INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 28 (2021) 201–221
DOI: 10.1111/itor.12796

Agile optimization of a two-echelon vehicle routing problem with pickup and delivery

Leandro do C. Martins[a,*], Patrick Hirsch[b] and Angel A. Juan[a,c]

[a] *IN3 – Computer Science Department, Universitat Oberta de Catalunya, Barcelona, Spain*
[b] *Institute of Production and Logistics, University of Natural Resources and Life Sciences, Vienna, Austria*
[c] *Euncet Business School, Terrassa, Spain*
E-mail: leandrocm@uoc.edu [do C. Martins]; patrick.hirsch@boku.ac.at [Hirsch]; ajuanp@uoc.edu [Juan]

Received 28 May 2019; received in revised form 8 January 2020; accepted 10 March 2020

Abstract

In this paper, we consider a vehicle routing problem in which a fleet of homogeneous vehicles, initially located at a depot, has to satisfy customers' demands in a two-echelon network: first, the vehicles have to visit intermediate nodes (e.g., a retail center or a consolidation center), where they deliver raw materials or bulk products and collect a number of processed items requested by the customers in their route; then, the vehicles proceed to complete their assigned routes, thus delivering the processed items to the final customers before returning to the depot. During this stage, vehicles might visit other intermediate nodes for reloading new items. In some real-life scenarios, this problem needs to be solved in just a few seconds or even milliseconds, which leads to the concept of "agile optimization." This might be the case in some rescue operations using drones in humanitarian logistics, where every second can be decisive to save lives. In order to deal with this real-time two-echelon vehicle routing problem with pickup and delivery, an original constructive heuristic is proposed. This heuristic is able to provide a feasible and reasonably good solution in just a few milliseconds. The constructive heuristic is extended into a biased-randomized algorithm using a skewed probability distribution to modify its greedy behavior. This way, parallel runs of the algorithm are able to generate even better results without violating the real-time constraint. Results show that the proposed methodology generates competitive results in milliseconds, being able to outperform other heuristics from the literature.

Keywords: agile optimization; disaster management; two-echelon vehicle routing problem; biased-randomized algorithms

1. Introduction

Real-time optimization, where decisions need to be made in just a few seconds or even milliseconds, has many application areas in logistics. For example, in the event of disasters, real-time optimization

*Corresponding author.

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

can be a life-saving differential. In this context, last-mile distribution logistics is related to the delivery of urgently needed goods to areas where roads are blocked by extreme weather events, disasters, or traffic congestion.

A motivational example is the distribution of drugs with drones in disaster situations. When disasters occur, an effective system of drug management should be established by health agencies in order to (a) ensure the efficient, cost-effective, and rational use of the drugs; (b) prevent and reduce excess mortality and morbidity; and (c) promote a return to normalcy (McConnan, 2004). This distribution process is based on selection, procurement, distribution, and use of pharmaceuticals in primary health care (Quick et al., 1997). We focus on the distribution stage in which urgently needed items (e.g., drugs) must be delivered to an affected area after a disaster occurred. These items should be delivered from a near pharmacy or laboratory (intermediate node) as fast as possible. On the other hand, these intermediate facilities should be served from a central warehouse or depot, which holds raw material used to fabricate the processed items requested by the final users.

Due to the resulting poor transportation infrastructure after a disaster event, the affected areas might become no longer accessible by conventional cargo vehicles. Therefore, the use of drones can be seen as an effective way of delivering life-saving treatments directly to disaster locations. Examples of drone delivery applications can be found in health care delivery, which includes the safe delivery of medicines, vaccines, defibrillators, blood samples, disease test samples, and test kits in remote areas out of reach (Balasingam, 2017; Scott and Scott, 2017).

In city logistics, a variant of the classical and well-known vehicle routing problem (VRP) is the two-echelon VRP (2E-VRP), which can be found in several transportation systems. This multilevel distribution system combines two delivery levels, in which the first level addresses the delivery from the depot to intermediate facilities, while the second level regards the delivery from these intermediate facilities to final customers. In the presented problem, a set of intermediate facilities (e.g., pharmaceutical laboratories, PLs) holds a limited inventory of drugs needed in the disaster areas and must be served with raw material from a single distribution center or depot. On the other hand, these intermediate facilities must serve as fast as possible a set of final delivery points in the affected area. The same fleet of drones is employed in both delivery levels. Although the motivational example for this paper is the drug distribution in disaster circumstances, similar problems can be found in other situations, too.

To solve this problem in "real time" (i.e., a few seconds or even milliseconds), we propose a fast constructive heuristic. This heuristic is then extended to a biased-randomized (BR) algorithm as described, for example, in Grasas et al. (2017). Hence, we introduce the concept of "agile optimization," which refers to the massive parallelization of a BR version of a constructive heuristics. Using parallel computing to solve real-life VRPs, the resulting methodology is able to provide in milliseconds "good" solutions to medium- and large-sized instances (Juan et al., 2013b). The use of BR techniques facilitates the design of powerful algorithms that can effectively be used to provide real-time solutions in a range of situations that arise in dynamic and emergency contexts (Ghiani et al., 2003).

The paper is arranged as follows. Section 2 presents a literature review on related topics; Section 3 describes the addressed problem; Section 4 introduces the proposed solution method; Section 5 presents an analysis of the results and a comparison between the proposed heuristic and other solving methods; finally, Section 6 highlights the main conclusions of this work and proposes some lines for future research.

2. Literature review

One of the first studies concerning two-level routing problems was presented by Jacobsen and Madsen (1980) in order to solve the daily distribution of newspapers. However, only decades later, Crainic et al. (2004) introduced the 2E-VRP, motivated using intermediate facilities to redistribute goods where large trucks were not able to circulate due to physical limitations of the streets. Consequently, the use of these intermediate facilities reduced the use of large vehicles by up to 72%. Years later, a study on the relationships between customer distribution, system layout, and the associated costs of the distribution process for two-echelon distribution systems was provided by Crainic et al. (2010). They measured and analyzed the impact of the number of customers, the quantity and location of intermediate facilities, the customer distribution, and the relationship between the first- and second-level costs on the total cost of distribution. The authors concluded that opening facilities reduce the global cost until a minimum cost is reached, and from that minimum, adding new ones increases the global cost. Different approximation methods have been proposed to solve the 2E-VRP. Hemmelmayr et al. (2012) proposed an adaptive large neighborhood search heuristic for solving the 2E-VRP. The authors developed new search operators based on the problem structure and were able to outperform existing results from the literature. Additionally, a new data set of large instances was proposed for the problem. More recent studies solve the 2E-VRP using hybrid methodologies. Crainic et al. (2013) presented a combination of a greedy randomized adaptive search procedure (GRASP) with path-relinking for solving the 2E-VRP while a combination of GRASP with a variable neighborhood descent (VND) has been presented by Zeng et al. (2014). Both hybrid methodologies were able to improve existing results in the literature. Recently, the 2E-VRP has been studied by introducing electric vehicles for the second-echelon deliveries (Breunig et al., 2019). In this case, a large neighborhood search metaheuristic was proposed to solve large-scale instances.

Another related problem is the VRP with pickup and delivery (VRPPD) and, in particular, the VRP with simultaneous pickup and delivery (VRPSPD). The VRPSPD was first tackled by Min (1989) for solving a real-life problem. The authors proposed a three-stage heuristic in order to minimize the total travel time of the routes. First, the customers were clustered complying the vehicle capacity per group. In the next step, one vehicle was assigned to each cluster, and finally, a traveling salesman problem (TSP) was solved for each group. From this work, several heuristics and metaheuristics have been proposed to solve the VRPSPD, which includes mainly the use of hybrid methodologies such as tabu search with VND (Crispim and Brandão, 2005; Bianchessi and Righini, 2007), ant colony optimization (ACO) with local search (Gajpal and Abad, 2009; Çatay, 2010), and particle swarm optimization combined with local search and VND (Ai and Kachitvichyanukul, 2009; Goksal et al., 2013), respectively. Finally, a parallel methodology based on simulated annealing (SA) has been considered to solve the problem (Mu et al., 2016).

An integration of 2E-VRP with simultaneous pickup and delivery was recently addressed by Belgin et al. (2018) who proposed a hybrid heuristic based on VND and local search to solve medium- and large-sized instances of the problem. In this work, the same fleet of homogeneous vehicles is employed to serve both delivery levels. However, a different vehicle capacity is imposed for each service level. Although the previous paper addresses a similar problem to the one studied here, our version is characterized by different constraints and decisions regarding the routing sequence, including the use of the same vehicle capacity for both service levels. To the best of our knowledge, the VRPPD in an omnichannel retailing context has been first introduced by Abdulkader et al.

