European Journal of Operational Research 224 (2013) 189–208

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Lower and upper bounds for location-arc routing problems with vehicle capacity constraints

Seyed Hossein Hashemi Doulabi *, Abbas Seifi

Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
Industrial Engineering and Productivity Research Center, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 28 January 2011
Accepted 6 June 2012
Available online 28 June 2012

Keywords:
Location-arc routing
Arc routing
Integrated logistics
Mixed integer programming
Heuristics

ABSTRACT

This paper addresses multi-depot location arc routing problems with vehicle capacity constraints. Two mixed integer programming models are presented for single and multi-depot problems. Relaxing these formulations leads to other integer programming models whose solutions provide good lower bounds for the total cost. A powerful insertion heuristic has been developed for solving the underlying capacitated arc routing problem. This heuristic is used together with a novel location–allocation heuristic to solve the problem within a simulated annealing framework. Extensive computational results demonstrate that the proposed algorithm can find high quality solutions. We also show that the potential cost saving resulting from adding location decisions to the capacitated arc routing problem is significant.

1. Introduction

Many researchers have studied three major logistic problems, namely, facility location, inventory management, and vehicle routing problems, and solved them in a consecutive manner, from strategic to operational levels. However, these problems are greatly interrelated and must be solved simultaneously in order to minimize the total cost during the design phase of a logistic system. Since introduction the integrated logistic system in 1980, many researchers have studied a combination of a Vehicle Routing Problem (VRP) and the decision on depots locations, which is known as location routing problems (Albareda-Sambola et al., 2007; Barreto et al., 2007; Nagy and Salhi, 2007; Aksen and Altinkemer, 2008; Prodhon, 2011; Prins et al., 2007). However, very few studies have appeared in the literature focusing on Location-Arc Routing Problem (LARP). An Arc Routing Problem (ARP) is similar to a VRP except for customer demands that are distributed over some edges and arcs in a network and the objective is to minimize the cost of traversing all such required links. LARP appears where in addition to the routing, the best locations of depots must be chosen and becomes especially difficult when the number of depots is itself a decision variable. It is conceivable that both the number of depots and their locations would have significant impacts on the distribution cost. This is what we investigate in this paper.

Generally, ARP can be categorized into three main problems: (1) Chinese Postman problem in which all edges or arcs of a graph are served, (2) Rural Postman problem in which only a subset of edges or arcs is required to be served, and (3) Capacitated Arc Routing Problem (CARP) which was first introduced by Golden and Wong (1981). CARP is a more difficult problem than the first two problems due to the consideration of vehicles capacity constraints. Most of real-world applications on mail collection or delivery, street sweeping, snow removal, and road maintenance fall in this category for which various solution methods have been proposed (see for example Belenguer et al., 2010; Laporte et al., 2010; Almeida and Mourao, 2000; Amado and Mourao, 2005; Ghiani et al., 2005; Haastrup et al., 1998; Mansini and GraziaSperanza, 1998). Some researchers have developed heuristic algorithms for solving CARP (Muyldermans, 2003; Beullens et al., 2003). Others have focused on solution methods based on integer programming (Feillet et al., 2005; Bartolini et al., in press). There exists an interesting work on approximation algorithms in (Wøhlk, 2008). Different formulations of CARP have been cited in the literature. Golden and Wong (1981) proposed the first formulation of CARP based on an exponential number of subtour elimination constraints. They also presented a compact version of their model using flow variables. Belenguer and Benavent (1998, 2003) formulated the problem with an exponential number of constraints in which a single indexed variable was associated with each edge. They used a cutting plane algorithm to obtain strict lower bounds for CARP which was later extended for the case of mixed graph by Belenguer et al. (2006). Gouveia et al. (2010) proposed a compact formulation for the Mixed CARP using some flow variables. They also presented

* Corresponding author at: Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran.
E-mail address: hashemidoulabi@aut.ac.ir (S.H. Hashemi Doulabi).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.06.015

a relaxed version of their model and demonstrated its efficiency in obtaining strict lower bounds.

LARPs have been solved using exact and heuristic methods (Levy and Bodin, 1989; Ghiani and Laporte, 1999). Due to the complexity of LARP, little research has been reported on developing exact methods for solving this class of problem. A branch and cut algorithm was devised for solving a special case of LARP in which the routing subproblem was a Rural Postman problem (Ghiani and Laporte, 1999). There exist two classes of heuristic approaches for solving LARPs: Location–Allocation–Routing and Allocation–Routing–Location methods (Ghiani and Laporte, 2001; Laporte, 1988). In the former, the locations of depots are determined first, then the required arcs or edges (to be served) are allocated to some clusters, and finally a routing problem is solved within each cluster. Using this method, a real-world postal carrier scheduling problem was solved by Levy and Bodin (1989) for the United States postal services in which the capacity constraint was modeled as a soft constraint. In the second approach, which has been reported to be more effective (Ghiani and Laporte, 2001), clustering and routing are done first. Then, the locations of central depots are determined based on the constructed routes.

