential facilities and a set of customers with a known demand, whose main decisions are: (i) the number and location of facilities to open, (ii) the allocation of customers to open facilities, and (iii) the design of routes to serve customers from each facility using a fleet of vehicles.

[*] Corresponding author.

This means that the LRP considers jointly the facility location problem (FLP) and the vehicle routing problem (VRP). As both problems are *NP-hard* in nature, the LRP maintains this characteristic (Nagy and Salhi, 2007). Hence, its inherent complexity makes necessary the use of approximate solution approaches, such as heuristic or metaheuristic algorithms, to solve it efficiently, especially when dealing with large-sized instances. Therefore, the research about this problem has increased mainly during the last decade, given the recent advances in computing power.

Different versions of the LRP have been considered in the scientific literature depending on the analyzed constraints. Among them, we can find: (i) the capacitated version in which only the vehicle capacity is limited; and (ii) the capacitated version that establish capacity constraints for both depots and vehicles. The latter variant assumes that the following parameters are known in advance: the facility opening cost, the traveling cost between two nodes, the demand of each customer, the capacity of each vehicle, and the capacity (size) of each open facility. Nevertheless, this latter is traditionally assumed as a fixed parameter, i.e., once a facility is open, a rigid known size is assigned. However, some real-world problems show the relevance of considering a set of available sizes to select those that fit better. Cases from different industries that employ either LRP or non-LRP approaches have considered this set. An example of the latter is shown by Tordecilla-Madera et al. (2017), who address the problem of locating a set of milk refrigeration tanks for a dairy cooperative in Colombia. Several tank sizes are found in the market, i.e., the considered problem must determine both the number and size of tanks that should be bought and their location, among other decisions. Correia and Melo (2016) state that, in applied problems, the capacity is often acquired in the market from a set of discrete sizes. Furthermore, economies of scale can be incurred when the facility size is an additional variable to model. The different available sizes are usually associated with investment activities, such as building facilities (Zhou et al., 2019), qualifying workforce (Correia and Melo, 2016), or purchasing equipment (Tordecilla-Madera et al., 2017). This means that considering facility sizing decisions is a strategy for decreasing the invested capital, if necessary, or even for reducing the operational costs by increasing the investment level, as we demonstrate in this work.

Allowing facility sizing decisions is a form of soft constraint (Juan et al., 2020). The traditional LRP considers a rigid value for the maximum capacity of a facility, however, this constraint can be “violated” by providing multiple size alternatives and incurring an additional opening cost for a bigger size. This approach is quite common in real-life cases. Nevertheless, our approach not only allows bigger sizes but also smaller ones in order to diminish costs. Besides, considering sizing decisions increases the hardness of the problem. Therefore, we propose an approach formed by a biased-randomized version of a savings-based constructive heuristic (Grasas et al., 2017) and the iterated local search (ILS) metaheuristic (Lourenço et al., 2019) to solve a deterministic version of the LRP with facility sizing decisions. Hence, the contributions of this work are fourfold: (i) to analyze a more realistic version of the capacitated LRP in which different sizes for each depot location are considered, (ii) to extend classical medium- and large-sized benchmark instances of the LRP in order to adapt them to the variant under study, (iii) to propose a competitive metaheuristic algorithm based on biased randomization techniques to deal with the LRP with facility sizing decisions, and (iv) to provide a numerical analysis of the results obtained by employing alternative mixed-integer linear programs (MIP), in terms of costs and computing times. The remainder of this paper is organized as follows: Section 2 presents a brief literature review on facility sizing decisions in both the LRP and other logistic problems. Section 3 presents a mixed-integer programming model of the

LRP with facility sizing decisions. Section 4 provides details on the solving approach used to deal with the problem. Section 5 describes the procedure used to extend classical instances and presents the results obtained when using both an exact approach and our metaheuristic approach; in addition, it includes a sensitivity analysis on the effects generated by different sizes. Section 6 draws some conclusions and future research perspectives. Finally, the appendix analyzes alternative MIP models for the considered problem.

2. Literature review

In general, only a handful of papers addressing the LRP includes facility sizing decisions. These papers are summarized in Subsection 2.1, along with traditional and recent works about the LRP. Alternatively, facility sizing decisions are usually considered by papers addressing strategic-tactical problems in a supply chain, i.e., by papers that exclude routing. These works are outlined in Subsection 2.2.

2.1. The location routing problem

The general topic regarding the LRP has been broadly studied, especially in the last few decades. Maranzana (1964) is perhaps the first author who combines location decisions with transport costs. Multiple highly cited papers were published some years later. For instance, Jacobsen and Madsen (1980) and Madsen (1983) assess three heuristics to solve an LRP for distributing newspapers. Perl and Daskin (1985) present a mixed integer program to solve a warehouse LRP. The authors propose a heuristic that decomposes the problem into three interdependent subproblems. They consider that both depots and vehicles are capacitated. The model is applied to a real distribution system in an area including Missouri, Oklahoma and Western Kansas. Theoretical problems are also addressed in this period, as well as the use of exact algorithms to solve them. For instance, Laporte et al. (1986) propose an integer linear program to solve a capacitated LRP. The capacitated part of the problem refers only to the vehicle capacity, i.e., open depots are uncapacitated. An exact algorithm applied after a constraint relaxation method is employed to solve the problem optimally. Laporte et al. (1988) study a cost-constrained LRP, where the cost of each designed route cannot exceed a known limit. Capacity-constrained and cost-constrained multi-depot VRPs are also analyzed. Finally, Laporte et al. (1989) are perhaps the first authors addressing a stochastic LRP, in which customers' demands are random. A chance constraint model and a bounded penalty model are proposed and solved optimally.

Aykin (1995) addresses a hub location routing problem where hubs can interact each other. An integer program is formulated and an iterative heuristic is proposed to solve the problem. Tuzun and Burke (1999) also show a mixed integer program, based on the work by Perl and Daskin (1985). Unlike these authors, Tuzun and Burke (1999) do not consider depots capacity. Additionally, they propose a two-phase tabu search algorithm as solution approach. Wu et al. (2002) consider a multi-depot LRP where vehicles are heterogeneous and the fleet of each type of vehicle is limited. A heuristic decomposition method is proposed, where the problem is divided into two subproblems. Then, each subproblem is solved through an embedded simulated annealing algorithm. Prins et al.

(2006) hybridize GRASP with a learning process and a path relinking to solve a capacitated LRP. A randomized version of the Clarke and Wright savings heuristic is employed, as well as several local search procedures. Prins et al. (2007) propose a metaheuristic that decomposes the LRP into two phases: the first one solves the facility location problem through a Lagrangean relaxation, and the second phase employs a granular tabu search to solve the routing part.

LRP cases for hazardous waste management are shown by Alumur and Kara (2007) and Samanlioglu (2013). These authors address facility sizing decisions, although this concept is employed indirectly, i.e., they consider a set of waste treatment technologies, and each technology has a different available capacity to be installed. For instance, Alumur and Kara (2007) propose a multi-objective LRP. They formulate a mixed integer programming model to minimize both total costs and transportation risk. A real-world problem in the Central Anatolian region of Turkey is considered and solved using an exact algorithm. This problem has also been tackled by Samanlioglu (2013). In this paper, three objective functions are intended to be minimized: total costs, transportation risk, and treatment and disposal centers risk. A mixed integer programming model is formulated and solved employing an exact algorithm. A real-world problem in the Marmara region of Turkey is considered. Yu et al. (2010) employ a simulated annealing algorithm to solve a capacitated LRP. Different sets of benchmark instances are used to test the proposed heuristic. Sustainability and food perishability are addressed by Govindan et al. (2014) in a two-echelon LRP with time windows. The authors propose a metaheuristic that hybridizes a multi-objective particle swarm optimization with an adapted multi-objective variable neighborhood search.

Fields such as supply chain network design (Lashine et al., 2006), horizontal cooperation (Quintero-Araujo et al., 2019a), city logistics (Nataraj et al., 2019), and humanitarian logistics (Ukkusuri and Yushimito, 2008) also show the application of the LRP in a deterministic context. Alternatively, the LRP has also been studied considering stochastic parameters. For instance, Quintero-Araujo et al. (2019b) propose a simheuristic algorithm to deal with demand uncertainty for the LRP. Rabbani et al. (2019) propose also a simheuristic approach to solve an LRP in the context of the hazardous waste management industry. Both generated waste and number of people at risk are stochastic. Literature reviews by Nagy and Salhi (2007), and more recently by Prodhon and Prins (2014) show the rise of the LRP. Traditional taxonomies, such as capacitated or uncapacitated vehicles, capacitated or uncapacitated depots, single or multiple periods, among others are tackled by these authors. A broad taxonomy is also presented by Lopes et al. (2013). Schneider and Drexl (2017) provide a review focused on solving approaches for the standard LRP, such as exact methods, matheuristics or metaheuristics. Albareda-Sambola and Rodríguez-Pereira (2019) review different mathematical formulations for the LRP, as well as heuristic algorithms and location-arc routing problems. Non-standard LRP approaches are addressed by Drexl and Schneider (2015). Papers regarding the use of fuzzy data, continuous locations, split deliveries, among other variants are reviewed. Nevertheless, the explicit consideration of facility sizing decisions is not taken into account by these reviews. To the best of our knowledge, only five papers consider it in an LRP context: two works considering only deterministic parameters, and three works addressing uncertainty. These papers are referenced below.

Hemmelmayr et al. (2017) consider a deterministic periodic LRP for collaborative recycling in hunger relief agencies. Possible depot locations belong to the same set as the customers, i.e., some customers are chosen to locate the depots there. A mixed-integer programming model is proposed, which is solved through CPLEX for small instances. Then, an adaptive large neighbourhood search heuristic is proposed to solve small and large instances. High cost savings for the agencies are attained through this approach. A variant in the problem considers that customers and depots belong to different sets of nodes. For instance, Tunalıoğlu et al. (2016) consider a multiperiod LRP for collecting olive oil mill wastewater. A mixed-integer non-linear model is proposed. Then, the problem is solved through a metaheuristic named *multiperiodic-adaptive large neighbourhood search (MP-ALNS)*. A case study in Turkey is considered. A sensitivity analysis is carried out and numerical results are showed as well as some managerial insights. Zhou et al. (2019) also consider different sets for depots and customers. They propose a hybrid approach combining a genetic algorithm and a simulated annealing approach to solve a bilevel multisized terminal LRP with simultaneous home delivery and customer's pickup services. A real-world case in an e-commerce company in China is considered. Parcels' deliveries can be carried out between a distribution center (DC) and intermediate terminals, and between the same DC and the customers. The customers have the option of either to receive the deliveries at their homes or pick up the parcels in a terminal. Hence, each customer's demand is computed considering the probability of selecting each alternative. This probability depends on the distance of the customer to its closest terminal. Once the demand has been calculated, this parameter is considered as deterministic.

A simheuristic algorithm is proposed by Tordecilla et al. (2020) to solve an LRP with facility sizing decisions. They consider all customers' demands to be stochastic. Results are assessed in terms of both total costs and reliability. Alternatively, Tordecilla et al. (2021) propose a fuzzy simheuristic to solve the same problem, considering a more general case in which the demand of a subset of customers is stochastic, while the demand of the complementary subset is modeled in a fuzzy fashion. Particular cases in which all customers' demands are deterministic, stochastic or fuzzy are also analyzed. Both aforementioned papers employ medium-sized benchmark instances (up to 40 customers and 5 alternative depots) to test their approach. Both consider a function cost composed of deterministic and failure costs. In turn, the deterministic part comprises opening and routing costs. In contrast to these works, we address a deterministic LRP and test our approach employing a larger set of benchmark instances with up to 200 customers and 10 alternative depots. In addition, the previous works in the literature only consider one fixed variability range between available sizes. Instead, we perform a sensitivity analysis where this range is also modified. Hence, a deeper analysis regarding the assigned sizes is carried out. Finally, in addition to the heuristic solution approach, the problem is modeled using three different mathematical programming approaches. Their performance, in terms of computational time and solution quality, shows to be superior to the one associated with other solving methods.

2.2. Facility sizing decisions in strategic-tactical problems

Papers considering only strategic or strategic-tactical decisions are more likely to include facility sizing decisions. These papers exclude routing and address problems such as the FLP or the supply chain network design (SCND). For instance, an early work in the FLP field is authored by Shulman (1991). He presents a multi-period problem that consists in determining the number and size of facilities to place in each available location, i.e., several facilities can be placed in the same site. The author proposes a MILP formulation and a Lagrangian relaxation to solve it. More recently, Correia and Melo (2016) address a multi-period FLP where some customers allow delayed deliveries. The authors highlight that the inclusion of facility sizing decisions increases the search space and that this topic is not frequently addressed in the literature. Besides, closing facilities is allowed depending on demands and costs trends. Two MILP models are proposed and linear relaxations are provided to solve the models. A similar problem and solving method are addressed by Correia and Melo (2017). However, these authors tackle facility sizing decisions by considering modular capacities, i.e., single-capacity modules are available, and sizing decisions consist in selecting the optimal number of modules for each location site. Alternatively, Sauvey et al. (2020) propose several constructive and improvement heuristic algorithms to solve efficiently this problem. Finally, Wu et al. (2006) address a single-period FLP that considers a general setup cost, i.e., the total location cost is formed by both a fixed term depending on the site and a variable term depending on the facility size. Two MILP models are proposed and solved through a Lagrangian heuristic algorithm.