(2018) who proposed a two-phase heuristic and a multiant colony (MAC) algorithm to solve the problem. The first phase of their heuristic is based on inserting customers into an initial route consisting only of retailers and on correcting infeasible solutions. The second phase is based on the well-known savings heuristic (Clarke and Wright, 1964). A complete set of instances has been generated to test both approaches.

The use of parallel computing is a breakthrough that allowed the resolution of larger and complex optimization problems (Migdalas et al., 2013). Its combination with optimization has made it possible to design powerful algorithms that can effectively be used to provide real-time solutions in dynamic contexts (Ghiani et al., 2003). Several parallel and distributed computing approaches have been already applied to different VRP variants. One of the studies regarding the use of parallel and distributed computing for solving VRPs in real-time has been written by Juan et al. (2013b). The authors pointed out potential applications of distributed computing to solve large-sized VRPs with real-life constraints. They proposed a solution approach, which combines parallel computing, simulation, and a BR heuristic for solving the VRP with stochastic demands. Recently, Rey et al. (2018) proposed a hybrid methodology based on ACO and local search procedures. Different levels of parallelism were tested, from the construction of TSP routes to the construction of a complete VRP solution from each TSP route. The parallel computing power was employed to generate high-quality solutions for the VRP.

Inspired by the works of Faulin and Juan (2008) and Faulin et al. (2008), which combined random sampling with heuristics for solving VRPs, Juan et al. (2010) were the first to use skewed probability distributions to bias the savings heuristic for solving VRPs. In their work, a geometric probability distribution was employed during the constructive stage of the savings heuristic to assign a selection probability to each candidate edge. Their methodology was improved in Juan et al. (2011) who incorporated cache and splitting techniques to reduce computational times. Since then, the use of BR techniques has been employed in solving different combinatorial optimization problems in areas such as transportation (e.g., Calvet et al., 2016; Dominguez et al., 2016), scheduling (e.g., Martin et al., 2016), or facility location (e.g., De Armas et al., 2017). Another class of BR algorithms, known as biased random-key genetic algorithms (BRKGA), was introduced by Gonçalves and Resende (2011) for solving combinatorial optimization problems. The idea behind these algorithms is to bias the selection of parents for mating. The use of BRKGA has been recently addressed for solving single-round scheduling problems (Brandão et al., 2015) and multiround scheduling problems (Brandão et al., 2017). In both cases, the proposed biased methodology was able to improve previously published best-known solutions.

Drones were initially used in military applications. However, their use in transportation has become a challenging trend in supply chains, in which companies from different industries have invested for delivery of goods, including food and medical products (Bamburry, 2015). When designing a system for delivering drugs using drones, several aspects from the application and physical limitations should be taken into account. From a physical point of view, Gatteschi et al. (2015) provided a detailed overview of these aspects applied to drug deliveries. Limitations such as battery, velocity, and weight were pointed out by Wan et al. (2018) who proposed a new mechanism designed to safely transport medical aids to the target area. From the application context, current and potential applications in health care are discussed by Balasingam (2017) who describes regulatory limitations and future innovations in drone technology, such as diagnostic capabilities using telemedicine to patients in hard-to-reach areas.

Although the use of drones for commercial purposes represents a significant advance in logistics, many technological and regulatory obstacles must be overcome. Several worst-case results regarding the use of drones in VRPs have been proposed by Wang et al. (2017). These worst cases reveal the benefits (amount of time that could be saved) of using drones combined with trucks instead of using only a fleet of trucks. Later, the same authors extended their previous work by explicitly considering limited battery life and cost objectives (Poikonen et al., 2017). Daknama and Kraus (2017) also studied the use of trucks and drones to deliver packages. In this case, the authors limited the number of packages that can be transported by drones at a time and imposed the return to a truck in order to charge their battery after each delivery. Both studies concluded that combining drones with trucks allows the truck to parallelize tasks, which represents a substantial improvement that must be considered in delivery systems. Dorling et al. (2017) proposed two multitrip VRPs for drone delivery, which minimize costs subject to a delivery time limit and minimize the overall delivery time subject to a budget constraint, respectively. A cost function that considers the energy consumption model and drone reuse was proposed and incorporated in an SA heuristic for finding near-optimal solutions to practical scenarios. According to the results, the reuse of drones and the optimization of battery size results in a substantial improvement in drone delivery VRPs.

3. Problem description

The problem addressed in this paper is based on the description presented by Abdulkader et al. (2018). In their work, a central depot holds a set of products delivered from different suppliers. This depot must serve a set of intermediate facilities (retail centers) using a fleet of homogeneous delivery vehicles. Each facility has a specific demand from the depot and holds a specific and limited inventory of each available item. These items are ordered by customers and should be delivered to them by the same fleet of vehicles. The items at the facilities are considered to be different from the products held at the depot for reasons of additional handling and/or packaging before being shipped to customers. Therefore, the products held at the depot cannot be shipped directly to the final delivery points.

In our application context, the depot holds raw materials, such as chemical and natural products, required by PLs in order to manufacture drugs to be delivered to final points. These final points can be seen as first-aid locations (FAL) in an area affected by a disaster. Each FAL is assigned to emergency staff (doctors, nurses, etc.) that requires these processed items to attend the population in an emergency situation. The delivery in both PLs and FALs is done by drones in order to allow the transportation of goods to nonaccessible areas, where conventional cargo vehicles cannot arrive by land. Sensors at PLs and FALs report in case of a disaster if a location is affected or not. They also give information about the impact of the disaster. This provides the depot with a situation picture of the affected areas. Needs for drugs are known in advance on the basis of an existing emergency plan in case of disasters. Authorities must quickly react to this new information in order to guarantee a fast delivery service. The affected area is a subset of the complete space, which might include some PLs. In this case, these facilities are treated as FALs, which require their specific medical supplies from an available nonaffected intermediate facility. Figure 1 provides an example of a scenario in which sensors in the affected area communicate with the depot when a disaster occurs. This area is composed of FALs and PLs. Each location is associated with a sensor. In this example, all affected

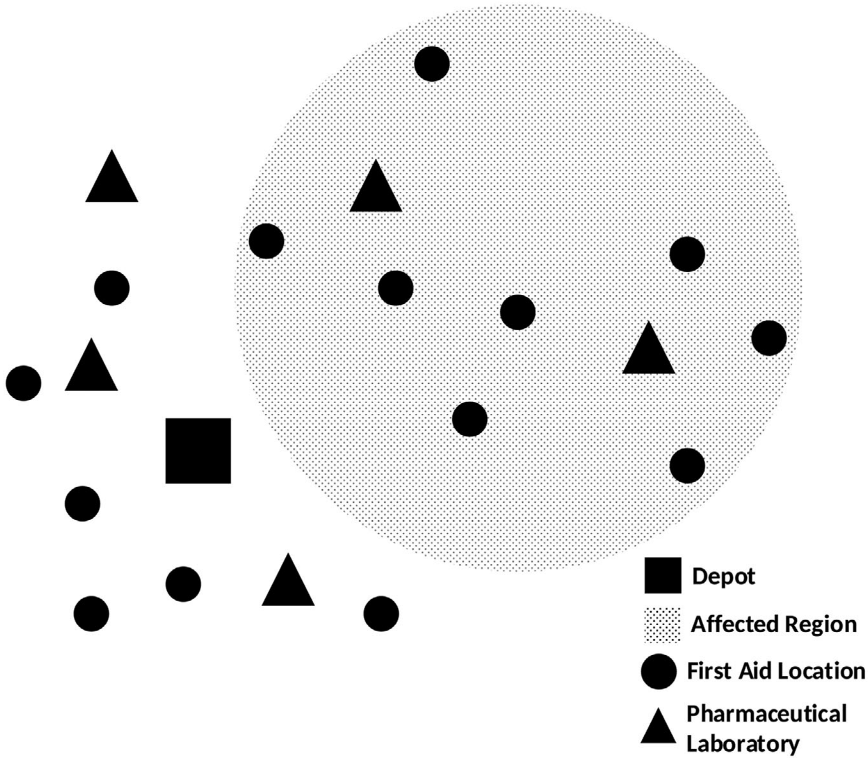

Fig. 1. An example of a situation picture provided by the sensors.

nodes (including FALs and PLs) must be served from the three available PLs that, on the other hand, must be served from the depot.

The drones are available and initially located at the depot. The capacity of drones is utilized to perform the required delivery of raw material from the depot to intermediate points and then to final points, in order to minimize the total transport time and guarantee the replenishment of medical supplies.