Ghiani and Laporte (2001) reviewed three main applications of LARP, namely, mail delivery, garbage collection and road maintenance. Liu et al. (2008) provided a comprehensive survey of LARP and suggested some potential future research on LARP. In most LARP applications, determining the best locations of depots is crucial. As such is the park-and-loop system used as a mode of mail delivery in the USA (Assad and Golden, 1995). Arc routing and scheduling with transshipment is another application in which the locations of depots could be considered as decision variables. As a real case of this situation, garbage is collected by specially equipped truck, taken to a transfer station, shredded and compacted, and finally transported to a dumping site by means of high-capacity trucks (De Rosa et al., 2002).

This paper is concerned with a more general case of LARP with vehicle capacity constraints and the possibility of choosing the locations of multiple depots. This problem is more realistic and has not been previously addressed in the literature. Two mixed integer programming models are presented herein for single and multi-depot LARPs. Relaxing these formulations leads to other integer programming models whose solutions provide good lower bounds for the total cost. A powerful insertion heuristic has been developed for solving the underlying capacitated arc routing problems. This heuristic is used together with a novel location–allocation heuristic in a simulated annealing framework which is designed to solve the problem. The rest of this paper has been organized as follows. Section 2 describes the problem definition. In Sections 3 and 4, mathematical formulations and the proposed solution method are presented. Extensive computational results are reported in Section 5 and some conclusions are made in Section 6.

2. Problem definition and notation

A LARP with vehicle capacity constraints is defined on a Graph $G' = (N, E \cup A')$ in which N is the set of nodes and $D \subseteq N$ is the set of potential locations of depots. E and A' are the sets of all edges and arcs of the graph, respectively. Let $A_R \subseteq A'$ and $E_R \subseteq E$ denote the sets of arc and edges on which the service is to be delivered. Each Link $(i, j) \in (A_R \cup E_R)$ contains a nonnegative demand q_{ij} with a service cost c_{ij}. Furthermore, a deadheading cost d_{ij} is incurred with each traversal of Link (i, j) without servicing that link. The problem is to choose the locations of an unknown number of depots and assign to each depot a number of tours, each of which is formed to serve a group of customers in such a way that the total cost of transportation, tours hiring and opening of depots is minimized. A hiring cost λ is considered for each tour that could be interpreted for example as the dumping cost in garbage collection. It should be noted that the number of selected depots must be less than a predetermined maximum number L. A constant opening cost G is also assumed for each newly established depot. Other classical assumptions used herein are as follows:

- Each required arc or edge must be served by exactly one vehicle.
- Each tour starts from and ends at the same depot and is assigned only to one vehicle.
- P is an upper bound on the number of tours assigned to each depot.
- Vehicles are homogeneous.
- The total demand to be served on each tour must not exceed the vehicle capacity W.
- Split delivery of customer demands is not permitted.

3. Mathematical formulations

Mathematical formulations presented herein are based on the flow variables proposed by Gouveia et al. (2010). In order to simplify the mathematical formulations, each Edge $(i, j) \in E$ is replaced by two opposite Arcs (i, j) and (j, i). Also, we make a copy of those nodes which are potential depots and denote them by D'. In fact each node $k \in D$ has a counterpart $k' \in D'$. Thereby, we define a directed Graph $G = (N \cup D', A)$ in which $A = A' \cup \{(i, j), (j, i): (i, j) \in E\} \cup \{(k, k'), (k', k): k \in D\}$; and $R \subseteq A$ is defined as the set of all required arc in the graph ($|R| = |A_R| + 2|E_R|$). Deadheading cost of all Arcs (k, k') and (k', k) is set to zero. In this section, the mathematical formulations for single and multiple depots LARP are constructed first. Then, a lower bound model is presented.

3.1. The single depot LARP formulation

The following variables are defined for a single depot LARP:

x_{ij}^p	Takes 1 if Link $(i, j) \in R$ is served by tour p, and 0 otherwise
y_{ij}^p	Is the number of times that arc $(i, j) \in A$ is deadheaded in tour p
f_{ij}^p	Is the pure flow in arc $(i, j) \in A$ in tour p.
$z_{k'}$	Takes 1 if node $k' \in D'$ is selected as a depot, and 0 otherwise
$DC_{k'}$	Total hiring cost associated with the tours assigned to node $k' \in D'$ if k' is selected to be a depot and 0 otherwise

In the mathematical models presented in this section, no node $k \in D$ will appear in the optimal solution as a selected depot. Instead, whenever the models select a dummy node $k' \in D'$ as a depot, its interpretation is that the corresponding counterpart (i.e., node $k \in D$) will be a depot in the original graph G'.