In the SCND field, Badri et al. (2013) propose a MILP model for designing a four-echelon supply chain network. Multiple periods and commodities are considered. Periods are classified into two sets for making strategic and tactical decisions, respectively. An initial capacity is available if a facility is decided to be open. Then, a set of options can be used to expand the initial capacity. Authors develop a heuristic based on a Lagrangian relaxation method to solve the model. Bashiri et al. (2012) address a similar problem, which is solved exactly through the CPLEX solver. Cortinhal et al. (2015) do not only address the design but also the re-design of a supply chain network, driven by changes in business and market conditions. Location decisions in the former case include the number, size and location of new facilities, whereas in the latter case closing existing facilities is also allowed. Multiple periods, products and transportation modes are considered. A MILP formulation is proposed to model this problem. Finally, Le et al. (2019) focus in the location of external intermediate warehouses in an SCND context. Multiple periods, products and transportation modes are also considered. Closing actions are not allowed, i.e., once a warehouse is chosen, it remains open until the end of the planning horizon, either with the same size or bigger.

3. Problem formulation

The capacitated location-routing problem (LRP) consists in opening one or more depots and designing for each open depot a number of routes whose total customer demand does not exceed the depot capacity. Each route must start and finish at the same depot. The total number of vehicles used (or routes performed) is a decision variable. The set of routes must serve all customers and minimize a total cost comprising the fixed and variable costs of open depots, the fixed costs of the used vehicles and the costs of the routes (transportation). Figure 1 depicts an example of a complete LRP solution where circles represent the customers, triangles symbolize the open depots, squares represent the non-open depots, and arrows symbolize the designed routes. For each open depot a set of routes starting and finishing at the corresponding depot location is designed to serve all customers demands.

Formally speaking, the LRP can be defined on a complete, weighted, and undirected graph $G(V, A, C)$, in which V is the set of nodes (comprising the subset J of potential depot locations and subset I of customers), A is the set of arcs, and C is the cost matrix of traversing each arc. A set of unlimited homogeneous vehicles with capacity constraints (K) is available to perform the routes. Moreover, it is assumed that all vehicles are shared by all depots (i.e., no depot has a specific

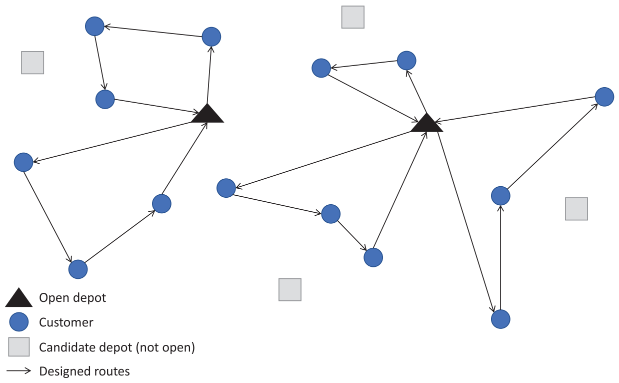


Fig. 1. Graphical representation of a CLRP solution.

fleet) and each arc $a \in A$ satisfies the triangle inequality. Customer demands are deterministic and known in advance. Each customer must be serviced from the depot to which it has been allocated by a single vehicle. The version studied in this paper considers that the capacity of each depot is not known in advance, instead it is a decision to be made. Hence, a discrete set L of available sizes is known, from which the best alternative for each depot is selected. The following constraints must be satisfied: (a) the total demand of customers assigned to one depot must not exceed its capacity, (b) each route begins and ends at the same depot, (c) each vehicle performs at most one trip, (d) each customer is served by one single vehicle (split deliveries are not allowed), and (e) the total demand of customers visited by one vehicle fits its capacity. The location routing problem with facility sizing decisions can be formulated as a mathematical programming model, whose sets, parameters, and variables are shown in Table 1.

$$Minimize \quad \sum_{j\in J}\sum_{l\in L}(f_j + o_{jl})y_{jl} + \sum_{a\in A}\sum_{k\in K} c_a w_{ak} + \sum_{a\in\delta^+(J)}\sum_{k\in K} v w_{ak} \tag{1}$$

s.t.

$$\sum_{k\in K}\sum_{a\in\delta^-(i)} w_{ak} = 1, \quad \forall i \in I \tag{2}$$

$$\sum_{i\in I}\sum_{a\in\delta^-(i)} d_i w_{ak} \leq q, \quad \forall k \in K \tag{3}$$

$$\sum_{a\in\delta^+(n)} w_{ak} = \sum_{a\in\delta^-(n)} w_{ak}, \quad \forall k \in K, \forall n \in V \tag{4}$$

$$\sum_{a\in\delta^+(J)} w_{ak} \leq 1, \quad \forall k \in K \tag{5}$$

$$u_{ik} + d_h \leq u_{hk} + M(1 - w_{ak}), \quad \forall a \in \delta^+(i \in I) \cap \delta^-(h \in I), \forall k \in K \tag{6}$$

Table 1
Sets, parameters, and variables of a 3-index model for the LRP with facility sizing decisions

Sets
$V =$ Set of nodes
$K =$ Set of vehicles
$L =$ Set of available sizes
$I =$ Set of customers, $I \subset V$
$J =$ Set of depots, $J \subset V$
$A =$ Set of arcs, $A = V \times V = \{(m, n) : m \in V, n \in V \wedge m \neq n\}$
$\delta^+(S) =$ Set of arcs leaving S, $S \subset V$, $\delta^+(S) \subset A$
$\delta^-(S) =$ Set of arcs entering S, $S \subset V$, $\delta^-(S) \subset A$

Parameters
$s_{jl} =$ Available size of type $l \in L$ for the depot $j \in J$
$d_i =$ Demand of customer $i \in I$
$f_j =$ Fixed opening cost of depot $j \in J$
$o_{jl} =$ Variable opening cost of depot $j \in J$ with size of type $l \in L$
$c_a =$ Cost of traversing arc $a \in A$
$v =$ Fixed cost for using a vehicle
$q =$ Capacity of each vehicle
$M =$ A very large number when compared to the magnitude of the rest of the parameters

Variables
$y_{jl} =$ Binary variable equal to 1 if depot $j \in J$ is open with size of type $l \in L$, 0 otherwise
$x_{ij} =$ Binary variable equal to 1 if customer $i \in I$ is assigned to depot $j \in J$, 0 otherwise
$w_{ak} =$ Binary variable equal to 1 if arc $a \in A$ is used in the route performed by vehicle $k \in K$, 0 otherwise
$u_{ik} =$ Accumulated deliveries by vehicle $k \in K$ until customer $i \in I$

$$\sum_{a \in \delta^+(j)} w_{ak} + \sum_{a \in \delta^-(i)} w_{ak} \leq 1 + x_{ij}, \quad \forall i \in I, \forall j \in J, \forall k \in K \tag{7}$$

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \tag{8}$$

$$\sum_{i \in I} d_i x_{ij} \leq \sum_{l \in L} s_{jl} y_{jl}, \quad \forall j \in J \tag{9}$$

$$\sum_{l \in L} y_{jl} \leq 1, \quad \forall j \in J \tag{10}$$

$$\forall\, y_{jl}, x_{ij}, w_{ak} \in \{0, 1\} \tag{11}$$

$$\forall\, u_{ik} \geq 0 \tag{12}$$

The objective function (1) minimizes the total costs. These are comprised by the opening cost—the distance-based cost and the cost of the usage of vehicles. The constraints of the model are explained next. Constraints (2) guarantee that each customer is served by a single route. Constraints (3) are associated to vehicle capacity. Constraints (4) and (5) guarantee the continuity of each route and the return of a route to the depot from which it has started. Constraints (6) are devoted to eliminate sub-tours. Constraints (7) guarantee that a customer is only assigned to a depot if there are routes serving that depot. Constraints (8) guarantee that a customer is assigned to only one depot. Constraints (9) ensure that the total demand of the customers allocated to a single depot does not exceed its assigned size. Constraints (10) guarantee that a single size is assigned to an open depot. Constraints (11) and (12) define the values of decision variables. This is the model employed to obtain our first set of results shown in Section 5.1. Nevertheless, different models can be formulated to represent our addressed problem. The appendix shows two additional models, which are compared with the aforementioned one.

4. Solving approach

The problem described in Section 3 is *NP-hard*, since it contains as special cases the capacitated vehicle routing problem or CVRP (single-depot case), the multi-depot VRP (case without location decisions), and the facility location problem, all of them known to be computationally hard. Hence, the LRP solution space is even much larger than the one of each individual problem, which makes prohibitive the use of exact methods to solve medium- and large-scale instances. Therefore, a metaheuristic approach is proposed. The implemented method is based on the work by Quintero-Araujo et al. (2017), who solve the LRP using a biased-randomized iterated local search (BR-ILS) metaheuristic. As discussed in Gruler et al. (2017) and Gonzalez-Martin et al. (2018), these frameworks are efficient, relatively easy-to-implement, do not contain a large number of parameters (therefore avoiding time-consuming setting processes), and offer an excellent trade-off between simplicity and performance. Thus, they have also been successfully employed in solving other combinatorial optimization problems (Ferrer et al., 2016; Guimarans et al., 2018; Muñoz-Villamizar et al., 2019; Londoño et al., 2020). The work by Quintero-Araujo et al. (2017) has a fixed input parameter for the depots size, which is the traditional approach for the LRP. Our approach extends this previous work considering that this parameter is not fixed, i.e., several known sizes are provided and our approach selects those that minimize the total routing and opening costs. Figure 2 depicts the flowchart of our approach, which is composed of two phases. The Phase 1 (blue) selects quickly some top complete solutions and the Phase 2 (pink) intensifies the search starting from these solutions as a base.

Firstly, the Phase 1 calculates a minimum (lb) and a maximum quantity (ub) of required depots, by dividing the total demand by the maximum and the minimum available size, respectively. This estimation conforms a set of necessary depots. Then, for each number of depots in this set ($b \in \{lb, lb + 1, \ldots, ub - 1, ub\}$), the algorithm selects randomly which depots must be open. Later, the algorithm chooses randomly and feasibly the size to assign to each open depot (s_l), considering only the available discrete sizes. Since this is a random procedure, a known number of iterations is carried out. Hence, each iteration generates an MDVRP instance to be solved, after the depots number, location and size have been selected. Two main decisions must be made in this problem: (i) how to allocate customers to open depots, and (ii) how to design the routes to serve all customers. The allocation problem is solved through a biased-randomized savings heuristic, where the greedy behavior of the heuristic is relaxed (Dominguez et al., 2016). Biased-randomized techniques induce a non-uniform random behavior by using skewed probability distributions. Through this process,