The solution to this problem is given by a set of drone routes. Each drone starts from the depot, visits a set of PLs and FALs, and returns empty to the depot. The goal is to minimize the total duration of the delivery routes such that

- each route must start and end at the depot;
- the routes do not exceed the maximum tour length;
- each PL or FAL is visited by only one drone and only once;
- the total delivery demand from the depot to the PLs does not exceed the drone capacity;
- the total demand of an FAL to be served from the PLs with a certain drug cannot exceed the available inventory of this drug at the PLs;
- the PLs designated to satisfy the demand of an FAL must be visited before the point and by the same delivery drone;
- decisions on distribution plans need to be made in real time.

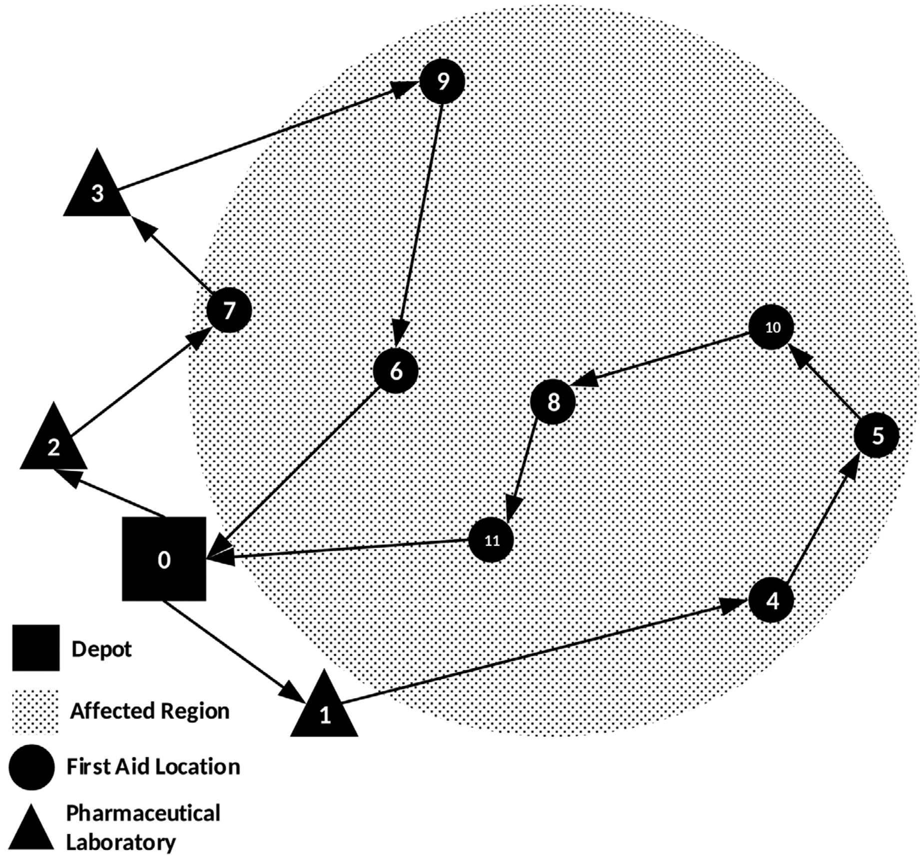

Fig. 2. An example of a two-echelon distribution system.

The distribution network of this problem can be defined as a directed graph $G = (V, A)$. The set V is composed of the central depot (node 0), the set of PLs and the set of FALs. The PLs and FALs are served by the same fleet of homogeneous drones with a certain capacity, which is available at the depot at the start time t_0. The complete mathematical formulation of this problem can be found in Abdulkader et al. (2018). Figure 2 provides a route example for the disaster scenario presented in Fig. 1. In this example, PL 2 serves FAL 7, while PL 3 serves FALs 9 and 6. It might be the case that FAL 9 or 6 is served by PL 2. In this case, the same visit order would be preserved as the pickup would be performed prior to delivery. In the second route, PL 1 serves FALs 4, 5, 10, 8, and 11. The routes satisfy constraints on vehicle capacity, maximum tour length, and inventory availability.

4. Methodology

In order to solve the proposed two-echelon VRP in real-time, a fast heuristic is proposed and then extended into a BR algorithm, which is executed using parallel computing techniques. The hybridization of BR heuristics and parallel computing leads us to the concept of agile optimization.

4.1. Agile optimization

Agile optimization has arisen as a new optimization concept for real-time decision making. It refers to the massive parallelization of BR algorithms, which are extremely fast in execution, easily parallelizable, flexible, and require the fine tuning of few, or even just a single parameter. The idea behind this technique is to run in parallel several hundreds or even thousands of threads, each thread being an execution of a BR heuristic. As a result, several alternative solutions are generated in the same wall-clock time as the one employed by the original heuristic, that is, milliseconds in most cases. Then, different solutions are provided—some of them outperforming the one generated by the original heuristic—and the best solution is chosen. Using skewed probability distributions, BR techniques employ the idea of introducing a biased (nonsymmetric) randomization effect into a heuristic procedure. As a result, a deterministic heuristic—which is extremely fast in execution, even for large-scale optimization problems—is extended into a probabilistic algorithm without losing the logic behind the original heuristic.

Agile optimization represents a new optimization perspective that allows to find reasonably good solutions for large-scale and *NP-hard* optimization problems in real time. This concept is also necessary when dealing with dynamic systems (e.g., traffic, vehicles location, unexpected demands, disruptions, etc.), where the environmental conditions are continuously changing and reoptimization of the system is required every few minutes or even seconds.

4.2. Heuristic

The proposed heuristic, henceforth named LH, is based on the aforementioned savings heuristic and is composed of four stages as described in Pseudocode 1. The first stage (line 1) consists of creating a dummy solution, which is composed of a set of "dummy" routes. Each dummy route is designed to serve one node $i \in V\backslash\{0\}$, which can be either an FAL or a PL. The route departs from the depot to the node and then returns to the depot.

In the second stage (lines 4–19), these initial routes visiting individual nodes are merged using a constructive heuristic. Initially, a list is constructed by considering all the possible pairs of nodes in the problem (line 2). The savings value of an edge $\{i, j\}$ is computed as $s_{ij} = d_{0i} + d_{j0} - d_{ij}$. In the first iteration, all edges from the list are eligible. This eligibility is related to constraints regarding the PLs' inventory. The list is initially sorted in descending order of the savings value (line 3) and the edge with the highest saving is selected. At this stage, we restrict the selection of edges in order to guarantee the assignment of a PL to each FAL in the problem. Hence, the selection is restricted to eligible edges $\{i, j\}$, where node j is an FAL in a dummy route and i is a node in a route with a PL that can supply FAL j (line 10). Figure 3 represents a merging case in which i is a PL that can supply node j. In Fig. 4, on the other hand, i is an FAL in a route with a PL that can supply node j. The two corresponding routes of an edge $\{i, j\}$ can be merged if (i) nodes i and j are exterior to their respective routes (a node is exterior to a route if it is adjacent to the depot); (ii) their respective routes are different (i and j belong to different routes); (iii) the maximum traveled distance constraint is not violated; and (iv) the vehicle capacity constraint is respected (line 11). This stage helps to assign each FAL to a PL supplier. Hence, we avoid infeasible solutions in which some FALs are not assigned to any PL. In our case, the selected edge is removed from the list L only if (a) the associated merge

Pseudocode 1. LH

```
   Data: set of nodes V
1  sol ← createDummySolution(V);
2  L ← createSavingsList(sol);
3  L ← sort(L);
4  while there are eligible edges in L do
5      e ← selectTheFirstEligibleEdgeFromList(L);
6      iNode ← getOrigin(e);
7      jNode ← getEnd(e);
8      iRoute ← getEvolvingRouteOfNode(iNode);
9      jRoute ← getEvolvingRouteOfNode(jNode);
10     if jNode is a non-served FAL and iRoute has a PL that can serve jNode then
11         if all route-merging conditions are satisfied then
12             sol ← mergeRoutesUsingEdge(e, iRoute, jRoute, sol);
13             edgesEligibility(L, true);
14         end
15         deleteEdgeFromList(e, L);
16     else
17         e ← eligibility(e, false);
18     end
19 end
20 while there are edges in L do
21     e ← selectTheFirstEdgeFromList(L);
22     iNode ← getOrigin(e);
23     jNode ← getEnd(e);
24     iRoute ← getEvolvingRouteOfNode(iNode);
25     jRoute ← getEvolvingRouteOfNode(jNode);
26     if all route-merging conditions are satisfied then
27         sol ← mergeRoutesUsingEdge(e, iRoute, jRoute, sol);
28     end
29     deleteEdgeFromList(e, L);
30 end
31 sol ← localSearch(sol);
32 return sol;
```

is completed or (b) at least one of the constraints (i)–(iv) is violated (line 15). Otherwise, the edge is not removed from the list (line 17) since it could be used in an ulterior iteration if new inventory is available after the incorporation of new distribution center to a route (line 12).