The formulation of single depot Location Arc Routing Problem (LARP_1) is then given by:

$$(\text{LARP}_1) \quad \text{Min} \sum_{p=1}^{P} \left[\sum_{(i,j) \in R} c_{ij} x_{ij}^p + \sum_{(i,j) \in A} d_{ij} y_{ij}^p \right] + \sum_{k' \in D'} DC_{k'} + \sum_{k' \in D'} (G \times z_{k'}) \tag{1}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij}^p + \sum_{j:(i,j)\in R} x_{ij}^p = \sum_{j:(j,i)\in A} y_{ji}^p + \sum_{j:(j,i)\in R} x_{ji}^p; \quad \forall i \in N \cup D', \quad \forall p \in \{1,\ldots,P\} \tag{2}$$

$$\sum_{p=1}^{P} x_{ij}^p = 1; \quad \forall (i,j) \in A_R \tag{3}$$

$$\sum_{p=1}^{P} (x_{ij}^p + x_{ji}^p) = 1; \quad \forall (i,j) \in E_R \tag{4}$$

$$y_{k'k}^p \leqslant z_{k'}; \quad \forall k' \in D', \quad \forall p \in \{1,\ldots,P\} \tag{5}$$

$$\sum_{j:(j,i)\in A} f_{ji}^p - \sum_{j:(i,j)\in A} f_{ij}^p = \sum_{j:(j,i)\in R} q_{ji} x_{ji}^p; \quad \forall i \in N, \quad \forall p \in \{1,\ldots,P\} \tag{6}$$

$$f_{k'k}^p \leqslant \sum_{(i,j)\in R} q_{ij} x_{ij}^p + M(1 - z_{k'}); \quad \forall k' \in D', \quad \forall p \in \{1,\ldots,P\} \tag{7}$$

$$f_{k'k}^p \geqslant \sum_{(i,j)\in R} q_{ij} x_{ij}^p - M(1 - z_{k'}); \quad \forall k' \in D', \quad \forall p \in \{1,\ldots,P\} \tag{8}$$

$$f_{k'k}^p \leqslant M \times z_{k'}; \quad \forall k' \in D', \quad \forall p \in \{1,\ldots,P\} \tag{9}$$

$$f_{kk'}^p = 0; \quad \forall k' \in D', \quad \forall p \in \{1,\ldots,P\} \tag{10}$$

$$f_{ij}^p \leqslant W\left(x_{ij}^p + y_{ij}^p\right); \quad \forall (i,j) \in A, \quad \forall p \in \{1,\ldots,P\} \tag{11}$$

$$DC_{k'} \geqslant \lambda\left(\sum_{p=1}^{P} y_{k'k}^p\right) - M(1 - z_{k'}); \quad \forall k' \in D' \tag{12}$$

$$\sum_{k'\in D'} z_{k'} = 1; \tag{13}$$

$$x_{ij}^p \in \{0,1\}; \quad \forall (i,j) \in R, \quad \forall p \in \{1,\ldots,P\} \tag{14}$$

$$z_{k'} \in \{0,1\}; \quad \forall k' \in D' \tag{15}$$

$$f_{ij}^p \geqslant 0; \quad \forall (i,j) \in A, \quad \forall p \in \{1,\ldots,P\} \tag{16}$$

$$y_{ij}^p \geqslant 0 \text{ integer}; \quad \forall (i,j) \in A, \quad \forall p \in \{1,\ldots,P\} \tag{17}$$

The objective function (1) is the summation of service cost, deadheading cost, hiring cost of tours and opening cost of depots. Eq. (2) shows the continuity of tours. Constraints (3) and (4) state that each required arc or edge must be serviced exactly in one tour. Constraint (5) implies that for each tour, the auxiliary arc (k', k) will be chosen at most once to exit from depot $k' \in D$. If $k' \in D$ is selected as a depot but tour p is not assigned to it, the left side of Constraint (5) will equal to 0. Constraints (6)–(10) are the flow conservation constraints. Generally, there exist two kinds of flow conservation constraints in this model as explained next:

(1) For each node $i \in N$, the flow conservation Constraint (6) ensures that, in each tour, the difference between inflow and outflow at this node must equal to the demand delivered to the entering arcs to node i.
(2) If a node $k' \in D$ happens to become a depot, Constraints (7) and (8) together state that, in each tour p, the total outgoing flow from depot k' must be equal to the total demand of all required arcs to be serviced in that tour. Constraint (9) also implies that if node $k' \in D$ is not selected to be a depot, the outgoing flow from this node will equal to zero. In addition, Constraint (10) ensures that, in each tour p, the returning flow to depot k' $\left(f_{kk'}^p\right)$ will equal to zero since no required arc is connected to this node. It does not make a difference whether or not node $k' \in D$ happens to be selected as a depot.

Constraint (11) defines an upper bound on the flow variables. In addition, it shows that a flow variable f_{ij}^p, can take a positive value only if arc (i, j) is traversed at least once in tour p. Constraint (12) calculates total hiring cost of all tours assigned to depot k' that is given by $\lambda\left(\sum_{p=1}^{P} y_{k'k}^p\right)$ when $z_{k'} = 1$. Constraint (13) implies that only one depot can be selected in the optimal solution.