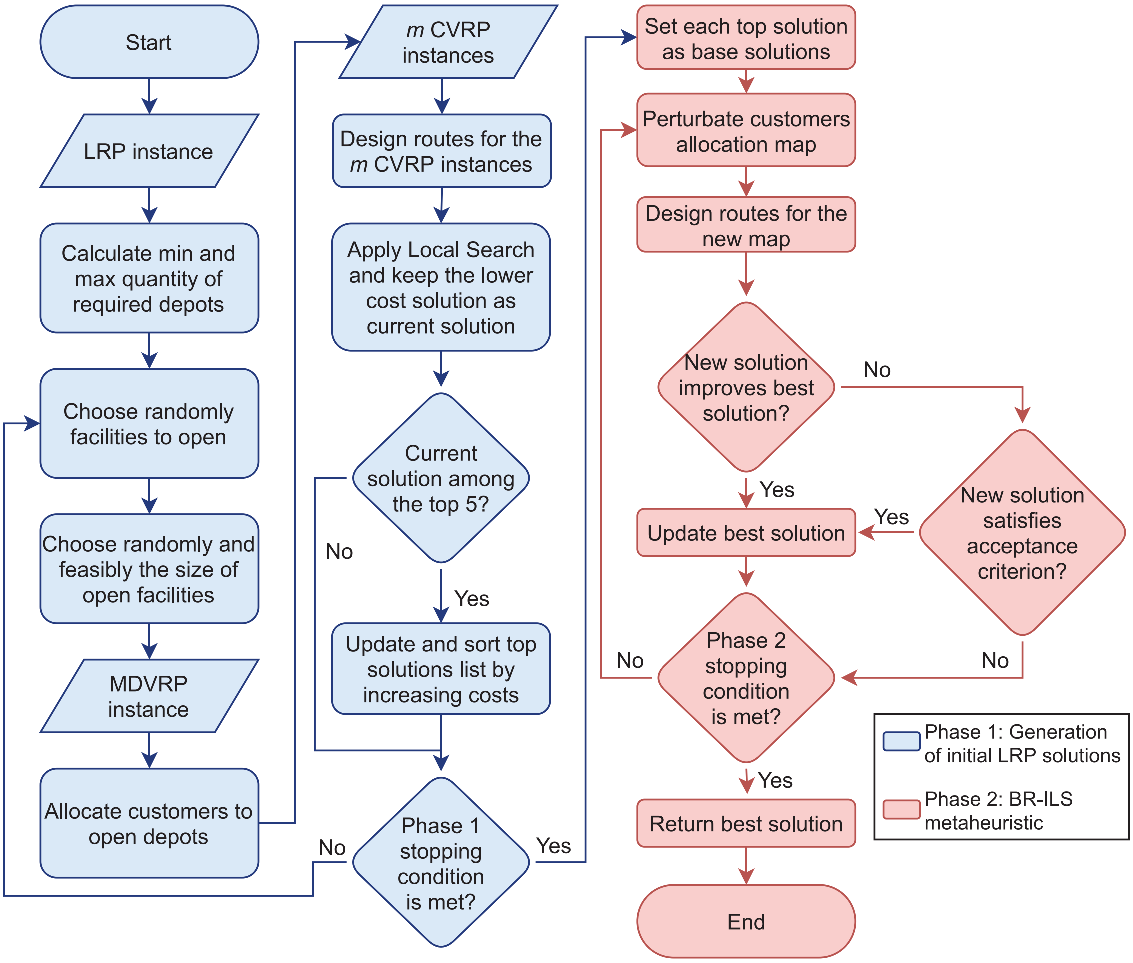

Start
LRP instance
Calculate min and max quantity of required depots
Choose randomly facilities to open
Choose randomly and feasibly the size of open facilities
MDVRP instance
Allocate customers to open depots
m CVRP instances
Design routes for the m CVRP instances
Apply Local Search and keep the lower cost solution as current solution
Current solution among the top 5?
No
Yes
Update and sort top solutions list by increasing costs
Phase 1 stopping condition is met?
No
Yes
Set each top solution as base solutions
Perturbate customers allocation map
Design routes for the new map
New solution improves best solution?
No
Yes
New solution satisfies acceptance criterion?
Yes
No
Update best solution
Phase 2 stopping condition is met?
No
Yes
Return best solution
End
Phase 1: Generation of initial LRP solutions
Phase 2: BR-ILS metaheuristic

Fig. 2. Our biased-randomized metaheuristic.

a deterministic heuristic is transformed into a randomized algorithm whilst preserving the logic behind the original greedy heuristic. The geometric or the triangular probability distributions are useful to guarantee this behaviour. In our algorithm we use the geometric probability distribution, which has only one parameter (β), such that $0 < \beta < 1$. This parameter controls the relative level of greediness present in the randomized behavior of our algorithm, and consequently, introduces the biased randomization process. Notice that biased randomization prevents the same solution from being obtained at every iteration. At the same time, using this biased-randomized procedure ensures that the perturbed solution is not far from the original solution. The savings for the allocation process are calculated following these steps:

1. Calculate the cost (Euclidean distance) c_{ij} between each customer $i \in I$ and each depot $j \in J$.
2. Find the cost (distance) c^*_{ij} between $i \in I$ and $j^* \in J$, where j^* is the depot alternative to j closer to i.

3. Calculate h_{ij}, the marginal savings of allocating the customer $i \in I$ to the depot $j \in J$, instead of the best alternative $j^* \in J$: $h_{ij} = c^*_{ij} - c_{ij}$.

This procedure generates a list of customers and savings for each depot. Positive savings mean that the customer i is closer to the depot j than to any other depot. Hence, if a depot j has several customers with positive savings, the customer with the highest savings is a priority for that depot, given their relative proximity. Then, the list of customers of each depot is sorted in descending order according to the savings. Later, each depot j is selected iteratively to perform a single customer allocation per iteration. Our algorithm selects the next depot according to a purely random policy, as long as its remaining capacity meets the demand of the next customer to allocate. Once the depot j is selected, the customer i is chosen from its savings list. This selection is performed randomly, according to a biased probability distribution, i.e., the first customer in the savings list has the highest probability to be selected, the second element has the second highest probability, and so on. The underlying idea is to preserve the savings criterion as a good heuristic to intensify the search process, and at the same time to provide diversification by enabling the selection of other alternatives.

This procedure generates m submaps, where each submap is formed by one depot and a subset of customers, i.e., m independent capacitated vehicle routing problem (CVRP) instances must be solved. Then, the final decision to obtain a complete LRP solution consists in designing the delivery routes to serve all customers. Several routes can be designed for each submap, depending on the vehicle capacity. We use a biased-randomized version of a savings-based heuristic. The procedure is similar to that used for the allocation decisions. In this case, a savings value is calculated for each edge in the submap, forming a list that is sorted in descending order. Then, each route is iteratively constructed by selecting an element of the list. This selection is carried out randomly by using a biased probability distribution, in the same fashion that in the allocation procedure. Then, a local search procedure is applied to each complete solution. Four local search operators are implemented: (i) a customer swap inter-route operator, where two customers of different routes and allocated to the same depot are swapped; (ii) an inter-depot node exchange, where two customers allocated to different depots are swapped, (iii) a two-opt inter-route operator, where two chains of customers are interchanged between different depots; and (iv) a cross-exchange operator, where three non-consecutive customers from different depots are exchanged. Also, a hash table is used to evaluate each new found CVRP solution. Finally, the Phase 1 is embedded into a multi-start approach (Martí, 2003), which means that it is repeated until the stopping condition is met, saving in memory the top solutions, i.e., those with the minimum cost.

The top solutions are the inputs of the Phase 2. This phase employs the BR-ILS metaheuristic to search for better solutions by performing successive construction and reconstruction processes. Each of the top solutions starts as a base solution whose allocation map is perturbed, i.e., the open depots and their assigned sizes are not modified further. The perturbation procedure consists in selecting a set of customers and trying to reallocate them to another depot, as long as its capacity is not violated. Then, the new allocation map contains a set of m CVRPs to solve through a more intensive biased-randomized savings. As well as in Phase 1, each new solution is both enhanced through the four local search operators and evaluated through a hash table. Whenever a new solution improves the best solution in terms of cost, the latter is updated. Nevertheless, if the best solution is not improved, the new solution is assessed through a Demon-like (Talbi, 2009) acceptance criterion to escape from local optima. Finally, our approach returns the best solution after the stopping condition for the Phase 2 is met.

5. Computational results and discussion

Both new and benchmark LRP instances have been used to test our approach. Ten small-scale instances were created: half has 8 customers and 2 alternative depots and the other half has 10 customers and 3 alternative depots. These instances were solved through an exact method using the MIP model described in Section 3. It yields optimal results useful to compare our algorithm's performance. However, benchmark instances cannot be run efficiently using this model due to their larger size. Three well-known sets of benchmark instances were considered: Akca's (Akca et al., 2009), Barreto's (Barreto et al., 2007) and Prodhon's (Belenguer et al., 2011). Each benchmark instance was slightly modified by introducing five known available sizes for facilities, hence, our algorithm selects a size for each open depot. All experiments were run in a PC with an Intel Core i7 processor and 16 GB RAM, and using Windows 10 as operating system.

5.1. Solving newly created small-sized instances

LRP benchmark instances are usually medium- and large-scale and they are not useful to test our MIP model. Therefore, we created 10 small-scale instances. Most parameters (Table 1) were generated randomly and others were assigned deliberately:

- $I = \{1, 2, 3, \dots, 8\}$ and $J = \{1, 2\}$ for 5 instances.
- $I = \{1, 2, 3, \dots, 10\}$ and $J = \{1, 2, 3\}$ for 5 instances.
- $L = \{1, 2, 3, 4, 5\}$.
- $d_i \sim U(50, 150), \ \forall i \in I$.
- $f_j \sim U(30, 40), \ \forall j \in J$.
- $v \sim U(20, 30)$.
- c_a is established as the Euclidean distance between nodes whose coordinates are $cx_h \sim U(0, 200)$ and $cy_h \sim U(0, 200); \forall h \in I \cup J$.
- $q \sim U(0.5 \sum_i d_i, 0.7 \sum_i d_i)$.
- $s_{jl} \in \{500, 750, 1000, 1250, 1500\} \ \forall j \in J$. Given the importance of facility sizing in our work, these alternatives were fixed. These values also avoid infeasibilities regarding d_i.
- $o_{jl} = \dfrac{s_{jl}}{2 s_{j3}} \cdot \dfrac{\sum_j f_j}{|J|}, \ \forall j \in J, \forall l \in L$. This definition keeps o_{jl} in the same order than f_j and proportional to s_{jl}.
- $M = 999999$. This number is large enough when compared to the magnitude of the rest of the parameters.

Generated instances were solved through both CPLEX and our approach. Given the random nature of our algorithm, 10 random seeds and the following parameters were used to test it:

Table 2
Results comparison between the exact algorithm and our approach

		Exact algorithm		Our approach					
Instance	Total demand	Optimal cost	CPU time (s)	Best found	CPU time (s)	Gap	Quantity of used vehicles	Quantity of open depots	Selected sizes
tor08x2a	767	751.23	0.88	751.23	1.25	0.0%	2	2	{500, 500}
tor08x2b	913	747.05	1.54	747.05	0.86	0.0%	3	2	{500, 1000}
tor08x2c	703	664.56	13.67	664.56	1.08	0.0%	2	2	{500, 500}
tor08x2d	764	606.30	2.96	606.30	0.12	0.0%	2	1	{1000}
tor08x2e	853	815.57	3.70	815.57	0.84	0.0%	2	2	{500, 500}
tor10x3a	1185	878.93	83.60	878.93	1.81	0.0%	2	1	{1250}
tor10x3b	1063	652.50	188.16	652.50	2.72	0.0%	2	2	{500, 750}
tor10x3c	1007	948.99	1028.08	948.99	1.88	0.0%	2	1	{1250}
tor10x3d	976	742.36	19.70	742.36	1.26	0.0%	2	2	{500, 750}
tor10x3e	1125	788.30	31.25	788.30	1.91	0.0%	2	1	{1250}
Average			137.35		1.37	0.0%			

- Iterations for MDVRP instances generation = 5000.
- Iterations for map perturbations = 350.
- Iterations for biased-randomized savings heuristic = 150.
- Iterations for splitting = 150.
- Geometric distribution parameter for biased allocation map in splitting process $(\beta_1) = 0.05 \leq \beta_1 \leq 0.80$.
- Geometric distribution parameter for biased-randomized savings heuristic $(\beta_2) = 0.07 \leq \beta_2 \leq 0.23$.

As is shown in Table 2, our approach reaches the optimal solution for all the tested instances, i.e., the gap between both algorithms is 0.0%. Regarding the computational time, our approach outperforms the exact algorithm for 9 out of 10 instances. Moreover, on average, our approach invests about 99% less computational time to reach the optimal solution, which shows its efficiency.

Results regarding location-routing characteristics show the flexibility provided by our approach. For both 8- and 10-customer instances the algorithm opens 1 or 2 depots, depending on what is less costly. For example, one single depot of size 1000 is open in the instance *tor08x2d*, instead of 2 depots of size 500 as is happening in *tor08x2a*, *tor08x2c* and *tor08x2e*. The total size of both cases is the same, but cost parameters, and customers and potential depots locations determine the quantity of depots to open. For instance, in the case of Fig. 3b, potential depots and most customers are close to each other. Optimal cost is obtained when the depot 2 (D2) is not used, and the depot 1 (D1) is open with a size of 1000 units. Using both depots would increase the opening costs and routing costs savings would be low. In all figures below, a black triangle represents an open depot, and a gray square represents a non-open depot.