The third stage (lines 20–30) starts from the reduced list and attempts to merge all the available possibilities. At this stage, all edges are eligible, that is, there is no restriction in the selection regarding their types and characteristics (line 26), since all the FALs are already assigned to a PL. At this stage, each selected edge is removed from the list, whether it is used or not. This process is repeated until the list L is empty (line 20).

4.2.1. Local search

After the constructive stages, a fast local search (stage 4) is applied to improve the quality of the solution (line 31). It consists of an adapted *2-opt* movement plus a memory-based data

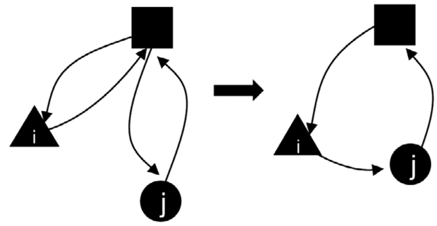

Fig. 3. Merging between a dummy PL route and a dummy FAL route, in which the PL i supplies the FAL j.

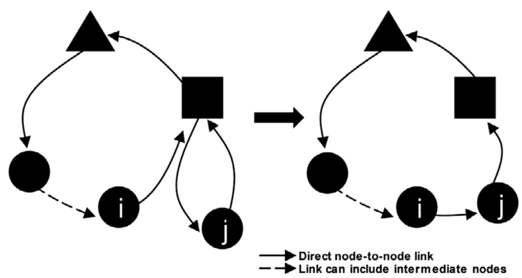

Fig. 4. Merging between a nondummy PL route and a dummy FAL route, in which the PL assigned to FAL i supplies the FAL j.

structure (hash map) that keeps the best-found route sequence for each set of nodes (Juan et al., 2011). The *2-opt* movement consists of swapping nodes inside each route. In our case, this movement is restricted to swaps that do not violate the precedence order between customers and their table suppliers.

4.3. Biased randomization

In order to extend the LH heuristic, its original greedy and deterministic behavior in the construction stage is modified by introducing a skewed probability distribution in the selection process (Pseudocode 2), which transforms the deterministic version of the algorithm into a probabilistic one without losing the logic behind the original idea (Juan et al., 2013a). In our case, the BR behavior is incorporated by employing a geometric distribution, which is controlled by a single parameter $\beta \in [0, 1]$.

Thus, in its extended version, BRLH, the selection of an edge from the sorted list is made accordingly to the probability provided by the parameter β (line 3). In other words, when using the geometric distribution, the edges at the top of the list are more likely to be chosen than those at the bottom. Therefore, the greedy behavior is smoothed and the methodology is able to generate different solutions, which allows the exploration of different solutions spaces when applying the local search mechanism. In this regard, lines 5 and 21 of Pseudocode 1 are replaced by the method *brSelection* (Pseudocode 2) in order to incorporate the BR strategy into the selection process. Note that, in stage 3 of Pseudocode 1, all the edges of L are eligible, then $l = |L|$ (line 2).

Pseudocode 2. brSelection

Data: savings list L, parameter $\beta \in [0, 1]$
$l \leftarrow$ getNumberOfEligibleEdgesFromList(L);
Randomly select position $x \in \{1, \dots, l\}$ according to distribution Geom(β);
$e \leftarrow$ selectTheXthEligibleEdgeFromList(x, L);
return e;

Finally, the resulting BRLH algorithm is embedded in a parallel framework to complete the agile optimization approach. Therefore, multiple runs of the same instance are executed in a concurrent/parallel way using different seeds for the pseudo-random number generator. This is possible due to the nonexistence of dependencies among the different runs. Hence, several solutions are generated, and the one with the lowest cost is chosen.

5. Computational experiments and results

The proposed methodology was tested on 80 instances proposed by Abdulkader et al. (2018). These instances are different in the number of retail stores and customers. In our application context, the retail centers, R, are equivalent to PLs and consumers, C, to the FALs. The first 20 instances are considered small-sized and were optimally solved with Cplex. The remaining 60 instances are large-sized instances that are different in the inventory scenarios of the retail centers (tight, relaxed, and abundant). According to Abdulkader et al. (2018), for each scenario, the total network inventory is computed as

- tight:

$$\sum_{i \in R} I_{ip} = \sum_{j \in C} D_{jp} + U[0.1, 0.2] \sum_{j \in C} D_{jp} \qquad \forall p \in P; \tag{1}$$

- relaxed:

$$\sum_{i \in R} I_{ip} = \sum_{j \in C} D_{jp} + U[0.5, 1.0] \sum_{j \in C} D_{jp} \qquad \forall p \in P; \tag{2}$$

- abundant: (at each retail store)

$$I_{ip} = \sum_{j \in C} D_{jp} \qquad \forall i \in R, \forall p \in P, \tag{3}$$

where P is the set of heterogeneous products, R is the set of retail centers, C is the set of customers, $\sum I_{ip}$ is the total inventory available of a product p, and $\sum D_{jp}$ is the total online demand of a product p.

The maximum tour length and the vehicle capacity are fixed to eight hours and 100 weight units, respectively. Regarding the BR process during the solution-construction stage, after some initial

Table 1
Comparison of results obtained by the different methodologies on solving small-sized instances

			1	2	3	4	5	Avg. Cost	%SD	Time			gap					
I	$\|R\|$	$\|C\|$	BRLH′	BRLH	BRLH″	AH	MAC	(2)	(2)	(1, 2)	(4)	(5)	(4-1)	(4-2)	(4-3)	(5-1)	(5-2)	(5-3)
a1	3	6	398.3	386.9	386.9	479.2	386.9[a]	386.9	0.0%	0.0	0.0	<1	−17%	−19%	−19%	3%	0%	0%
a2	3	9	430.2	416.3	416.3	589.6	416.4[a]	416.3	0.0%	0.0	0.0	<1	−27%	−29%	−29%	3%	0%	0%
a3	3	12	476.2	438.3	427.7	512.3	424.3[a]	447.9	2.2%	0.0	0.0	<1	−7%	−14%	−17%	12%	3%	1%
a4	3	15	571.7	500.6	487.7	767.1	455.3[a]	513.0	2.4%	0.0	0.0	<1	−25%	−35%	−36%	26%	10%	7%
a5	3	18	714.3	675.6	661.0	936.2	601.4[a]	690.4	1.3%	0.0	0.0	<1	−24%	−28%	−29%	19%	12%	10%
a6	4	6	454.3	425.7	425.0	512.1	419.3[a]	435.8	2.1%	0.0	0.0	<1	−11%	−17%	−17%	8%	2%	1%
a7	4	9	463.5	455.9	455.9	688.4	455.9[a]	462.0	0.7%	0.0	0.0	<1	−33%	−34%	−34%	2%	0%	0%
a8	4	12	483.0	479.7	466.3	711.9	449.4	484.1	1.0%	0.0	0.0	<1	−32%	−33%	−34%	7%	7%	4%
a9	4	15	530.5	477.6	461.4	746.6	457.3[a]	518.5	4.7%	0.0	0.0	<1	−29%	−36%	−38%	16%	4%	1%
a10	4	18	718.9	608.5	602.8	740.8	514.4[a]	676.6	5.3%	0.0	0.0	<1	−3%	−18%	−19%	40%	18%	17%
a11	5	6	512.1	492.0	488.5	545.7	486.0[a]	495.7	1.1%	0.0	0.0	<1	−6%	−10%	−10%	5%	1%	1%
a12	5	9	688.9	624.8	624.8	882.0	624.8[a]	637.3	3.3%	0.0	0.0	<1	−22%	−29%	−29%	10%	0%	0%
a13	5	12	626.7	579.5	539.0	961.7	535.4[a]	618.6	3.0%	0.0	0.0	<1	−35%	−40%	−44%	17%	8%	1%
a14	5	15	739.1	680.2	656.8	838.9	605.0[a]	710.3	2.7%	0.0	0.0	<1	−12%	−19%	−22%	22%	12%	9%
a15	5	18	850.6	777.9	747.4	898.8	709.9	804.6	2.4%	0.0	0.0	<1	−5%	−13%	−17%	20%	10%	5%
a16	6	6	505.5	502.6	484.5	582.1	468.9[a]	505.0	0.2%	0.0	0.0	<1	−13%	−14%	−17%	8%	7%	3%
a17	6	9	500.3	481.0	468.6	609.0	468.6[a]	491.0	2.0%	0.0	0.0	<1	−18%	−21%	−23%	7%	3%	0%
a18	6	12	715.2	659.3	638.8	924.4	586.6[a]	670.1	1.2%	0.0	0.0	<1	−23%	−29%	−31%	22%	12%	9%
a19	6	15	796.9	767.2	751.5	964.1	750.9[a]	781.3	1.6%	0.0	0.0	<1	−17%	−20%	−22%	6%	2%	0%
a20	6	18	778.8	717.6	674.6	1005.7	601.7	743.3	3.3%	0.0	0.1	<1	−23%	−29%	−33%	29%	19%	12%
Average									2.0%	0.0	0.0	–	−19%	−24%	−26%	14%	7%	4%

[a]Optimal solutions provided by Abdulkader et al. (2018).

tests, we randomly selected the parameter β in the interval [0.45, 0.75]. The number of parallel runs was fixed with 64 (each run using a different seed for the pseudo-random number generator). For each instance, a total of 10 repetitions were performed in order to collect statistical data. The performance of the heuristics was measured by means of the percentage gap between the best-found solution using that methodology, that is, the one with the lowest cost value, and the best-found solution obtained with the alternative solution methodology. Thus, the lower the gap is, the better the performance of the method is. The solution cost is given in travel time (in minutes). The entire algorithm was coded in Java and the tests were performed on an Intel Core i7-8550U processor with 16 GB of RAM.