If the location of the central depot is fixed by a decision maker, $LARP_1$ reduces to a basic flow based model of CARP similar to the one proposed by Gouveia et al. (2010). However, instead of making a copy of the central depot, they use such a constraint in their model $\sum_{j:(0,j)\in A} y_{0j}^p + \sum_{j:(0,j)\in A} x_{0j}^p \leqslant 1$ in which 0 denote the central depot. Obviously, this constraint does not allow each tour to visit the central depot more than once. We have resolved this problem by defining set D' and adding duplicate nodes to the graph. Doing so, if a decision maker considers node $k \in D$ as the central depot ($z_{k'} = 1$), each tour connected to depot k' can visit node $k \in D$ several times in its path. Gouveia et al. (2010) proved that their flow conservation constraints together with Constraint (11) would prevent possible subtours to appear in the solutions. In the above proposed model, if we set one $z_{k'}$ equal to 1, the reduced model becomes equivalent to a CARP formulation whose validity proof will be similar to that presented by Gouveia et al. (2010). Furthermore, they have proposed some valid inequalities which have been slightly modified for $LARP_1$ as follows:

$$\sum_{p=1}^{P}\left(y_{k'k}^p\right) \geqslant \left\lceil \frac{Q_T}{W} \right\rceil - M(1 - z_{k'}); \quad \forall k' \in D' \tag{18}$$

$$f_{ij}^p \geqslant q_{ij} x_{ij}^p; \quad \forall (i,j) \in R, \quad \forall p \in \{1,\ldots,P\} \tag{19}$$

$$f_{ij}^p \geqslant \left(y_{ij}^p - 1\right) \times \min_{(i',j')\in R}(q_{i'j'}); \quad \forall (i,j) \in A \setminus R, \quad \forall p \in \{1,\ldots,P\} \tag{20}$$

In equality (18), Q_T denotes the total demand of all required edges and arcs and is equal to $\sum_{(i,j)\in A_R \cup E_R} q_{ij}$. This inequality implies that the number of constructed tours must be greater than or equal to $\lceil Q_T/W \rceil$. Constraints (19) and (20) impose lower bounds on flow variables. This model can be extended to a multi-depot model for LARP as discussed in the next subsection.

3.2. The multi-depot LARP formulation

To extend the formulation of $LARP_1$ for the case of multiple depots, it is not possible to simply replace Constraint (13) with $1 \leqslant \sum_{k'\in D'} z_{k'} \leqslant L$ because it would not be clear from the solution which tours are assigned to which depots. Thus, we have to define an auxiliary label $l \in \{1,2,\ldots,L\}$ for the selected depots to assert which set of tours has been assigned to a selected depot labeled with l. In other words, each tour has to be characterized by a pair (p, l). An upper bound on the total number of tours that will be formed in the final solution is given by $|P \times L|$.

The following notations are used in the proposed multi-depot LARP model:

L	Denotes the maximum number of nodes that can be selected as depots
$T_{k',l}$	Takes the value of 1 if node $k' \in D$ is selected to be a depot ($z_{k'} = 1$) and label $l \in \{1,\ldots,L\}$ is assigned to it
$x_{ij}^{p,l}$	Assumes 1 if $(i, j) \in R$ is serviced in tour (p, l), and zero otherwise
$y_{ij}^{p,l}$	Equals to the number of times that arc $(i, j) \in A$ is deadheaded in tour (p, l)
$f_{ij}^{p,l}$	Denotes the flow traversing in arc $(i, j) \in A$ in tour (p, l)

The formulation of multi-depot Location-Arc Routing Problem (LARP_M) is then given by:

$$(\text{LARP}_M) \quad \min \sum_{l=1}^{L}\sum_{p=1}^{P}\left[\sum_{(i,j)\in R} c_{ij}x_{ij}^{p,l} + \sum_{(i,j)\in A} d_{ij}y_{ij}^{p,l}\right] + \sum_{k'\in D'} DC_{k'} + \sum_{k'\in D'}(G\times z_{k'}) \tag{21}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij}^{p,l} + \sum_{j:(i,j)\in R} x_{ij}^{p,l} = \sum_{j:(j,i)\in A} y_{ji}^{p,l} + \sum_{j:(j,i)\in R} x_{ji}^{p,l}; \quad \forall i\in N, \quad \forall p \in \{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\}. \tag{22}$$

$$\sum_{l=1}^{L}\sum_{p=1}^{P} x_{ij}^{p,l} = 1; \quad \forall (i,j)\in A_R \tag{23}$$

$$\sum_{l=1}^{L}\sum_{p=1}^{P}\left(x_{ij}^{p,l} + x_{ji}^{p,l}\right) = 1; \quad \forall (i,j)\in E_R \tag{24}$$

$$y_{k'k}^{p,l} \leqslant T_{k',l}; \quad \forall k'\in D', \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{25}$$