An opposite case is showed in Fig. 3a. Potential depots are far from each other and clusters of customers can be identified easily. Both depots are open with a size of 500 each. The additional cost incurred in opening a second depot is made up for routing costs savings. That is, if only one depot were open in this instance, at least one route would be very long. Results displayed in Table 3

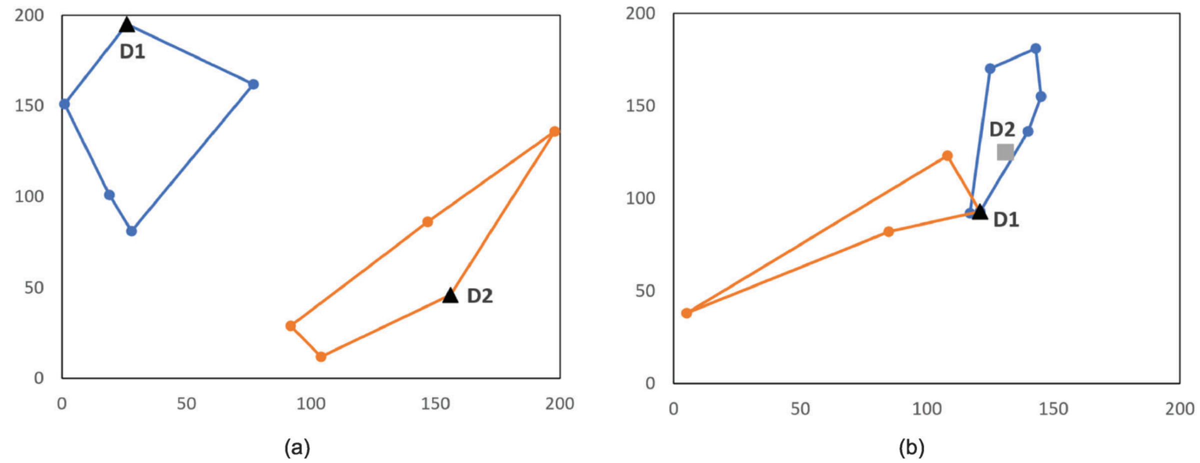


Fig. 3. Optimal location-routing for instances *tor08x2a* (a) and *tor08x2d* (b).

Table 3
Comparison between opening either 1 or 2 depots

		Instance				
	Output	tor08x2a	tor08x2b	tor08x2c	tor08x2d	tor08x2e
Optimal case	Quantity of open depots	2	2	2	1	2
	Selected sizes	{500, 500}	{500, 1000}	{500, 500}	{1000}	{500, 500}
	Opening cost	91.25	104.50	81.25	56.75	78.75
	Routing cost	613.98	579.55	541.31	491.55	688.82
	Vehicle cost	46.00	63.00	42.00	58.00	48.00
	Total cost	751.23	747.05	664.56	606.30	815.57
Modified case	Quantity of open depots	1	1	1	2	1
	Selected sizes	{1000}	{1000}	{750}	{500, 500}	{1000}
	Opening cost	56.25	58.00	45.19	93.75	47.75
	Difference	−38.4%	−44.5%	−44.4%	65.2%	−39.4%
	Routing cost	757.80	740.18	608.38	432.69	807.69
	Difference	23.4%	27.7%	12.4%	−12.0%	17.3%
	Vehicle cost	46.00	42.00	42.00	87.00	48.00
	Difference	0.0%	−33.3%	0.0%	50.0%	0.0%
	Total cost	860.05	840.18	695.57	613.44	903.44
	Difference	14.5%	12.5%	4.7%	1.2%	10.8%

support this idea. This table shows a comparison between the optimal case and a slightly modified case in which the model is forced to open a different number of depots, e.g., if two depots are used to obtain the optimal cost, the modified case shows the situation in which only one depot is open. Instances that reduce the quantity of open depots show higher total cost increases, given that: (i) routing costs always grow in this situation, and (ii) routing costs are always much greater than opening and vehicle costs. The highest total cost difference between modified and optimal case is 14.5% for the instance *tor08x2a*. Table 3 also shows these differences for each cost component. For both

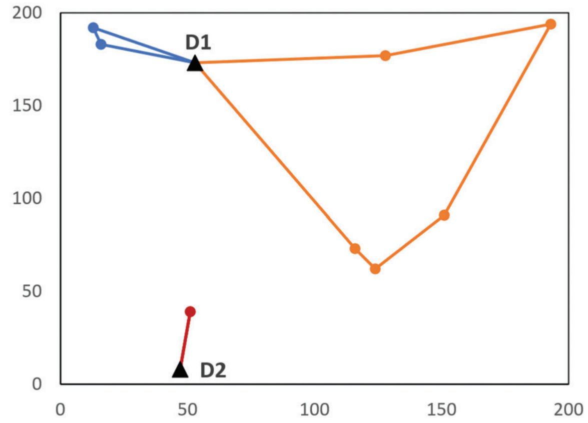


Fig. 4. Optimal location-routing for the instance *tor08x2b*.

tor08x2a and *tor08x2e* instances, selected sizes of 500 and 500 are replaced by an equivalent size of 1000. However, the selected size is 750 for the instance *tor08x2c* in the modified case. Obviously, a 750-size depot is enough to meet a demand of 703, but the use of two depots of size 500 each yields cost savings by generating shorter routes. All these considerations show the flexibility of our approach. For example, consider a traditional LRP with fixed sizes of 1000. Costs would be always higher for the instance *tor08x2c*, regardless of the number of open depots.

A particular case is identified for the instance *tor08x2b*. Figure 4 shows the optimal location-routing plan. Total served demand is: *blue route = 189*, *orange route = 644*, and *red route = 80*. That is, D1 meets a total demand of 833, and D2, a total demand of 80. Minimal available depot sizes that satisfy feasibly such demands are 1000 and 500, respectively. Therefore, the total demand is 913 and the total capacity is 1500, which exceeds demand in about 65%. That is, one single depot of size 1000, two depots of size 500, or even two depots of sizes 500 and 750 each would be theoretically enough, although routing costs would increase. For instance, the modified case in Table 3 shows that routing costs are 27.7% higher when one single 1000-size depot is open. If the instance *tor08x2b* were a real-world case, a decision-maker may formulate the question if opening D2 is worth, since its used capacity is only 16% and D1 can meet the whole demand. In terms of total costs, it is really worth since the modified case in Table 3 shows a total cost that is 12.5% higher. Opening and vehicle costs decrease but routing costs increase. Besides, when mid- and long-term planning is considered, demand can change over time and D2 may become necessary. A final test with the MIP model was done. It shows that parameters f_j and o_{jl} must be at least 3.1 times the original values to open only one depot in the optimal case. That is, f_j and o_{jl} must be at least 3.1 times higher to consider that opening a second depot is not worth.

5.2. Solving medium- and large-sized benchmark instances

Known LRP benchmark instances have been used to test our approach. Nevertheless, traditional algorithms using them assume that the depot size is fixed, i.e., algorithms choose if a depot is open or not, and if it is, only one alternative size is available to assign. This increases costs and decreases flexibility in decision making, as we will demonstrate below. Since benchmark instances have a single value for the size per potential depot, they were slightly modified to introduce new alternative sizes. 5 alternatives were considered: the original size parameter in the instance, 2 sizes smaller than the original, and 2 sizes greater than the original. If s_j is the original size for each potential depot, each available size is given by the elements in the set: $s_{jl} \in \{(1-2r)s_j, (1-r)s_j, s_j, (1+r)s_j, (1+2r)s_j\}$, where r is the variability range between available sizes, and $0.0 < r < 0.5$. For these initial experiments, $r = 0.25$. Other values of r are considered in Section 5.3.

The variable *Opening cost* (o_{jl}) is another non-considered parameter in benchmark instances. We calculate this parameter according to Equation (13). This equation keeps o_{jl} in the same order than f_j and allows to assign negative variable opening costs for sizes smaller than the original, positive costs for sizes greater than the original, and zero variable cost for the original size. The goal of this definition is to compare properly our results with those obtained when using the traditional benchmark instances in previous LRP papers. Finally, the number of iterations and geometric distribution's parameters are the same as those in Section 5.1:

$$o_{jl} = \frac{s_{jl} - s_j}{2s_j} \cdot \frac{\sum_j f_j}{|J|}, \quad \forall j \in J, \forall l \in L \tag{13}$$

Our approach results were compared with those obtained by Quintero-Araujo et al. (2019a) in the so-called *Fully cooperative scenario* (a traditional LRP). This paper was chosen since it does not show only a total cost per instance but also details about cost components. Table 4 shows this comparison for Akca's and Barreto's instances, and Table 5 shows it for Prodhon's instances. A total of 59 instances were tested. In terms of *Total costs*, our results always outperform Quintero-Araujo's, except for the Barreto's instance *Gas-32x5b*, in which we attain a slight positive gap of 0.09%. The rest of the instances shows a negative gap, i.e., we obtain smaller costs by allowing the selection of size for each facility. The last two columns of Tables 4 and 5 show that our results also outperform the best-known solution (BKS) for most original instances. Small positive gaps were obtained for only 6 out of 59 instances. Hence, for Barreto's and Prodhon's instances the average gap between our results and Quintero-Araujo's are −4.32% and −7.54%, respectively, which are greater than our average gap in regard to the BKS (−3.90% and −7.12%, respectively). The average gap is the same for Akca's instances (−3.12%).

Cost components are also shown, namely: *Opening*, *Routing* and *Vehicle costs*. However, since the input parameter v is not equal to zero only in Prodhon's instances, the *Vehicle cost* is not included in Table 4. Our approach decreases the *Opening cost* on 44 out of 59 instances, which is a direct consequence of offering several alternative sizes. For example, the BKS for the Akca's instance *Cr30x5b-2* opens two depots of size 1000. Our approach finds that opening one depot of size 750 and one depot of size 1000 is enough, generating savings of 6.25% in the opening cost. In fact, more than half of the instances attains opening cost savings of at least 18%, with a maximum of 46.38% on the Prodhon's instance *Coord20-5-1b*.

This instance is very useful to illustrate what is happening. The total demand is 308 in this case, and the only available size is originally 300. Therefore, the traditional LRP needs to open at least 2 depots to meet such demand, with a total size of 600. Our flexible approach only requires to open a single bigger depot. Since differences between available sizes are 25% ($r = 0.25$) for the experiments

Table 4
Results on Akca's and Barreto's instances

	Quintero-Araujo et al. (2019a)			Our approach					Gap			BKS	
Instance	Opening cost	Routing cost	Total cost	Opening cost	Routing cost	Total Cost	Selected sizes	CPU Time (s)	Opening Cost	Routing Cost	Total Cost	Total Cost	Gap
Cr30x5a-1	200.0	619.51	819.51	200.00	575.14	775.14	{500, 1500}	25.87	0.00%	−7.16%	−5.41%	819.51	−5.41%
Cr30x5a-2	200.0	621.45	821.45	200.00	607.28	807.28	{750, 1250}	26.30	0.00%	−2.28%	−1.73%	821.45	−1.73%
Cr30x5a-3	200.0	502.29	702.29	187.50	507.92	695.42	{750, 1000}	21.56	−6.25%	1.12%	−0.98%	702.29	−0.98%
Cr30x5b-1	200.0	680.03	880.03	225.00	623.22	848.22	{500, 500, 500}	20.71	12.50%	−8.35%	−3.61%	880.02	−3.61%
Cr30x5b-2	200.0	625.32	825.32	187.50	625.32	812.82	{750, 1000}	18.86	−6.25%	0.00%	−1.51%	825.32	−1.51%
Cr30x5b-3	200.0	684.58	884.58	187.50	684.58	872.08	{750, 1000}	19.04	−6.25%	0.00%	−1.41%	884.58	−1.41%
Cr40x5a-1	200.0	728.10	928.10	162.50	731.84	894.34	{875, 1312}	43.19	−18.75%	0.51%	−3.64%	928.10	−3.64%
Cr40x5a-2	200.0	688.42	888.42	225.00	637.26	862.26	{875, 875, 875}	39.46	12.50%	−7.43%	−2.95%	888.42	−2.94%
Cr40x5a-3	200.0	747.26	947.26	162.50	752.88	915.38	{875, 1312}	31.92	−18.75%	0.75%	−3.37%	947.26	−3.37%
Cr40x5b-1	200.0	852.04	1052.04	162.50	852.04	1014.54	{875, 1312}	38.06	−18.75%	0.00%	−3.56%	1052.04	−3.56%
Cr40x5b-2	200.0	781.54	981.54	225.00	690.57	915.57	{875, 875, 875}	34.18	12.50%	−11.64%	−6.72%	981.54	−6.72%
Cr40x5b-3	200.0	764.33	964.33	175.00	764.33	939.33	{875, 1750}	34.16	−12.50%	0.00%	−2.59%	964.33	−2.59%
							Average Akca's	**29.44**	**−4.17%**	**−2.87%**	**−3.12%**		**−3.12%**
Perl-12x2	100.0	103.98	203.98	100.00	103.98	203.98	{280}	0.21	0.00%	0.00%	0.00%	203.98	0.00%
Gas-21x5	100.0	324.90	424.90	93.75	324.90	418.65	{112, 150}	10.22	−6.25%	0.00%	−1.47%	424.90	−1.47%
Gas-22x5	50.0	535.11	585.11	43.75	535.11	578.86	{11250}	3.52	−12.50%	0.00%	−1.07%	585.10	−1.07%
Min-27x5	544.0	2518.02	3062.02	442.00	2518.02	2960.02	{450, 675}	12.87	−18.75%	0.00%	−3.33%	3062.00	−3.33%
Gas-29x5	100.0	412.10	512.10	81.25	412.10	493.35	{7500, 11250}	14.89	−18.75%	0.00%	−3.66%	512.10	−3.66%
Gas-32x5	50.0	512.22	562.22	75.00	476.20	551.20	{17500, 17500}	22.24	50.00%	−7.03%	−1.96%	562.20	−1.96%
Gas-32x5b	50.0	454.33	504.33	50.00	454.77	504.77	{35000}	6.92	0.00%	0.10%	0.09%	504.33	0.09%
Gas-36x5	50.0	425.99	475.99	37.50	420.83	458.33	{300}	27.27	−25.00%	−1.21%	−3.71%	460.37	−0.44%
Chr-50x5ba	80.0	485.62	565.62	60.00	489.59	549.59	{5000, 5000}	51.51	−25.00%	0.82%	−2.83%	565.62	−2.83%
Chr-50x5be	80.0	485.60	565.60	60.00	485.60	545.60	{5000, 5000}	47.95	−25.00%	0.00%	−3.54%	565.60	−3.54%
Perl-55x15	720.0	392.38	1112.38	510.00	417.90	927.90	{550, 687}	78.01	−29.17%	6.50%	−16.58%	1112.06	−16.56%
Chr-75x10ba	120.0	730.13	850.13	90.00	705.87	795.87	{5000, 5000, 5000}	114.21	−25.00%	−3.32%	−6.38%	844.40	−5.75%
Perl-85x7	1116.0	509.08	1625.08	790.5	538.39	1328.89	{850, 1062}	200.83	−29.17%	5.76%	−18.23%	1622.50	−18.10%
Das-88x8	83.6	272.68	356.28	75.88	260.84	336.72	{12500000, 18750000, 18750000}	312.85	−9.23%	−4.34%	−5.49%	355.78	−5.36%
Chr-100x10	80.0	760.52	840.52	60.00	759.31	819.31	{750, 750}	466.58	−25.00%	−0.16%	−2.52%	833.40	−1.69%
Min-134x8	804.0	4993.53	5797.53	804.00	4989.06	5793.06	{3000, 3000, 3000}	852.19	0.00%	−0.09%	−0.08%	5709.00	1.47%
Das-150x10	15000.0	29190.59	44190.59	11875.00	31145.92	43020.92	{37500000, 45000000}	1918.82	−20.83%	6.70%	−2.65%	43919.90	−2.05%
							Average Barreto's	**243.59**	**−12.92%**	**0.22%**	**−4.32%**		**−3.90%**