Table 1 presents the results obtained by our BR heuristic (BRLH) on the set of small-sized instances. Our results are compared with those obtained by the two approaches proposed by Abdulkader et al. (2018), that is, the two-phase heuristic (AH) and the metaheuristic (MAC), besides the results obtained by the BRLH when only a single thread/run is available (BRLH′). A different BRLH version was similarly considered by allowing the method to generate solutions during a limited amount of time and then returning the best-found solution when this stop criterion is met (BRLH″). For each instance, the following information is provided: the solution cost obtained by the different methodologies (BRLH′, BRLH, BRLH″, AH, MAC), the average cost, and percentage standard deviation (%SD) of our results, the CPU time (in seconds) required by each methodology, and their gaps. While negative gaps between methods A and B (A–B) correspond to worse solution costs obtained by solving methodology A, positive gaps correspond to better solutions obtained by method A. Since BRLH′ differs from BRLH only in the number of parallel runs, their CPU time is aggregated in a single column. In the case of BRLH″, the maximum execution time was set to five seconds. According to Abdulkader et al. (2018), the CPU time required by both AH and MAC methods to generate the solutions was less than one second. Nevertheless, we have freely implemented the AH heuristic in order to collect the CPU time in our machine environment, which is less powerful than their one. When comparing our solution method with those proposed by Abdulkader et al. (2018), we are able to find results, on average, 24% better than the AH, column *gap (4-2)*, and 7% worse than the MAC, column *gap (5-2)*. In terms of CPU time, our results are quite competitive mainly when compared with the ones from the AH method, which require the same computational time but are significantly worse. Although our method requires basically the same CPU time as MAC for solving these small instances, this is not the case when larger sized instances are considered, in which MAC time is orders of magnitude larger than the one requested by our BRLH. Comparing with the optimal solutions provided by Abdulkader et al. (2018), our methodology is able to find four optimal solutions. By allowing five seconds during the execution of our methodology, the resulting BRLH″ is able to find six optimal solutions and five near-optimal solutions (solutions up to 1% worse than the optimal ones), column *gap (5-3)*, being only 4% worse than the MAC results, and 26% better than the AH results, column *gap (4-3)*. Analyzing the single thread version of our BRLH, its results are, on average, 14% worse than MAC, column *gap (5-1)*, but approximately 19% better than the alternative AH heuristic, column *gap (4-1)*. Regarding the variance of our results, the average standard deviation is 2%.

Another characteristic that directly influences the performance of our methodology is the number of threads, that is, its number of parallel runs. Therefore, in order to verify the behavior on different hardware settings that are characterized by a different number of threads, our BRLH was tested on settings consisting of 1, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096 threads. Figure 5 presents the convergence of the solutions when compared with the MAC solutions. By increasing the number of threads, our methodology is able to find up to nine optimal solutions when using more than 1024 threads, achieving a positive average gap of 3.6%.

Due to the satisfactory performance of our BR approach in small-sized instances, the BRLH was also tested in solving large-sized instances. Apart from the approximation methods (AH and MAC) proposed by Abdulkader et al. (2018), our results are compared with those obtained by the deterministic version of our heuristic (LH). Tables 2–4 present the obtained results on inventory scenarios of tight, relaxed and abundant, respectively.

Now, this first analysis aims to quantify the improvement when employing the use of agile optimization in the deterministic version of our proposed methodology. As we can see in column *gap (1-3)* of Tables 2, 3 and 4, which compares our both methodologies, with the use of agile optimization, the resulting heuristic is able to improve its deterministic version at between 8% and 11%, without increasing the required CPU time. This particularity of our methodology, which refers to parallel executions of BR algorithms, allow us to generate multiple alternative solutions in the same clock time. Comparing the results obtained by LH with those produced by the two-phase heuristic, column *gap (5-1)*, our results are between 12% and 16% better even without considering any stochasticity in the search-guidance process. When comparing the results generated

Table 2
Comparison of results obtained by the different methodologies in the scenario of tight inventory

			1	2	3	4	5	6	Avg. Cost	%SD	Time				gap						
I	$\|R\|$	$\|C\|$	LH	BRLH′	BRLH	BRLH″	AH	MAC	(3)	(3)	(1)	(2,3)	(5)	(6)	(1-3)	(5-1)	(5-2)	(5-3)	(6-2)	(6-3)	(6-4)
b1	10	25	1277.5	1199.8	1134.5	1095.1	131.6	1002.5	1153.0	1.7%	0.0	0.0	0.1	7.0	−11%	−22%	−26%	−30%	20%	13%	9%
b2	10	50	1641.1	1495.4	1392.5	1345.9	2057.5	1192.0	1448.1	1.5%	0.0	0.0	0.0	47.0	−15%	−20%	−27%	−32%	25%	17%	13%
b3	10	75	2663.7	2678.5	2463.4	2264.7	3006.2	1815.4	2554.8	1.9%	0.0	0.0	0.0	79.0	−8%	−11%	−11%	−18%	48%	36%	25%
b4	10	100	2415.1	2189.5	1968.7	1865.2	2830.2	1529.0	2062.3	2.5%	0.0	0.0	0.0	286.0	−18%	−15%	−23%	−30%	43%	29%	22%
b5	10	150	2678.2	2524.8	2395.0	2327.8	3478.7	1905.2	2450.4	1.3%	0.0	0.0	0.0	576.0	−11%	−23%	−27%	−31%	33%	26%	22%
b6	15	25	1540.5	1474.4	1408.1	1379.1	1774.4	1313.7	1428.9	1.0%	0.0	0.0	0.0	7.0	−9%	−13%	−17%	−21%	12%	7%	5%
b7	15	50	2059.0	1979.9	1788.8	1680.6	2461.8	1522.3	1898.4	3.0%	0.0	0.0	0.0	44.0	−13%	−16%	−20%	−27%	30%	18%	10%
b8	15	75	3105.3	2757.7	2649.4	2521.1	3545.1	2101.8	2678.4	0.7%	0.0	0.0	0.0	131.0	−15%	−12%	−22%	−25%	31%	26%	20%
b9	15	100	3121.5	2964.5	2826.3	2781.9	3529.0	2329.5	2905.7	1.5%	0.0	0.0	0.0	209.0	−9%	−12%	−16%	−20%	27%	21%	19%
b10	15	150	4292.4	3953.0	3749.6	3707.5	4916.8	3012.2	3858.2	1.3%	0.1	0.1	0.0	430.0	−13%	−13%	−20%	−24%	31%	24%	23%
b11	20	25	2035.2	1920.4	1889.5	1818.1	2432.6	1611.3	1914.0	0.7%	0.0	0.0	0.0	11.0	−7%	−16%	−21%	−22%	19%	17%	13%
b12	20	50	2335.4	2187.8	2126.0	2068.8	2695.3	1800.9	2160.0	0.8%	0.0	0.0	0.0	50.0	−9%	−13%	−19%	−21%	21%	18%	15%
b13	20	75	3212.7	2973.1	2784.6	2689.8	3936.7	2406.0	2913.4	1.9%	0.0	0.0	0.0	127.0	−13%	−18%	−24%	−29%	24%	16%	12%
b14	20	100	3025.2	3070.4	2869.4	2800.3	3826.1	2483.8	2933.1	1.1%	0.0	0.0	0.0	327.0	−5%	−21%	−20%	−25%	24%	16%	13%
b15	20	150	3934.3	3307.2	3241.8	3226.5	4496.1	2679.2	3345.4	2.1%	0.1	0.1	0.0	708.0	−18%	−12%	−26%	−28%	23%	21%	20%
b16	25	25	2019.4	1950.8	1897.7	1824.8	2254.9	1669.6	1925.4	1.3%	0.0	0.0	0.0	13.0	−6%	−10%	−13%	−16%	17%	14%	9%
b17	25	50	2665.9	2523.3	2430.3	2370.4	3020.8	1965.6	2494.4	1.5%	0.0	0.0	0.0	46.0	−9%	−12%	−16%	−20%	28%	24%	21%
b18	25	75	3207.6	3044.9	2946.4	2803.8	3963.5	2449.8	2974.5	0.7%	0.0	0.0	0.0	136.0	−8%	−19%	−23%	−26%	24%	20%	14%
b19	25	100	4064.0	3730.2	3502.3	3478.3	4933.9	2788.5	3622.3	1.9%	0.1	0.1	0.0	257.0	−14%	−18%	−24%	−29%	34%	26%	25%
b20	25	150	3782.7	3707.4	3544.4	3405.8	4721.3	2890.3	3593.0	1.1%	0.1	0.2	0.0	712.0	−6%	−20%	−21%	−25%	28%	23%	18%
Average										1.5%	0.0	0.0	0.0	210.2	−11%	−16%	−21%	−25%	27%	21%	16%