$$\sum_{j:(j,i)\in A} f_{ji}^{p,l} - \sum_{j:(i,j)\in A} f_{ij}^{p,l} = \sum_{j:(j,i)\in R} q_{ji}x_{ji}^{p,l}; \quad \forall i\in N, \quad \forall p \in\{1,\ldots,P\}, \forall l\in\{1,\ldots,L\} \tag{26}$$

$$f_{k'k}^{p,l} \leqslant \sum_{(i,j)\in R} q_{ij}x_{ij}^{p,l} + M(1-T_{k',l}); \quad \forall k'\in D', \quad \forall p \in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{27}$$

$$f_{k'k}^{p,l} \geqslant \sum_{(i,j)\in R} q_{ij}x_{ij}^{p,l} - M(1-T_{k',l}); \quad \forall k'\in D', \quad \forall p \in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{28}$$

$$f_{k'k}^{p,l} \leqslant M\times T_{k',l}; \quad \forall k'\in D', \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{29}$$

$$f_{kk'}^{p,l} = 0; \quad \forall k'\in D', \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{30}$$

$$f_{ij}^{p,l} \leqslant W\left(x_{ij}^{p,l} + y_{ij}^{p,l}\right); \quad \forall (i,j)\in A, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l \in\{1,\ldots,L\} \tag{31}$$

$$DC_{k'} \geqslant \lambda\left(\sum_{p=1}^{P} y_{kk'}^{p,l}\right) - M(1-T_{k',l}); \quad \forall k'\in D', \quad \forall l \in\{1,\ldots,L\} \tag{32}$$

$$\sum_{k'\in D'} z_{k'} \leqslant L; \tag{33}$$

$$\sum_{l=1}^{L} T_{k',l} = z_{k'}; \quad \forall k'\in D' \tag{34}$$

$$\sum_{k'\in D'} T_{k',l} \leqslant 1; \quad \forall l\in\{1,\ldots,L\} \tag{35}$$

$$x_{ij}^{p,l} \leqslant \sum_{k'\in D'} T_{k',l}; \quad \forall (i,j)\in A, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l \in\{1,\ldots,L\} \tag{36}$$

$$y_{ij}^{p,l} \leqslant M\times\sum_{k'\in D'} T_{k',l}; \quad \forall (i,j)\in A, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l \in\{1,\ldots,L\} \tag{37}$$

$$x_{ij}^{p,l} \in \{0,1\}; \quad \forall (i,j)\in R, \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{38}$$

$$z_{k'}\in\{0,1\}; \quad \forall k'\in D' \tag{39}$$

$$T_{k',l}\in\{0,1\}; \quad \forall k'\in D', \quad \forall l\in\{1,\ldots,L\} \tag{40}$$

$$f_{ij}^{p,l} \geqslant 0; \quad \forall (i,j)\in A, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{41}$$

$$y_{ij}^{p,l} \geqslant 0 \text{ integer}; \quad \forall (i,j)\in A, \forall p\in\{1,\ldots,P\}, \quad \forall l \in\{1,\ldots,L\} \tag{42}$$

Constraints (22)–(32) could be interpreted in a similar way to Constraints (2)–(12). The only difference is that $z_{k'}$ has been replaced by $T_{k',l}$. To see why $T_{k',l}$ is used instead of $z_{k'}$, we explain Constraints (27) and (28) as examples. In these constraints, if we use $z_{k'}$ instead of $T_{k',l}$, it leads to $f_{k'k}^{p,l} = \sum_{(i,j)\in R} q_{ij}x_{ij}^{p,l}$ when $z_{k'} = 1$. It means that for all tours (p, l), the outgoing flow from node k' to node k is equal to total demands to be served in that tour. However, we would expect to impose this constraint only on those tours originating from node k'. Therefore, it is necessary to use $T_{k',l}$ instead of $z_{k'}$ in Constraints (27) and (28). The replacement of $z_{k'}$ by $T_{k',l}$ in other constraints could be justified similarly.

Constraint (33) imposes an upper bound on the number of selected depots. If node $k'\in D'$ is selected as a depot, Constraint (34) assigns only one label to this depot. Constraint (35) implies that each label could be assigned to at most one node. Constraint (36) and (37) states if label l is not assigned to any node $k'\in D'$, then no tour (p, l) could be constructed and the corresponding variables $x_{ij}^{p,l}$ and $y_{ij}^{p,l}$ would be zero.