Table 5
Results on Prodhon's instances

	Quintero-Araujo et al. (2019a)				Our approach						Gap				BKS	
Instance	Opening cost	Routing cost	Vehicle cost	Total cost	Opening cost	Routing cost	Vehicle cost	Total cost	Selected sizes	CPU time (s)	Opening cost	Routing cost	Vehicle cost	Total cost	Total cost	Gap
Coord20-5-1	25549	24472	5000	55021	20250	25929	5000	51179	{140, 210}	7.06	−20.74%	5.95%	0.00%	−6.98%	54793	−6.60%
Coord20-5-1b	15497	20607	3000	39104	8310	23688	3000	34998	{375}	8.10	−46.38%	14.95%	0.00%	−10.50%	39104	−10.50%
Coord20-5-2	24196	19712	5000	48908	16940	21499	5000	43439	{140, 210}	7.21	−29.99%	9.07%	0.00%	−11.18%	48908	−11.18%
Coord20-5-2b	13911	20631	3000	37542	9322	21081	3000	33403	{375}	5.43	−32.99%	2.18%	0.00%	−11.02%	37542	−11.02%
Coord50-5-1	25442	52822	12000	90264	21780	52669	12000	86449	{262, 262, 315}	45.30	−14.39%	−0.29%	0.00%	−4.23%	90111	−4.06%
Coord50-5-1b	15385	41908	6000	63293	15385	41180	6000	62565	{262, 525}	55.64	0.00%	−1.74%	0.00%	−1.15%	63242	−1.07%
Coord50-5-2	29319	46979	12000	88298	19420	48690	12000	80110	{262, 525}	34.44	−33.76%	3.64%	0.00%	−9.27%	88298	−9.27%
Coord50-5-2b	29319	32314	6000	67633	19420	31494	6000	56914	{262, 525}	36.04	−33.76%	−2.54%	0.00%	−15.85%	67308	−15.44%
Coord50-5-2BIS	19785	52871	12000	84656	15964	50982	12000	78946	{437, 437}	34.02	−19.31%	−3.57%	0.00%	−6.75%	84055	−6.08%
Coord50-5-2bBIS	18763	27120	6000	51883	14929	22437	6000	43366	{450, 450}	30.33	−20.44%	−17.27%	0.00%	−16.42%	51822	−16.32%
Coord50-5-3	18961	55307	12000	86268	18961	55114	12000	86075	{315, 525}	59.69	0.00%	−0.35%	0.00%	−0.22%	86203	−0.15%
Coord50-5-3b	18961	37021	6000	61982	10711	44273	6000	60984	{210, 630}	60.78	−43.51%	19.59%	0.00%	−1.61%	61830	−1.37%
Coord100-5-1	132890	120083	24000	276973	96994	128451	24000	249445	{770, 875}	361.22	−27.01%	6.97%	0.00%	−9.94%	274814	−9.23%
Coord100-5-1b	132890	71008	12000	215898	96994	74036	11000	182030	{770, 875}	526.38	−27.01%	4.26%	−8.33%	−15.69%	213615	−14.79%
Coord100-5-2	102246	70248	24000	196494	102246	68384	24000	194630	{770, 840}	239.07	0.00%	−2.65%	0.00%	−0.95%	193671	0.50%
Coord100-5-2b	102246	44427	11000	157673	102246	43864	11000	157110	{770, 840}	272.53	0.00%	−1.27%	0.00%	−0.36%	157095	0.01%
Coord100-5-3	88287	89260	24000	201547	88287	89277	23000	200564	{770, 840}	378.62	0.00%	0.02%	−4.17%	−0.49%	200079	0.24%
Coord100-5-3b	88287	54002	11000	153289	88287	53205	11000	152492	{770, 840}	450.39	0.00%	−1.48%	0.00%	−0.52%	152441	0.03%
Coord100-10-1	165068	102669	26000	293737	133684	114980	24000	272664	{840, 840}	360.97	−19.01%	11.99%	−7.69%	−7.17%	287695	−5.22%
Coord100-10-1b	154942	69510	12000	236452	133684	66873	12000	212557	{840, 840}	381.92	−13.72%	−3.79%	0.00%	−10.11%	230989	−7.98%
Coord100-10-2	149586	66641	23000	239227	120190	72683	23000	215873	{735, 840}	196.20	−19.65%	9.07%	0.00%	−9.76%	243590	−11.38%
Coord100-10-2b	149586	42456	13000	205042	120190	44604	11000	175794	{735, 840}	236.33	−19.65%	5.06%	−15.38%	−14.26%	203988	−13.82%
Coord100-10-3	136123	92082	25000	253205	112079	100924	23000	236003	{735, 840}	338.44	−17.66%	9.60%	−8.00%	−6.79%	250882	−5.93%
Coord100-10-3b	136123	52008	11000	199131	112079	58267	11000	181346	{735, 840}	301.97	−17.66%	12.03%	0.00%	−8.93%	204317	−11.24%
Coord200-10-1	266151	164764	47000	477915	241539	182115	47000	470654	{455, 892, 1785}	2577.42	−9.25%	10.53%	0.00%	−1.52%	475294	−0.98%
Coord200-10-1b	253840	104306	22000	380146	196905	133469	21000	351374	{1365, 1785}	4654.19	−22.43%	27.96%	−4.55%	−7.57%	377043	−6.81%
Coord200-10-2	280370	123205	48000	451575	208501	162640	47000	418141	{1225, 1890}	717.70	−25.63%	32.01%	−2.08%	−7.40%	449006	−6.87%
Coord200-10-2b	280370	72608	23000	375978	208501	89469	22000	319970	{1225, 1890}	1146.87	−25.63%	23.22%	−4.35%	−14.90%	374280	−14.51%
Coord200-10-3	272528	156128	46000	474656	202277	202273	46000	450550	{1470, 1680}	3367.43	−25.78%	29.56%	0.00%	−5.08%	469433	−4.02%
Coord200-10-3b	234660	109891	22000	366551	193088	116413	22000	331501	{1400, 1680}	611.54	−17.72%	5.93%	0.00%	−9.56%	362653	−8.59%
									Average Prodhon's	**583.44**	**−19.44%**	**6.96%**	**−1.82%**	**−7.54%**		**−7.12%**

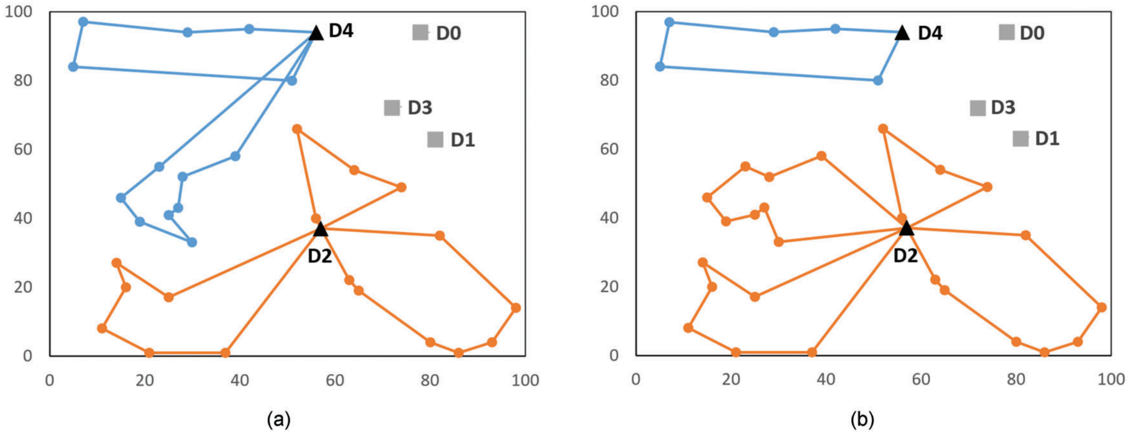


Fig. 5. Best found solution by the non-flexible LRP (a) and our approach (b) for the Akca's instance *Cr30x5a-1*.

in this section, the chosen size by our approach is 375, which is the minimum available size to meet a demand of 308. Notice in Table 5 that our approach increases the routing cost in 14.95% for this instance, although the total cost remains lower than Quintero-Araujo's in 10.50%. The explanation is the same as in Section 5.1: if one depot is open instead of two, more and longer routes must be designed, increasing routing costs and generating savings in opening costs. These savings are greater than the increase in routing costs.

Total costs decrease is not only a consequence of the reduction in opening costs. 4 out of 59 instances show an increase in these costs because of selecting bigger facilities, which results in a drop in routing costs. Moreover, 8 instances show 0.00% in opening costs savings but still routing costs decrease. The Akca's instance *Cr30x5a-1* is an example of this situation. The total demand to meet is 1662. The original non-flexible best solution is 819.51 ($Opening\ cost = 200.00$ and $Routing\ cost = 619.51$), by opening 2 depots with a size of 1000 each. Designed routes are shown in Fig. 5a. Our approach attains the same opening cost by opening the same depots than the original LRP but assigning them sizes of 500 and 1500 for D4 and D2, respectively. Given our formula for costs calculation in Equation (13), a total size of 1000 + 1000 costs the same as a total size of 500 + 1500, but conditions may be different in real-world problems, depending on the cost structure of each company or supply chain. Regardless of this situation, assigning different depot sizes leads to design better routes, as can be seen in Fig. 5b. Our routing cost is 575.14, since the depot D2 has now more capacity to serve some customers that are closer to it than to the depot D4. This shows the flexibility and cost-efficiency of our approach.