Table 3
Comparison of results obtained by the different methodologies in the scenario of relaxed inventory

			1	2	3	4	5	6	Avg. Cost	%SD	Time				gap						
I	$\|R\|$	$\|C\|$	LH	BRLH′	BRLH	BRLH″	AH	MAC	(3)	(3)	(1)	(2,3)	(5)	(6)	(1-3)	(5-1)	(5-2)	(5-3)	(6-2)	(6-3)	(6-4)
b21	10	25	1233.0	1220.6	1087.6	1007.1	1571.6	879.2	1110.3	1.5%	0.0	0.0	0.0	10.0	−12%	−22%	−22%	−31%	39%	24%	15%
b22	10	50	1490.8	1406.4	1355.9	1197.5	1920.6	1083.7	1374.4	1.0%	0.0	0.0	0.0	85.0	−9%	−22%	−27%	−29%	30%	25%	11%
b23	10	75	2468.0	2190.6	2121.2	2008.2	2699.2	1591.5	2156.9	1.0%	0.0	0.0	0.0	167.0	−14%	−9%	−19%	−21%	38%	33%	26%
b24	10	100	1885.0	1761.5	1645.9	1599.2	2305.1	1437.7	1706.0	2.3%	0.0	0.0	0.0	528.0	−13%	−18%	−24%	−29%	23%	14%	11%
b25	10	150	1998.6	2136.9	2003.1	1875.2	2700.4	1520.5	2030.3	0.8%	0.0	0.0	0.0	1836.0	0%	−26%	−21%	−26%	41%	32%	23%
b26	15	25	1591.4	1462.0	1268.0	1268.0	1665.2	1180.8	1353.0	3.5%	0.0	0.0	0.0	11.0	−20%	−4%	−12%	−24%	24%	7%	7%
b27	15	50	1940.7	1756.4	1674.9	1603.5	2320.7	1329.3	1692.0	0.8%	0.0	0.0	0.0	73.0	−14%	−16%	−24%	−28%	32%	26%	21%
b28	15	75	2436.3	2269.4	2204.4	2076.9	3016.5	1692.4	2255.4	1.3%	0.0	0.0	0.0	279.0	−10%	−19%	−25%	−27%	34%	30%	23%
b29	15	100	2648.3	2628.0	2430.1	2299.9	3302.4	2016.4	2486.5	1.7%	0.0	0.0	0.0	567.0	−8%	−20%	−20%	−26%	30%	21%	14%
b30	15	150	3373.2	2963.9	2763.6	2767.0	3919.0	2399.6	2890.3	2.6%	0.0	0.1	0.0	1407.0	−18%	−14%	−24%	−29%	24%	15%	15%
b31	20	25	1835.5	1774.4	1687.2	1645.5	1993.5	1495.8	1747.6	1.6%	0.0	0.0	0.0	16.0	−8%	−8%	−11%	−15%	19%	13%	10%
b32	20	50	2320.5	2140.7	2005.0	1867.8	2713.0	1656.9	2045.6	1.1%	0.0	0.0	0.0	76.0	−14%	−14%	−21%	−26%	29%	21%	13%
b33	20	75	2404.6	2360.4	2312.0	2232.3	3393.3	1799.6	2348.2	0.7%	0.0	0.0	0.0	262.0	−4%	−29%	−30%	−32%	31%	28%	24%
b34	20	100	2751.9	2677.7	2497.7	2423.8	3127.5	2018.5	2532.6	1.0%	0.0	0.0	0.0	740.0	−9%	−12%	−14%	−20%	33%	24%	20%
b35	20	150	3157.4	2832.5	2818.7	2739.6	3742.2	2291.0	2877.0	1.3%	0.1	0.1	0.0	2141.0	−11%	−16%	−24%	−25%	24%	23%	20%
b36	25	25	1844.0	1786.4	1744.7	1643.7	2032.1	1550.0	1765.6	0.8%	0.0	0.0	0.0	15.0	−5%	−9%	−12%	−14%	15%	13%	6%
b37	25	50	2663.9	2555.2	2409.1	2279.3	3130.5	1939.5	2455.3	1.1%	0.0	0.0	0.0	73.0	−10%	−15%	−18%	−23%	32%	24%	18%
b38	25	75	2790.7	2770.3	2547.4	2507.6	3433.2	2088.6	2608.6	1.1%	0.0	0.0	0.0	283.0	−9%	−19%	−19%	−26%	33%	22%	20%
b39	25	100	3352.9	3207.2	3076.8	2964.6	3824.5	2244.1	3113.1	1.0%	0.0	0.0	0.0	656.0	−8%	−12%	−16%	−20%	43%	37%	32%
b40	25	150	2971.1	2926.0	2811.1	2764.8	3447.9	2229.4	2848.8	0.9%	0.1	0.1	0.0	2077.0	−5%	−14%	−15%	−18%	31%	26%	24%
Average										1.3%	0.0	0.0	0.0	565.1	−10%	−16%	−20%	−24%	30%	23%	18%

Table 4
Comparison of results obtained by the different methodologies in the scenario of abundant inventory

			1	2	3	4	5	6	Avg. Cost	%SD	Time				gap						
I	$\|R\|$	$\|C\|$	LH	BRLH′	BRLH	BRLH″	AH	MAC	(3)	(3)	(1)	(2,3)	(5)	(6)	(1-3)	(5-1)	(5-2)	(5-3)	(6-2)	(6-3)	(6-4)
b41	10	25	805.4	793.4	760.8	743.9	897.6	711.3	775.9	1.5%	0.0	0.0	0.0	16.0	−6%	−10%	−12%	−15%	12%	7%	5%
b42	10	50	1014.2	910.4	872.9	870.0	1287.8	875.2	885.0	1.2%	0.0	0.0	0.0	143.0	−14%	−21%	−29%	−32%	4%	−0.4%	−1%
b43	10	75	1463.5	1332.3	1260.3	1225.6	1531.1	1132.1	1307.0	1.7%	0.0	0.0	0.0	358.0	−14%	−4%	−13%	−18%	18%	11%	8%
b44	10	100	1379.4	1382.4	1303.3	1280.8	1636.5	1224.1	1309.9	0.5%	0.0	0.0	0.0	978.0	−6%	−16%	−16%	−20%	13%	6%	5%
b45	10	150	1499.3	1509.9	1415.9	1353.9	1551.8	1273.9	1438.0	1.1%	0.0	0.0	0.0	2085.0	−6%	−3%	−3%	−9%	19%	11%	6%
b46	15	25	1137.5	1063.2	1039.6	1013.5	1264.3	996.9	1048.0	0.7%	0.0	0.0	0.0	22.0	−9%	−10%	−16%	−18%	7%	4%	2%
b47	15	50	1247.6	1225.7	1154.2	1118.8	1488.1	1080.3	1186.0	1.1%	0.0	0.0	0.0	159.0	−7%	−16%	−18%	−22%	13%	7%	4%
b48	15	75	1595.3	1483.4	1370.0	1349.6	1815.2	1252.4	1401.3	2.6%	0.0	0.0	0.0	559.0	−14%	−12%	−18%	−25%	18%	9%	8%
b49	15	100	2021.3	1979.9	1785.9	1748.2	2242.4	1594.0	1867.7	1.9%	0.0	0.0	0.0	1167.0	−12%	−10%	−12%	−20%	24%	12%	10%
b50	15	150	2059.6	1941.5	1869.6	1841.3	2459.5	1691.4	1901.2	0.7%	0.0	0.1	0.0	4126.0	−9%	−16%	−21%	−24%	15%	11%	9%
b51	20	25	1507.2	1476.0	1425.7	1350.3	1660.9	1302.9	1446.8	0.7%	0.0	0.0	0.0	33.0	−5%	−9%	−11%	−14%	13%	9%	4%
b52	20	50	1464.7	1392.5	1367.8	1357.2	1740.7	1301.0	1379.3	0.4%	0.0	0.0	0.0	156.0	−7%	−16%	−20%	−21%	7%	5%	4%
b53	20	75	1797.7	1757.5	1626.2	1559.9	2096.8	1421.8	1661.6	1.0%	0.0	0.0	0.0	605.0	−10%	−14%	−16%	−22%	24%	14%	10%
b54	20	100	2066.2	2084.0	1839.8	1801.0	2226.4	1640.6	1956.5	2.6%	0.0	0.0	0.0	1370.0	−11%	−7%	−6%	−17%	27%	12%	10%
b55	20	150	2214.0	2073.1	2044.3	1963.2	2518.2	1763.3	2072.3	0.8%	0.0	0.1	0.0	5321.0	−8%	−12%	−18%	−19%	18%	16%	11%
b56	25	25	1423.2	1399.9	1382.4	1356.0	1550.7	1311.6	1395.9	0.5%	0.0	0.0	0.0	36.0	−3%	−8%	−10%	−11%	7%	5%	3%
b57	25	50	1670.8	1631.2	1548.5	1532.8	1835.4	1468.1	1599.4	1.4%	0.0	0.0	0.0	203.0	−7%	−9%	−11%	−16%	11%	5%	4%
b58	25	75	2047.5	1922.6	1870.0	1776.1	2276.9	1654.9	1899.1	1.0%	0.0	0.0	0.0	791.0	−9%	−10%	−16%	−18%	16%	13%	7%
b59	25	100	1856.4	1860.7	1808.1	1743.5	2061.9	1575.7	1829.8	0.7%	0.0	0.0	0.0	1262.0	−3%	−10%	−10%	−12%	18%	15%	11%
b60	25	150	1968.1	1940.8	1856.1	1820.7	2347.8	1653.3	1878.0	0.8%	0.0	0.1	0.0	4549.0	−6%	−16%	−17%	−21%	17%	12%	10%
Average										1.2%	0.0	0.0	0.0	1,197.0	−8%	−12%	-15%	−19%	15%	9%	6%