The following inequalities are valid for the multi-depot LARP model and are added during the solution procedure:

$$f_{ij}^{p,l} \geqslant q_{ij}x_{ij}^{p,l}; \quad \forall (i,j)\in R, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{43}$$

$$f_{ij}^{p,l} \geqslant \left(y_{ij}^{p,l} - 1\right)\times \min_{(i',j')\in R}(q_{i'j'}); \quad \forall (i,j)\in A\setminus R, \quad \forall p\in\{1,\ldots,P\}, \quad \forall l\in\{1,\ldots,L\} \tag{44}$$

$$\sum_{k'\in D'}(k'\times T_{k',l}) \geqslant \sum_{k'\in D'}(k'\times T_{k',l+1}); \quad \forall l\in\{1,\ldots,L-1\} \tag{45}$$

$$\sum_{k'\in D'} DC_{k'} \geqslant \lambda\times\left\lceil\frac{Q_T}{W}\right\rceil \tag{46}$$

Inequalities (43) and (44) are similar to (19) and (20) in the single depot case. In (45), the term $\sum_{k'\in D'}(k'\times T_{k',l})$ determines the index of a depot to which label l is assigned. This inequality breaks the following symmetries:

(1) If the number of selected depots in the final solution is less than L, this constraint behaves like $\sum_{k'\in D'} T_{k',l} \geqslant \sum_{k'\in D'} T_{k',l+1}$ and forces selection of labels with smaller index l.
(2) Labels from 1 to L are assigned to selected nodes with decreasing indices.

Valid inequality (46) imposes a lower bound on total hiring cost of tours in which $\lceil Q_T/W \rceil$ is the minimum number of required tours in a feasible solution.

3.3. The lower bounding model

To obtain a lower bound on the objective value of the multi-depot LARP, we extend the approach used by Gouveia et al. (2010) in which the variables are aggregated over the set of tours assigned to the depot with label l, i.e., $X^l_{ij} = \sum_{p=1}^{P} x^{p,l}_{ij}$, $Y^l_{ij} = \sum_{p=1}^{P} y^{p,l}_{ij}$ and $F^l_{ij} = \sum_{p=1}^{P} f^{p,l}_{ij}$. This aggregation is also applied to those constraints defined for each $p \in \{1,\ldots,P\}$. The relaxed model of LARP$_M$ may be given as:

$$(\text{LARP}_{\text{LB}})\quad min \sum_{l=1}^{L}\left[\sum_{(i,j)\in R} c_{ij} x^{p,l}_{ij} + \sum_{(i,j)\in A} d_{ij} y^{p,l}_{ij}\right] + \sum_{k'\in D'} DC_{k'} + \sum_{k'\in D'} (G \times z_{k'}) \tag{47}$$

Subject to:

$$\sum_{j:(i,j)\in A} Y^l_{ij} + \sum_{j:(i,j)\in R} X^l_{ij} = \sum_{j:(j,i)\in A} Y^l_{ji} + \sum_{j:(j,i)\in R} X^l_{ji};\quad \forall i \in N,\ \forall l \in \{1,\ldots,L\} \tag{48}$$

$$\sum_{l=1}^{L} X^l_{ij} = 1;\quad \forall (i,j) \in A_R \tag{49}$$

$$\sum_{l=1}^{L}\left(X^l_{ij} + X^l_{ji}\right) = 1;\quad \forall (i,j) \in E_R \tag{50}$$

$$Y^l_{k'k} \leqslant P \times T_{k',l};\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{51}$$

$$\sum_{j:(j,i)\in A} F^l_{ji} - \sum_{j:(i,j)\in A} F^l_{ij} = \sum_{j:(j,i)\in R} q_{ji} X^l_{ji};\quad \forall i \in N,\quad \forall l \in \{1,\ldots,L\} \tag{52}$$

$$F^l_{k'k} \leqslant \sum_{(i,j)\in R} q_{ij} X^l_{ij} + M(1 - T_{k',l});\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{53}$$

$$F^l_{k'k} \geqslant \sum_{(i,j)\in R} q_{ij} X^l_{ij} - M(1 - T_{k',l});\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{54}$$

$$F^l_{k'k} \leqslant M \times T_{k',l};\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{55}$$

$$F^l_{kk'} = 0;\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{56}$$

$$F^l_{ij} \leqslant W\left(X^l_{ij} + Y^l_{ij}\right);\quad \forall (i,j) \in A,\quad \forall l \in \{1,\ldots,L\} \tag{57}$$

$$DC_{k'} \geqslant \lambda \times Y^l_{kk'} - M(1 - T_{k',l});\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\} \tag{58}$$

$$\sum_{k'\in D'} z_{k'} \leqslant L; \tag{59}$$

$$\sum_{l=1}^{L} T_{k',l} = z_{k'};\quad \forall k' \in D' \tag{60}$$

$$\sum_{k'\in D'} T_{k',l} \leqslant 1;\quad \forall l \in \{1,\ldots,L\} \tag{61}$$

$$X^l_{ij} \leqslant \sum_{k'\in D'} T_{k',l};\quad \forall (i,j) \in A,\quad \forall l \in \{1,\ldots,L\} \tag{62}$$

$$Y^l_{ij} \leqslant M \times \sum_{k'\in D'} T_{k',l};\quad \forall (i,j) \in A,\quad \forall l \in \{1,\ldots,L\} \tag{63}$$

$$F^l_{ij} \geqslant q_{ij} X^l_{ij};\quad \forall (i,j) \in R,\quad \forall p \in \{1,\ldots,P\} \tag{64}$$