Prodhon's instances consider also a vehicle cost. Such consideration leads to reduce total costs not only by decreasing opening costs or traveled distance, but also by reducing the number of routes. 8 out of 30 instances show this performance, which is a direct consequence of the flexibility in facility sizing. For example, our approach creates one route less than Quintero-Araujo et al. (2019a) for the instance *Coord100-5-1b*. Our algorithm opens 2 depots with capacities of 770 and 875, respectively, whereas the non-flexible approach opens 3 depots with capacities of 700, 770, and 770, respectively. As our open facilities are bigger, only 2 depots are necessary and, therefore, the algorithm finds more flexibility to distribute the customers differently by using less vehicles. Obviously,

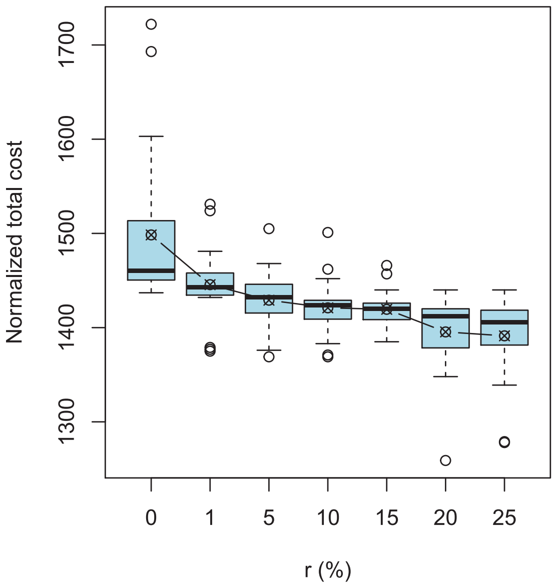


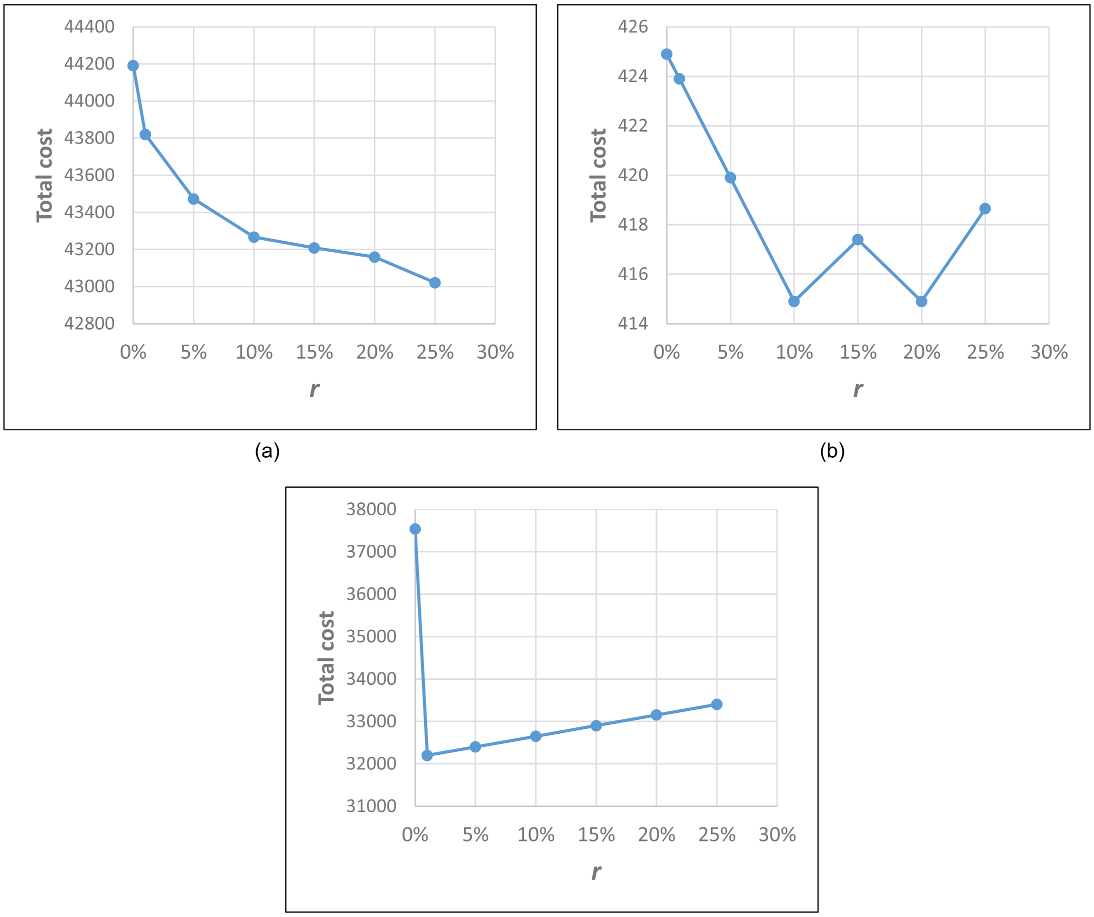
Fig. 6. Variation in total costs in relation to *r*.

this is also subject to the capacity of vehicles. The non-flexible approach yields an average vehicle utilization of 87.9%, which means a margin for improvement. The bigger facilities in our approach leads to a reorganization of the routes and an average vehicle utilization of 95.9%.

5.3. *Sensitivity analysis regarding available sizes*

So far, the variability range between available sizes remained fixed in 25%, i.e., $r = 0.25$. This section's objective is to analyze the effect that other values of r have in the obtained results. Hence, 7 values of r are considered, namely: $r \in \{0.00, 0.01, 0.05, 0.10, 0.15, 0.20, 0.25\}$. The case in which $r = 0.00$ corresponds to the non-flexible case shown by Quintero-Araujo et al. (2019a). A total of 20 Akca's, Barreto's and Prodhon's instances were selected to carry out our experiments. These instances show different number of customers and depots from each other. Regardless of the instance, the highest total cost is always obtained when $r = 0.00$. In average, the total costs show a decreasing trend when increasing r, as Fig. 6 displays. As costs in all instances have very different scales, they were normalized to create this chart. These results indicate that providing sizes with broader variability has a positive impact in total costs, i.e., the greater the differences among input sizes, the smaller the average costs. Fig. 6 demonstrates the advantages of considering our flexible approach, since even a range as small as 1% in available sizes is enough to yield cost savings.

It is important to highlight that this is an average performance, which means that most instances show a total cost decrease when increasing r. However, some individual instances do not perform in this way. Whenever considering $r > 0$, three types of results are identified: (i) 12 instances show a steady decrease in total cost; (ii) 7 instances show a fluctuating performance; and (iii) 1 instance's total cost increases steadily with r. Fig. 7 shows an example of each case. The chart (a) represents



(c)

Fig. 7. Variation of the total cost with *r* for instances *Das-150x10* (a), *Gas-21x5* (b), and *Coord20-5-2b* (c).

the general trend in which providing sizes with a broader variability yields smaller total costs, the chart (b) shows a performance with no clear trend, and the chart (c) shows the only instance with an opposite performance. The relation between the total demand and available sizes is the cause of such behavior. For example, the total demand in the instance *Coord20-5-2b* is 302. The available original size is 300, i.e., at least 2 depots are necessary to meet the demand when $r = 0$. However, if a size 1% bigger is available, only one depot is enough. In this case, providing still bigger sizes is redundant and, therefore, opening costs increase with r. The underlying idea is that the algorithm searches for a total capacity as close as possible to the total demand in order to minimize the total cost. Nevertheless, such as real-world cases show, s_{jl} is not usually a continuous parameter. Hence, our metaheuristic tries to find the less-costly combination of available sizes so that total demand is met. Most times, providing sizes with a bigger range helps to attain this objective, but sometimes, they cannot be combined so that the total capacity is closer to the total demand. This also explains the fluctuating performance of the instance *Gas-21x5*, as observed in Fig. 7b.

6. Conclusions

Despite its importance in modern applications (e.g., last-mile delivery logistics, e-commerce, or 5G telecommunication networks), the location routing problem (LRP) with facility sizing decisions has been scarcely studied in the literature. As the original LRP, the heterogeneous version is also *NP-hard*, which justifies the use of metaheuristics when solving large-sized instances. Our work proposes a biased-randomized iterated local search (BR-ILS) metaheuristic. After running a series of tests, the associated results show the great advantages of considering facility sizing decisions instead of having a fixed value as traditional approaches do. Noticeable cost savings are obtained with our approach due to: (i) the possibility of designing customized facilities that adjust to the current and forecasted demand in each region; and (ii) reallocating customers and redesigning routes by locating either larger- or smaller-size facilities. Both alternatives have been proved to decrease total costs, which are formed by opening costs (investment capital), and operational costs (routing and vehicle costs). The former alternative allows to save routing costs, although the opening cost can grow. The latter alternative may increase routing costs, but the initial investment is lower.

Regardless of the size of the instance, our approach has been proved to yield very competitive results in terms of total costs. Small-, medium-, and large-sized instances have been used in our experiments. Initially, three mixed-integer linear programming (MIP) models are proposed and tested by solving optimally a few newly created small-sized instances, as well as benchmark instances whose number of nodes is smaller than 30. The same instances were solved using our BR-ILS metaheuristic. Then, this approach is employed to solve both medium- and large-sized benchmark instances. They were slightly modified to consider facility sizing decisions, by providing both a set of alternative sizes and a variable cost according to each size. The experiments' results show not only that cost savings are attained after considering flexibility in facility sizes, but also that our metaheuristic is both time- and cost-efficient. Additionally, all proposed MIP models have been proved to be quite inefficient when compared with our metaheuristic approach. Finally, a sensitivity analysis is carried out, in which we study the effect of different sets of sizes in the cost. Average results show that those sets with a bigger range of difference between sizes yield smaller total costs. Nevertheless, a few instances do not follow this trend, which indicates that total costs in an LRP with facility sizing decisions depend on the relation between demand and available sizes for each instance or real-world case.

Future work in this field may include: (i) random variations affecting sizing decisions, i.e., the consideration of demand stochasticity from the phase in which facilities are designed; and (ii) a robustness and resilience analysis, given that smaller facilities are designed. This may affect the system flexibility if serious disruption events affect the supply chain. Finally, the effect of a non-linear cost variation as a function of the facility size can also be studied, since each real-world supply chain faces different market conditions.

Acknowledgment

This work has been supported by the Doctoral Schools of the Universitat Oberta de Catalunya (Spain) and the Universidad de La Sabana (ColombBerlin,ia, Grants INGPhD-12-2020, INGPhD-10-2019), by the Spanish Ministry of Science (PID2019-111100RB-C21, RED2018-102642-T), by the EU-IoF2020 (282300206-UC005), and by a mobility fellowship from the Ibero-American Association of Postgraduate Universities (AUIP).

References

Akca, Z., Berger, R., Ralphs, T., 2009. A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In Chinneck, J.W., Kristjansson, B., Saltzman, M.J. (eds) *Operations Research and Cyber-Infrastructure*. Springer, pp. 309–330.

Albareda-Sambola, M., Rodríguez-Pereira, J., 2019. Location-routing and location-arc routing. In Laporte, G., Nickel, S., Saldanha da Gama, F. (eds) *Location Science*. Springer, Berlin, pp. 431–451.

Alumur, S., Kara, B.Y., 2007. A new model for the hazardous waste location-routing problem. *Computers & Operations Research* 34, 5, 1406–1423.

Aykin, T., 1995. The hub location and routing problem. *European Journal of Operational Research* 83, 1, 200–219.

Badri, H., Bashiri, M., Hejazi, T.H., 2013. Integrated strategic and tactical planning in a supply chain network design with a heuristic solution method. *Computers & Operations Research* 40, 4, 1143–1154.

Baldacci, R., Mingozzi, A., Wolfler Calvo, R., 2011. An exact method for the capacitated location-routing problem. *Operations Research* 59, 5, 1284–1296.

Barreto, S., Ferreira, C., Paixao, J., Santos, B.S., 2007. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research* 179, 3, 968–977.

Bashiri, M., Badri, H., Talebi, J., 2012. A new approach to tactical and strategic planning in production–distribution networks. *Applied Mathematical Modelling* 36, 4, 1703–1717.

Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Calvo, R.W., 2011. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research* 38, 6, 931–941.

Correia, I., Melo, T., 2016. Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research* 255, 3, 729–746.

Correia, I., Melo, T., 2017. A multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment. *Computers & Industrial Engineering* 110, 307–321.

Cortinhal, M.J., Lopes, M.J., Melo, M.T., 2015. Dynamic design and re-design of multi-echelon, multi-product logistics networks with outsourcing opportunities: a computational study. *Computers & Industrial Engineering* 90, 118–131.

Dominguez, O., Juan, A.A., De La Nuez, I., Ouelhadj, D., 2016. An ILS-biased randomization algorithm for the two-dimensional loading HFVRP with sequential loading and items rotation. *Journal of the Operational Research Society* 67, 1, 37–53.

Drexl, M., Schneider, M., 2015. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research* 241, 2, 283–308.

Ferrer, A., Guimarans, D., Ramalhinho, H., Juan, A.A., 2016. A BRILS metaheuristic for non-smooth flow-shop problems with failure-risk costs. *Expert Systems with Applications* 44, 177–186.

Gonzalez-Martin, S., Juan, A.A., Riera, D., Elizondo, M.G., Ramos, J.J., 2018. A simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation* 12, 1, 53–66.

Govindan, K., Jafarian, A., Khodaverdi, R., Devika, K., 2014. Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics* 152, 9–28.