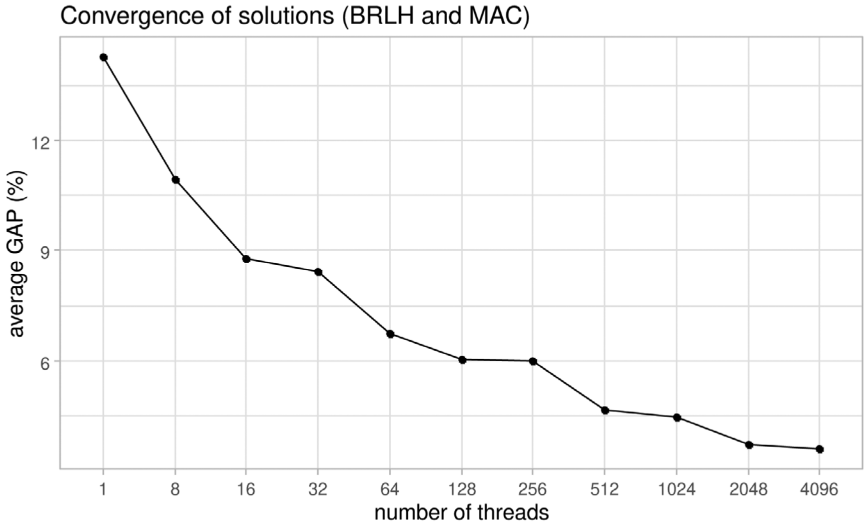

Fig. 5. Comparison between BRLH and MAC on solving small-sized instances with a different number of parallel runs.

by our BRLH with those generated by the AH methodology, column *gap (5-3)*, our approach can improve at least 19% of the solutions on abundant inventory scenario and between 24% and 25% on the remaining ones. Now, comparing the BRLH results with the best-known solutions, column *gap (6-3)*, our results are at most 18% worst on average (considering all the solutions). However, the average CPU time required by the MAC is hugely more extensive than ours, which does not represent a suitable time for our application context that requires solutions in extremely short computational times. Note that for some instances, MAC is able to generate noticeable better solutions in reduced computing times (less than a minute). However, this is only the case for small instances. In the abundant inventory scenario, our methodology is only 9% worse, being able to find a new best solution for instance *b42*. When comparing the BRLH′ results, its performance is, on average, 24% worse than MAC, column *gap (6-2)*, but approximately 19% better than the AH heuristic, column *gap (5-2)*. Regarding the BRLH″, an average improvement of about 5% is achieved, when comparing against MAC, column *gap (6-4)*, with respect the BRLH, column *gap (6-3)*. Regarding the variance of our results, the average percentage standard deviation varies from 1.2% to 1.5%. Additionally, all the average values are below the solution cost obtained by the AH heuristic. These last two analyses allow us to certify the robustness of our proposed methodology, which is capable of generating good solutions with a small variance of the cost of the solutions.

Figures 6 and 7 present the convergence of BRLH large-sized instances solutions when comparing with the methodologies of AH and MAC, respectively, for each inventory scenario. As we can see, the convergence behavior is similar for all inventory scenarios when comparing with both solution approaches (AH and MAC). By increasing the number of threads, our methodology is able to reach up to 27% of improvement when comparing with AH, being only 8% worse than MAC's results.

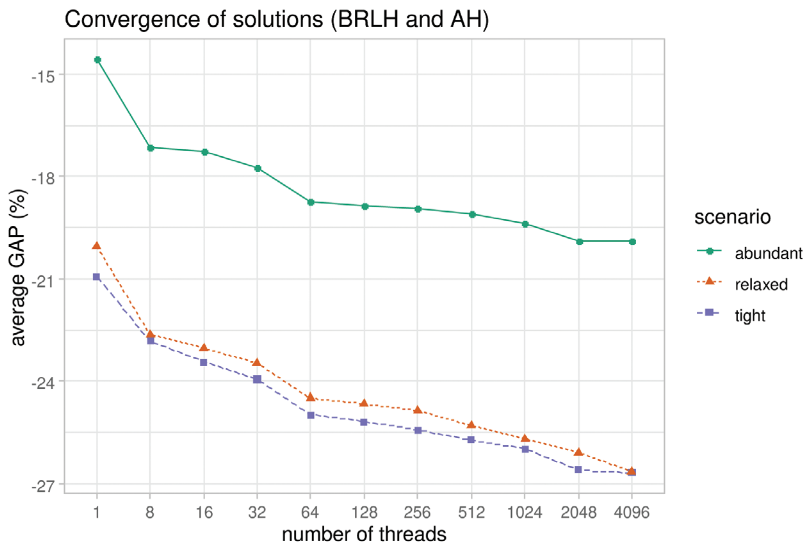

Fig. 6. Comparison between BRLH and AH on solving large-sized instances with a different number of parallel runs.

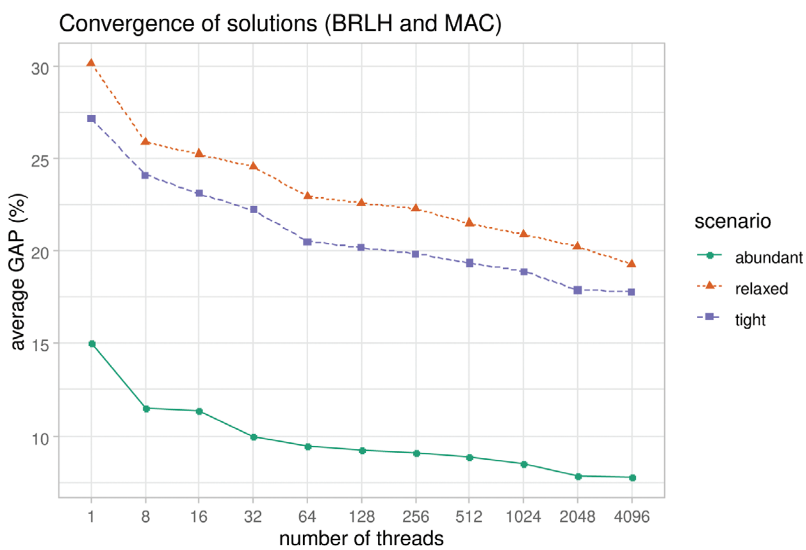

Fig. 7. Comparison between BRLH and MAC on solving large-sized instances with a different number of parallel runs.

6. Conclusions

In this paper, we proposed a BR algorithm to solve a 2E-VRP. In our case, this problem is motivated by the distribution of drugs with drones in disaster situations, where the affected areas might become no longer accessible by conventional cargo vehicles. This might be the case in rescue operations in humanitarian logistics, where every second can be decisive to save lives. Therefore, our methodology must be able to provide good solutions in a very short computational time. The concept of agile optimization was introduced in order to meet this goal.