$$F^l_{ij} \geqslant \left(Y^l_{ij} - L\right) \times \min_{(i,j)\in R}(q_{ij});\quad \forall (i,j) \in A \setminus R,\ \forall p \in \{1,\ldots,P\} \tag{65}$$

$$\sum_{k'\in D'} (k' \times T_{k',l}) \geqslant \sum_{k'\in D'} (k' \times T_{k',l+1});\quad \forall l \in \{1,\ldots,L-1\} \tag{66}$$

$$\sum_{k'\in D'} DC_{k'} \geqslant \lambda \times \left\lceil \frac{Q_T}{W} \right\rceil \tag{67}$$

$$X^l_{ij} \in \{0,1\};\quad \forall (i,j) \in R,\quad \forall l \in \{1,\ldots,L\} \tag{68}$$

$$z_{k'} \in \{0,1\};\quad \forall k' \in D' \tag{69}$$

$$T_{k',l} \in \{0,1\};\quad \forall k' \in D',\quad \forall l \in \{1,\ldots,L\}. \tag{70}$$

$$F^l_{ij} \geqslant 0;\quad \forall (i,j) \in A,\quad \forall l \in \{1,\ldots,L\} \tag{71}$$

$$Y^l_{ij} \geqslant 0 \text{ integer};\quad \forall (i,j) \in A,\quad \forall l \in \{1,\ldots,L\} \tag{72}$$

The number of variables and constraints in LARP$_{\text{LB}}$ model is significantly less than those of LARP$_M$ and thus the model is expected to be solved in a much shorter time. We solve this model to find good lower bounds on the objective values of the solutions to the multi-depot LARP$_M$. However, the solution of this model may become infeasible for LARP$_M$ since it may not necessarily satisfy the capacity constraints of the vehicles.

4. Solution method

The proposed solution method is a simulated annealing (SA) framework consisting of an arc routing and a location–allocation heuristic. SA is an enhanced local search algorithm whose inspiration is the process of cooling hot substances slowly. The main goal of SA is to achieve the lowest level of energy in the system. Thus, if the cooling process is slow enough, substances move from a high level of energy to a lower level. However, in some cases, transferring to a higher energy level could possibly enable the algorithm to skip from local optima, but the probability of such moves decreases during the search procedure.

The flowchart of the proposed method is shown in Fig. 1. The problem data is input into the algorithm first. Then, an initial solution is produced by the routing heuristic and sent to the location–allocation heuristic for improving the depot locations. These heuristics search the solution space in an iterative manner until no further improvement is possible. The next step in the algorithm is to decide whether the current solution should be replaced by a previously found solution. The stopping criteria are checked next and the algorithm continues if they are not met. Next, in the neighboring solution generating module, the tours in a current solution are decomposed to some neighboring subtours and sent to the heuristic modules. In fact, the generated subtours are meant to form a basis for finding a better routing solution. In the following section, the arc routing heuristic, the location–allocation heuristic as well as the solution generating operator are explained.

4.1. The proposed insertion arc routing heuristic

The proposed heuristic has been designed to meet the following criteria:

- It should be fast enough so that it can be used many times in an iterative manner.
- It must be capable of admitting a neighboring solution as an input and constructing a complete solution.

The main idea is to insert small subtours in each other so that it leads to some cost saving. The steps of the insertion heuristic are as follows. In the following steps, whenever we mention a cluster, we point to a series of arcs that start from a depot and end at the same depot.

Step 1: If this procedure is called for the first time and its input is the initial data, then, generate a trivial solution with one tour for each required edge or arc. All of these singleton tours are

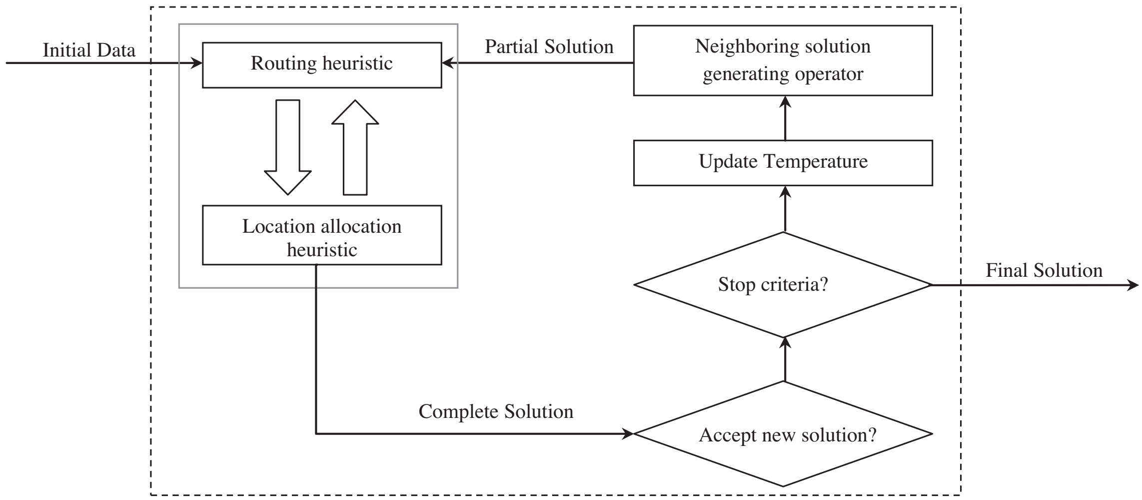

Fig. 1. Flowchart of the proposed simulated annealing algorithm.