Grasas, A., Juan, A.A., Faulin, J., de Armas, J., Ramalhinho, H., 2017. Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering* 110, 216–228.

Gruler, A., Fikar, C., Juan, A.A., Hirsch, P., Contreras-Bolton, C., 2017. Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization. *Journal of Simulation* 11, 1, 11–19.

Guimarans, D., Dominguez, O., Panadero, J., Juan, A.A., 2018. A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory* 89, 1–14.

Hemmelmayr, V., Smilowitz, K., de la Torre, L., 2017. A periodic location routing problem for collaborative recycling. *IISE Transactions* 49, 4, 414–428.

Jacobsen, S., Madsen, O., 1980. A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research* 5, 6, 378–387.

Juan, A.A., Corlu, C.G., Tordecilla, R.D., de la Torre, R., Ferrer, A., 2020. On the use of biased-randomized algorithms for solving non-smooth optimization problems. *Algorithms* 13, 1, 8.

Laporte, G., Louveaux, F., Mercure, H., 1989. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research* 39, 1, 71–78.

Laporte, G., Nobert, Y., Arpin, D., 1986. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research* 6, 9, 291–310.

Laporte, G., Nobert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science* 22, 3, 161–172.

Lashine, S.H., Fattouh, M., Issa, A., 2006. Location/allocation and routing decisions in supply chain network design. *Journal of Modelling in Management* 1, 2, 173–183.

Le, C.T.D., Buddhakulsomsiri, J., Jeenanunta, C., Dumrongsiri, A., 2019. Determining an optimal warehouse location, capacity, and product allocation in a multi-product, multi-period distribution network: a case study. *International Journal of Logistics Systems and Management* 34, 4, 510–532.

Londoño, J.C., Tordecilla, R.D., Martins, L.d.C., Juan, A.A., 2020. A biased-randomized iterated local search for the vehicle routing problem with optional backhauls. *TOP* 29, 387–416.

Lopes, R.B., Ferreira, C., Santos, B.S., Barreto, S., 2013. A taxonomical analysis, current methods and objectives on location-routing problems. *International Transactions in Operational Research* 20, 6, 795–822.

Lourenço, H.R., Martin, O.C., Stützle, T., 2019. Iterated local search: framework and applications. In Gendreau, M., Potvin, J.Y. (eds) *Handbook of Metaheuristics*. Springer, Berlin, pp. 129–168.

Madsen, O., 1983. Methods for solving combined two level location-routing problems of realistic dimensions. *European Journal of Operational Research* 12, 3, 295–301.

Maranzana, F., 1964. On the location of supply points to minimize transport costs. *Journal of the Operational Research Society* 15, 3, 261–270.

Martí, R., 2003. Multi-start methods. In Glover, F., Kochenberger, G.A. (eds) *Handbook of Metaheuristics*. Springer, Berlin, pp. 355–368.

Muñoz-Villamizar, A., Quintero-Araújo, C.L., Montoya-Torres, J.R., Faulin, J., 2019. Short-and mid-term evaluation of the use of electric vehicles in urban freight transport collaborative networks: a case study. *International Journal of Logistics Research and Applications* 22, 3, 229–252.

Nagy, G., Salhi, S., 2007. Location-routing: issues, models and methods. *European Journal of Operational Research* 177, 2, 649–672.

Nataraj, S., Ferone, D., Quintero-Araujo, C., Juan, A.A., Festa, P., 2019. Consolidation centers in city logistics: a cooperative approach based on the location routing problem. *International Journal of Industrial Engineering Computations* 10, 3, 393–404.

Perl, J., Daskin, M., 1985. A warehouse location-routing problem. *Transportation Research Part B* 19, 5, 381–396.

Prins, C., Prodhon, C., Calvo, R.W., 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR* 4, 3, 221–238.

Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Calvo, R.W., 2007. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. *Transportation Science* 41, 4, 470–483.

Prodhon, C., Prins, C., 2014. A survey of recent research on location-routing problems. *European Journal of Operational Research* 238, 1, 1–17.

Quintero-Araujo, C.L., Caballero-Villalobos, J.P., Juan, A.A., Montoya-Torres, J.R., 2017. A biased-randomized metaheuristic for the capacitated location routing problem. *International Transactions in Operational Research* 24, 5, 1079–1098.

Quintero-Araujo, C.L., Gruler, A., Juan, A.A., Faulin, J., 2019a. Using horizontal cooperation concepts in integrated routing and facility-location decisions. *International Transactions in Operational Research* 26, 2, 551–576.

Quintero-Araujo, C.L., Guimarans, D., Juan, A.A., 2019b. A simheuristic algorithm for the capacitated location routing problem with stochastic demands. *Journal of Simulation* 0, 1–18.

Rabbani, M., Heidari, R., Yazdanparast, R., 2019. A stochastic multi-period industrial hazardous waste location-routing problem: integrating NSGA-II and Monte Carlo simulation. *European Journal of Operational Research* 272, 3, 945–961.

Samanlioglu, F., 2013. A multi-objective mathematical model for the industrial hazardous waste location-routing problem. *European Journal of Operational Research* 226, 2, 332–340.

Sauvey, C., Melo, T., Correia, I., 2020. Heuristics for a multi-period facility location problem with delayed demand satisfaction. *Computers & Industrial Engineering* 139, 106171.

Schneider, M., Drexl, M., 2017. A survey of the standard location-routing problem. *Annals of Operations Research* 259, 1-2, 389–414.

Shulman, A., 1991. An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes. *Operations Research* 39, 3, 423–436.

Talbi, E.G., 2009. *Metaheuristics: From Design to Implementation*, Vol. 74, John Wiley & Sons, New York.

Tordecilla, R.D., Copado-Méndez, P.J., Panadero, J., Quintero-Araujo, C.L., Montoya-Torres, J.R., Juan, A.A., 2021. Combining heuristics with simulation and fuzzy logic to solve a flexible-size location routing problem under uncertainty. *Algorithms* 14, 2, 23.

Tordecilla, R.D., Panadero, J., Quintero-Araujo, C.L., Montoya-Torres, J.R., Juan, A.A., 2020. A simheuristic algorithm for the location routing problem with facility sizing decisions and stochastic demands. In *Proceedings of the 2020 Winter Simulation Conference*, IEEE, Piscataway, New Jersey, pp. 1265–1275.

Tordecilla-Madera, R., Polo, A., Muñoz, D., González-Rodríguez, L., 2017. A robust design for a colombian dairy cooperative's milk storage and refrigeration logistics system using binary programming. *International Journal of Production Economics* 183, 710–720.

Tunalıoğlu, R., Koç, Ç., Bektaş, T., 2016. A multiperiod location-routing problem arising in the collection of olive oil mill wastewater. *Journal of the Operational Research Society* 67, 7, 1012–1024.

Tuzun, D., Burke, L.I., 1999. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research* 116, 1, 87–99.

Ukkusuri, S.V., Yushimito, W.F., 2008. Location routing approach for the humanitarian prepositioning problem. *Transportation Research Record* 2089, 1, 18–25.

Wu, T.H., Low, C., Bai, J.W., 2002. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research* 29, 10, 1393–1415.

Wu, L.Y., Zhang, X.S., Zhang, J.L., 2006. Capacitated facility location problem with general setup cost. *Computers & Operations Research* 33, 5, 1226–1241.

Yu, V.F., Lin, S.W., Lee, W., Ting, C.J., 2010. A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering* 58, 2, 288–299.

Zhou, L., Lin, Y., Wang, X., Zhou, F., 2019. Model and algorithm for bilevel multisized terminal location-routing problem for the last mile delivery. *International Transactions in Operational Research* 26, 1, 131–156.

Appendix A: Alternative MIP models for the LRP with facility sizing decisions

Section 3 shows a model that yields optimal solutions relatively quickly for our small newly-created instances. Nevertheless, alternative formulations can be made for our addressed problem. This appendix shows a comparison between three MIP models. The first model is the one shown in Section 3 (called the 3-index model henceforth). The second model is a modification of the first one, in which available sizes are not considered as an independent set (2-index model). Instead, we consider multiple copies of each facility, and each copy has a different capacity. Finally, the third model is an adaptation of a set-partitioning model (Baldacci et al., 2011), where a set of alternative sizes is included (SP model). Tables A1 and A2 show the sets, parameters, and variables of the 2-index model and the SP model, respectively. The 3-index and 2-index models are very similar, however, the 2-index model (Equations (A1)–(A4)) requires a set of dummy depots. For example, if an instance has 3 potential depot locations and there are 5 alternative sizes for each open depot, the set J has 15 dummy depots:

$$Minimize \quad \sum_{j\in J}(f_j + o_j)y_j + \sum_{a\in A}\sum_{k\in K} c_a w_{ak} + \sum_{a\in\delta^+(J)}\sum_{k\in K} v w_{ak} \tag{A1}$$

Table A1
Sets, parameters, and variables of a 2-index model for the LRP with facility sizing decisions

Sets
V = Set of nodes
K = Set of vehicles
I = Set of customers, $I \subset V$
J = Set of dummy depots, $J \subset V$
P = Set of potential depot locations, $P \subset V$
J_p = Set of dummy depots in each location $p \in P$, $J_p \subset J$
A = Set of arcs, $A = V \times V = \{(m, n) : m \in V, n \in V \wedge m \neq n\}$
$\delta^+(S)$ = Set of arcs leaving S, $S \subset V$, $\delta^+(S) \subset A$
$\delta^-(S)$ = Set of arcs entering S, $S \subset V$, $\delta^-(S) \subset A$

Parameters
s_j = Available size of depot $j \in J$
d_i = Demand of customer $i \in I$
f_j = Fixed opening cost of depot $j \in J$
o_j = Variable opening cost of depot $j \in J$
c_a = Cost of traversing arc $a \in A$
v = Fixed cost for using a vehicle
q = Capacity of each vehicle
M = A very large number when compared to the magnitude of the rest of the parameters

Variables
y_j = Binary variable equal to 1 if depot $j \in J$ is open, 0 otherwise
x_{ij} = Binary variable equal to 1 if customer $i \in I$ is assigned to depot $j \in J$, 0 otherwise
w_{ak} = Binary variable equal to 1 if arc $a \in A$ is used in the route performed by vehicle $k \in K$, 0 otherwise
u_{ik} = Accumulated deliveries by vehicle $k \in K$ until customer $i \in I$

Table A2
Sets, parameters, and variables of a set-partitioning model for the LRP with facility sizing decisions

Sets
V = Set of nodes
R = Set of feasible routes
L = Set of available sizes
I = Set of customers, $I \subset V$
J = Set of depots, $J \subset V$
R_j = Set of feasible routes passing through the depot $j \in J$, $R_j \subset R$
R_{ij} = Set of feasible routes of depot $j \in J$ passing through the customer $i \in I$, $R_{ij} \subset R$

Parameters
s_{jl} = Available size of type $l \in L$ for the depot $j \in J$
d_r = Demand of route $r \in R$
f_j = Fixed opening cost of depot $j \in J$
o_{jl} = Variable opening cost of depot $j \in J$ with size of type $l \in L$
c_{rj} = Cost of route $r \in R_j$ of depot $j \in J$
v = Fixed cost for using a vehicle

Variables
y_{jl} = Binary variable equal to 1 if depot $j \in J$ is open with size of type $l \in L$, 0 otherwise
x_{rj} = Binary variable equal to 1 if route $r \in R$ of depot $j \in J$ is included in the solution, 0 otherwise

s.t.

$$Constraints\ (2)–(8)$$

$$\sum_{i \in I} d_i x_{ij} \leq s_j y_j, \quad \forall j \in J \tag{A2}$$

$$\sum_{j \in J_p} y_j \leq 1, \quad \forall p \in P \tag{A3}$$

$$\forall\ y_j, x_{ij}, w_{ak} \in \{0, 1\} \tag{A4}$$

The objective function (A1) minimizes the total cost, formed by the depot fixed and variable opening costs, the routing costs, and the vehicles fixed costs. Constraints (A2) guarantee that the total demand of the customers assigned to an open depot does not exceed its capacity. Constraints (A3) ensure that at most one depot is open in each location. Finally, Constraints (A4) indicate the variables that are binary.