As results show, the use of BR techniques together with parallel computing is able to improve about 10% the solutions of the deterministic version of our heuristic, without increasing the required wall-clock time. This is an attractive characteristic of the proposed approach, which allows us to take advantage of the current devices, which are more and more efficient nowadays.

In general, our methodology showed to be very competitive when solving both small- and large-sized instances. At this point, we contrast with the alternative solution methods that are efficient but require high computational times to provide high-quality solutions. In both set of instances, small- and large-sized, we were able to find results in real-time (within milliseconds) that were quite close to the best-known solutions. Particularly, in the abundant inventory scenario of large-sized instances, our methodology was able to generate a new better solution than the best-known solution. Increasing the number of parallel threads, the performance of our BR heuristic is even better. However, it is a limitation that depends on hardware settings.

Future works include reacting to unexpected events during the planning of the routes, such as to include new emergency locations, to consider the unavailability of stock caused, for instance, by the expansion of the disaster, etc. However, changes in the problem formulation are required in order to deal with this dynamic information and to propose an efficient solution methodology. When incorporating new information, our methodology becomes capable of dealing with the world dynamism, which is continuously changing and being affected by external circumstances and events.

Acknowledgments

This work has been partially supported by the IoF2020-H2020 and AGAUR 2018-LLAV-00017 projects, as well as by the Erasmus+ program (2018-1-ES01-KA103-049767).

References

Abdulkader, M.M.S., Gajpal, Y., ElMekkawy, T.Y., 2018. Vehicle routing problem in omni-channel retailing distribution systems. *International Journal of Production Economics* 196, 43–55.

Ai, T.J., Kachitvichyanukul, V., 2009. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research* 36, 5, 1693–1702.

Balasingam, M., 2017. Drones in medicine—the rise of the machines. *International Journal of Clinical Practice* 71, 9, e12989.

Bamburry, D., 2015. Drones: designed for product delivery. *Design Management Review* 26, 1, 40–48.

Belgin, O., Karaoglan, I., Altiparmak, F., 2018. Two-echelon vehicle routing problem with simultaneous pickup and delivery: mathematical model and heuristic approach. *Computers & Industrial Engineering* 115, 1–16.

Bianchessi, N., Righini, G., 2007. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research* 34, 2, 578–594.

Brandão, J.S., Noronha, T.F., Resende, M.G., Ribeiro, C.C., 2015. A biased random-key genetic algorithm for single-round divisible load scheduling. *International Transactions in Operational Research* 22, 5, 823–839.

Brandão, J.S., Noronha, T.F., Resende, M.G., Ribeiro, C.C., 2017. A biased random-key genetic algorithm for scheduling heterogeneous multi-round systems. *International Transactions in Operational Research* 24, 5, 1061–1077.

Breunig, U., Baldacci, R., Hartl, R.F., Vidal, T., 2019. The electric two-echelon vehicle routing problem. *Computers & Operations Research* 103, 198–210.

Calvet, L., Ferrer, A., Gomes, M.I., Juan, A.A., Masip, D., 2016. Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. *Computers & Industrial Engineering* 94, 93–104.

Çatay, B., 2010. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications* 37, 10, 6809–6817.

Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* 12, 4, 568–581.

Crainic, T.G., Mancini, S., Perboli, G., Tadei, R., 2013. GRASP with path relinking for the two-echelon vehicle routing problem. In Di Gaspero, L., Schaerf, A., Stützle, T. (eds) *Advances in Metaheuristics*. Springer, New York, pp. 113–125.

Crainic, T.G., Perboli, G., Mancini, S., Tadei, R., 2010. Two-echelon vehicle routing problem: a satellite location analysis. *Procedia-Social and Behavioral Sciences* 2, 3, 5944–5955.

Crainic, T.G., Ricciardi, N., Storchi, G., 2004. Advanced freight transportation systems for congested urban areas. *Transportation Research Part C: Emerging Technologies* 12, 2, 119–137.

Crispim, J., Brandão, J., 2005. Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society* 56, 11, 1296–1302.

Daknama, R., Kraus, E., 2017. Vehicle routing with drones, pp. 124. Available at arXiv:1705.06431v1.

De Armas, J., Juan, A.A., Marquès, J.M., Pedroso, J.P., 2017. Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society* 68, 10, 1161–1176.

Dominguez, O., Guimarans, D., Juan, A.A., de la Nuez, I., 2016. A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research* 255, 2, 442–462.

Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 1, 70–85.

Faulin, J., Gilibert, M., Juan, A.A., Vilajosana, X., Ruiz, R., 2008. SR-1: A simulation-based algorithm for the capacitated vehicle routing problem. 2008 Winter Simulation Conference. IEEE, pp. 2708–2716.

Faulin, J., Juan, A.A., 2008. The ALGACEA-1 method for the capacitated vehicle routing problem. *International Transactions in Operational Research* 15, 5, 599–621.

Gajpal, Y., Abad, P., 2009. An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research* 36, 12, 3215–3223.

Gatteschi, V., Lamberti, F., Paravati, G., Sanna, A., Demartini, C., Lisanti, A., Venezia, G., 2015. *New frontiers of delivery services using drones: a prototype system exploiting a quadcopter for autonomous drug shipments. 2015 IEEE 39th Annual Computer Software and Applications Conference*, Vol. 2. IEEE, pp. 920–927.

Ghiani, G., Guerriero, F., Laporte, G., Musmanno, R., 2003. Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research* 151, 1, 1–11.

Goksal, F.P., Karaoglan, I., Altiparmak, F., 2013. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering* 65, 1, 39–53.

Gonçalves, J.F., Resende, M.G., 2011. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics* 17, 5, 487–525.

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.

Hemmelmayr, V.C., Cordeau, J.-F., Crainic, T.G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research* 39, 12, 3215–3228.

Jacobsen, S.K., Madsen, O.B., 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., Faulin, J., Ferrer, A., Lourenço, H.R., Barrios, B., 2013a. MIRHA: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *Top* 21, 1, 109–132.

Juan, A.A., Faulin, J., Jorba, J., Caceres, J., Marquès, J.M., 2013b. Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research* 207, 1, 43–65.

Juan, A.A., Faulín, J., Jorba, J., Riera, D., Masip, D., Barrios, B., 2011. On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. *Journal of the Operational Research Society* 62, 6, 1085–1097.

Juan, A.A., Faulin, J., Ruiz, R., Barrios, B., Caballé, S., 2010. The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing* 10, 1, 215–224.

Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A.A., Burke, E.K., 2016. A multi-agent based cooperative approach to scheduling and routing. *European Journal of Operational Research* 254, 1, 169–178.

McConnan, I., 2004. *Humanitarian charter and minimum standards in disaster response*. 2nd ed., Geneva, Switzerland: Sphere Association.

Migdalas, A., Pardalos, P.M., Storøy, S. (eds), 2013. *Parallel Computing in Optimization*, Vol. 7. Springer Science & Business Media, Berlin.

Min, H., 1989. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General* 23, 5, 377–386.

Mu, D., Wang, C., Zhao, F., Sutherland, J.W., 2016. Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm. *International Journal of Shipping and Transport Logistics* 8, 1, 81–106.

Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: extended models and connections. *Networks* 70, 1, 34–43.

Quick, J.D., Rankin, J.R., Laing, R.O., O'Connor, R.W., Hogerzeil, H.V., Dukes, M.N.G. & Garnett, A., 1997. *Managing Drug Supply: The Selection, Procurement, Distribution, and Use of Pharmaceuticals*. Kumarian Press, 2nd ed., West Hartford, CT.

Rey, A., Prieto, M., Gómez, J.I., Tenllado, C., Hidalgo, J.I., 2018. A CPU-GPU parallel ant colony optimization solver for the vehicle routing problem. In Sim, K., Kaufmann, P. (eds) *International Conference on the Applications of Evolutionary Computation*. Springer, Berlin, pp. 653–667.

Scott, J., Scott, C., 2017. Drone delivery models for healthcare. Proceedings of the 50th Hawaii International Conference on System Sciences, Hawaii.

Wan, W.M., Azrie, M.S., Shuib, A., 2018. Mechanism for drones delivering the medical first aid kits. *Advances in Transportation and Logistics Research* 1, 1, 303–315.

Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: several worst-case results. *Optimization Letters* 11, 4, 679–697.

Zeng, Z.-Y., Xu, W.-S., Xu, Z.-Y., Shao, W.-H., 2014. A hybrid GRASP+ VND heuristic for the two-echelon vehicle routing problem arising in city logistics. *Mathematical Problems in Engineering*, 2014 1–11. https://doi.org/10.1155/2014/51746.