The SP model (Equations (A5)–(A9)) requires as an input a set of all feasible routes in the problem, i.e., these routes must be constructed before each instance is run in the optimization software. Additionally, each route has both a cost (or distance) and a demand, formed by the addition of all customers' demands in that route. These routes must be feasible, i.e., the vehicle capacity is used to construct them. After this procedure finishes, the vehicle capacity is not used further:

$$Minimize\ \sum_{j \in J} \sum_{l \in L} \left(f_j + o_{jl}\right) y_{jl} + \sum_{j \in J} \sum_{r \in R_j} \left(c_{rj} + v\right) x_{rj} \tag{A5}$$

s.t.

$$\sum_{j \in J} \sum_{r \in R_{ij}} x_{rj} = 1, \quad \forall i \in I \tag{A6}$$

$$\sum_{r \in R_j} d_r x_{rj} \leq \sum_{l \in L} s_{jl} y_{jl}, \quad \forall j \in J \tag{A7}$$

$$\sum_{l \in L} y_{jl} \leq 1, \quad \forall j \in J \tag{A8}$$

$$\forall\ y_{jl}, x_{rj} \in \{0, 1\} \tag{A9}$$

The objective function (A5) minimizes the total cost, formed by the depot fixed and variable opening costs, the distance-based costs of the routes, and the vehicles fixed costs. Constraints (A6) guarantee that each customer is served by only one route. Constraints (A7) ensure that the total demand of the routes assigned to each open depot does not exceed its assigned capacity. Constraints (A8) guarantee that at most one depot is open in each potential location. Finally, Constraints (A9) indicate that all variables are binary. All experiments in this appendix were run in a PC with an Intel Core i7 processor with 16 GB RAM, and using Windows 10 as operating system.

Table A3
Comparison of our MIP models using newly created instances

Instance	Optimal solution	Single equations	Single variables	Discrete variables	IGT (s)	MGT (s)	ST (s)	Total time (s)
				3-index model				
tor08x2a	751.23	273	315	290	-	0.20	0.68	0.88
tor08x2b	747.05	273	315	290	-	0.13	1.41	1.54
tor08x2c	664.56	273	315	290	-	0.15	13.52	13.67
tor08x2d	606.30	273	315	290	-	0.12	2.84	2.96
tor08x2e	815.57	273	315	290	-	0.12	3.58	3.70
tor10x3a	878.93	432	526	495	-	0.15	83.45	83.60
tor10x3b	652.50	432	526	495	-	0.14	188.02	188.16
tor10x3c	948.99	432	526	495	-	0.18	1027.90	1028.08
tor10x3d	742.36	432	526	495	-	0.16	19.54	19.70
tor10x3e	788.30	432	526	495	-	0.14	31.11	31.25
Average		353	421	393	-	0.15	137.21	137.35
				2-index model				
tor08x2a	751.23	497	763	738	-	0.13	4.51	4.64
tor08x2b	747.05	497	763	738	-	0.16	21.05	21.21
tor08x2c	664.56	497	763	738	-	0.14	43.73	43.87
tor08x2d	606.30	497	763	738	-	0.14	21.90	22.04
tor08x2e	815.57	497	763	738	-	0.15	11.42	11.57
tor10x3a	878.93	840	1366	1335	-	0.14	373.01	373.15
tor10x3b	652.50	840	1366	1335	-	0.17	2663.10	2663.27
tor10x3c	948.99	840	1366	1335	-	0.18	4246.76	4246.94
tor10x3d	742.36	840	1366	1335	-	0.17	75.89	76.06
tor10x3e	788.30	840	1366	1335	-	0.17	355.92	356.09
Average		669	1065	1037	-	0.15	781.73	781.88
				SP model				
tor08x2a	751.23	13	2939	2938	0.11	0.22	0.13	0.46
tor08x2b	747.05	13	26097	26096	0.66	1.97	0.12	2.75
tor08x2c	664.56	13	5171	5170	0.16	0.23	0.09	0.48
tor08x2d	606.30	13	2219	2218	0.12	0.15	0.07	0.34
tor08x2e	815.57	13	4259	4258	0.16	0.24	0.09	0.49
tor10x3a	878.93	17	265843	265842	16.65	218.28	0.83	235.76
tor10x3b	652.50	17	441673	441672	26.80	790.59	1.35	818.74
tor10x3c	948.99	17	86941	86940	3.39	18.73	0.29	22.41
tor10x3d	742.36	17	152089	152088	4.91	45.37	0.48	50.76
tor10x3e	788.30	17	134626	134625	4.48	38.46	0.42	43.36
Average		15	112186	112185	5.74	111.42	0.39	117.56

Table A3 displays the obtained results for our newly-created instances introduced in Section 5.1. Regardless of the MIP model, the optimal solution has always been found. Nevertheless, both the total necessary time to find these solutions and the number of single variables and equations are noticeably different for the three models. The 2-index model shows a higher number of variables and equations than the 3-index model. Despite the 2-index has one index less than the 3-index model, in the former the number of elements of the set J is multiplied by 5, which affects the size of the entire model. Additionally, the SP model shows both a significantly smaller number of variables and a greater number of equations than the other two models. The total time is formed by 3 terms:

1. *Instance generation time (IGT):* it is the time required to generate a file readable by the optimization software (e.g., GAMS). Since the SP model requires a list of all feasible routes (sets R, R_j and R_{ij}) as an input, as well as the parameters d_r and c_{rj}, the number of single input parameters can be really large. Hence, an application in Python was programmed to generate this instance file. Conversely, the 3-index and 2-index models do not require an automatic instance generation procedure, since the number of single input parameters is significantly smaller than the inputs for the SP model.
2. *Model generation time (MGT):* it is the time employed by GAMS to read and check the syntax of the input code, as well as the time spent to generate the model before it can be solved.
3. *Solving time (ST):* it is the time employed by GAMS to find the optimal solution after the model has been generated.

The 2-index model's average total time is about 6 times longer than the 3-index model's time, which shows how inefficient the 2-index model is. Additionally, the SP model's average total time is slightly smaller than the 3-index model's, despite the addition of the IGT. Given the large size of the input file for the SP model, the average MGT is significantly greater than the average ST. Contrarily, the average MGT is significantly smaller than the average ST for the 3-index and 2-index models. That is, the latter models are really easy to read and hard to solve, and the SP model shows an opposite performance.

Table A4 displays our obtained results for 9 small benchmark instances. Concretely, we use the Barreto's and Prodhon's instances whose number of customers is smaller than 30. The second column in these tables shows the Best found solution (BFS). An asterisk indicates that the BFS is the optimal solution. Since instances have been modified to include the alternative sizes, there is no reference in the literature where the optimal solution is provided. Hence, the 3-index and 2-index models found efficiently the optimal solution for the instance *Perl-12x2*, and the SP model found it for the instances *Coord20-5-1* and *Coord20-5-2*. The rest of the BFSs are obtained employing our metaheuristic approach (Tables 4 and 5). The solving time limit was set on 10,000 seconds. Three new indicators are added to Table A4: (i) the MIP solution, which is the best integer solution found by CPLEX when reaching the time limit, (ii) the optimality gap tolerance obtained by CPLEX when reaching the time limit, and (iii) the gap between the MIP solution and the BFS. The smaller these gaps, the better the results. Hence, the average gaps show again a higher efficiency of the 3-index model.

Table A5 shows the results obtained by the SP model for the same benchmark instances of Table A4. In this case, the problem size increases dramatically. In order to illustrate this statement, an upper bound (UB) for the number of feasible set partitions of customers – i.e., depots are not included, is calculated as follows:

1. Sort the customers' demands in ascending order.
2. Determine the maximum number of customers (N) that can be served in a single route. To attain this, add iteratively the demands of the first h elements until the vehicle capacity is reached, such that $\sum_{h=1}^{N} d_h \leq q$ and $\sum_{h=1}^{N+1} d_h > q$.

Table A4
Comparison of the 3-index and 2-index models using small benchmark instances

Instance	BFS	Single equations	Single variables	Discrete variables	MIP solution	CPLEX gap	BFS gap	MGT (s)	ST (s)	Total time (s)
					3-index model					
Perl-12x2	203.98*	545	611	574	203.98	0.00%	0.00%	0.17	21.81	21.98
Coord20-5-1	51165.49*	2586	3126	3025	58661.08	43.48%	14.65%	0.33	10000.00	10000.33
Coord20-5-2	43426.36*	2586	3126	3025	45322.41	32.25%	4.37%	0.32	10000.00	10000.32
Coord20-5-1b	34998.10	1572	1926	1865	34499.32	23.58%	−1.43%	0.31	10000.00	10000.31
Coord20-5-2b	33403.25	1572	1926	1865	33912.96	20.39%	1.53%	0.34	10000.00	10000.34
Gaskell-21x5	418.65	2265	2735	2650	458.26	24.35%	9.46%	0.39	10000.00	10000.39
Gaskell-22x5	578.86	1858	2248	2181	580.45	4.98%	0.27%	0.58	10000.00	10000.58
Min-27x5	2960.02	3549	4157	4048	3475.48	37.59%	17.41%	0.43	10000.00	10000.43
Gaskell-29x5	493.35	4041	4695	4578	501.83	26.88%	1.72%	0.35	10000.00	10000.35
Average	18627.52	2286	2728	2646	19735.09	23.72%	5.33%	0.36		
					2-index model					
Perl-12x2	203.98*	865	1283	1246	203.98	0.00%	0.00%	0.15	38.69	38.84
Coord20-5-1	51165.49*	4706	7526	7425	65550.80	50.27%	28.12%	0.36	10000.00	10000.36
Coord20-5-2	43426.36*	4706	7526	7425	53105.74	42.08%	22.29%	0.44	10000.00	10000.44
Coord20-5-1b	34998.10	2852	4726	4665	37535.61	31.52%	7.25%	0.40	10000.00	10000.40
Coord20-5-2b	33403.25	2852	4726	4665	33789.50	20.68%	1.16%	0.50	10000.00	10000.50
Gaskell-21x5	418.65	4045	6515	6430	483.71	29.15%	15.54%	0.37	10000.00	10000.37
Gaskell-22x5	578.86	3258	5328	5261	593.65	11.58%	2.56%	0.36	10000.00	10000.36
Min-27x5	2960.02	5809	9017	8908	4185.74	49.40%	41.41%	0.41	10000.00	10000.41
Gaskell-29x5	493.35	6461	9915	9798	804.77	55.48%	63.12%	0.67	10000.00	10000.67
Average	18627.52	3950	6285	6203	21805.94	32.24%	20.16%	0.41		

Table A5
Results of the SP model using small benchmark instances

Instance	Optimal solution	UB Number of feasible set partitions	Single equations	Single variables	Discrete variables	IGT (s)	MGT (s)	ST (s)	Total time (s)
Perl-12x2	203.98	2.38 $\times 10^{6}$	17	4757295	4757294	4219.16	168769.50	19.19	173007.86
Coord20-5-1	51165.49	9.92 $\times 10^{5}$	31	495551	495550	31.67	816.26	11.55	859.48
Coord20-5-2	43426.36	9.92 $\times 10^{5}$	31	779411	779410	44.70	1867.81	5.25	1917.76
Coord20-5-1b	–	3.72 $\times 10^{12}$							
Coord20-5-2b	–	3.72 $\times 10^{12}$							
Gaskell-21x5	–	6.98 $\times 10^{11}$			Out of memory to generate the instance file				
Gaskell-22x5	–	1.23 $\times 10^{20}$							
Min-27x5	–	4.44 $\times 10^{15}$							
Gaskell-29x5	–	1.00 $\times 10^{27}$							
Average		1.11 $\times 10^{26}$	26	2010752	2010751	1431.84	57151.19	12.00	58595.03

3. Calculate the number p of h-permutations of $|I|$, where $|I|$ is the instance's total number of customers, according to Equation (A10). Notice that this expression still does not include the set partitions with one single customer.

$$p = \sum_{h=2}^{N} P(|I|, h) \tag{A10}$$

4. Calculate UB according to Equation (A11). Two terms are included: firstly, the set partitions with one single customer, and secondly, the division of p by 2, which is useful to decrease UB. Since all arcs in the network are assumed to be symmetric, a route traversed in one direction has the same distance-based cost and demand than the same route traversed in the opposite direction:

$$UB = |I| + \frac{p}{2} \tag{A11}$$

Calculated UBs show the reason why a computer with the aforementioned characteristics is not even able to generate the instance file. For example, the IGT for the instance *Perl-12x2* is greater than one hour, with a UB equal to 2.38 x 10^6. In turn, the instance *Gaskell-21x5* has a UB about 300,000 times greater than the *Perl-12x2*'s, which shows the large size of that instance, as well as the size of the rest of instances whose optimal solution is not known. The MGT in Table A5 also shows how large the instance files are. For instance, GAMS took more than 46 hours only to generate the *Perl-12x2* model. Conversely, solving times are quite short in comparison.

Multiple conclusions can be drawn from the study shown in this appendix. Firstly, the 3-index model shows a better performance than the 2-index model under all considered indicators. Secondly, when considering our newly-created small instances, the SP model is 14% more time-efficient than the 3-index model in finding the optimal solution. However, the SP model efficiency is lost when increasing slightly the instance size, given the sharp rise in the size of the feasible routes set. Finally, after a solving time of 10,000 seconds, the 3-index model did not reach the best found solution for most benchmark instances. Our metaheuristic approach obtained these BFSs in less than 15 seconds (Tables 4 and 5), which shows its high time- and cost-efficiency.