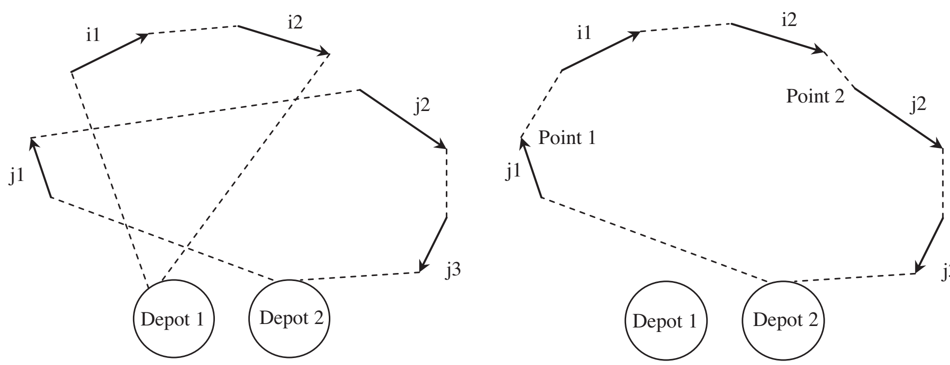

Fig. 2. Clusters *i* and *j* before and after insertion using sequence Type 1.

connected to a potential depot such that the best objective value is obtained. If the input is a neighboring solution coming from the SA algorithm or the location–allocation heuristic, this is ignored.

Step 2: Calculate an improvement matrix *I* with dimension $n \times n$ (n is the number of current clusters) whose entry (i, j) is the cost saving obtained by inserting cluster i into cluster j. Such an entry (i, j) is computed by considering all possible places where cluster i can be inserted into cluster j. In other words, all deadheading paths in the cluster j will be examined to choose the insertion place. The corresponding link is removed from cluster j and then cluster i will be connected to cluster j using two shortest deadheading paths. It should be noted that if an insertion caused violation of vehicle capacity in the new constructed tour, such an insertion would be illegal and the corresponding entry in matrix I would be set to zero.

An example of such insertion is depicted in Fig. 2. In this example, the deadheading path that connects $j1$ to $j2$ is the selected insertion place to be removed. The endpoint of the first part of cluster j (point 1) is connected to $i1$; and $i2$ is connected to the

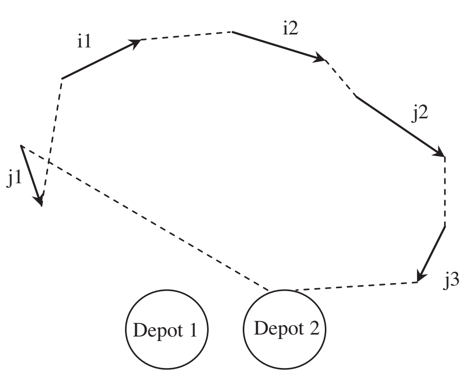

Fig. 3. Clusters *i* and *j* after insertion using sequence Type 2.

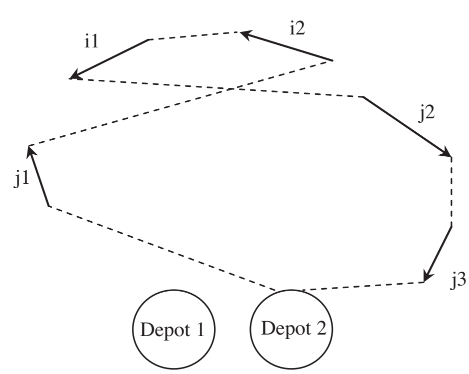

Fig. 4. Clusters *i* and *j* after insertion using sequence Type 3.

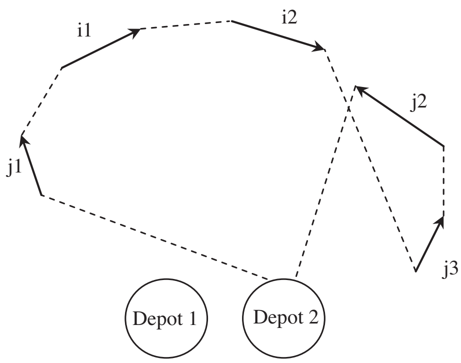

Fig. 5. Clusters *i* and *j* after insertion using sequence Type 4.

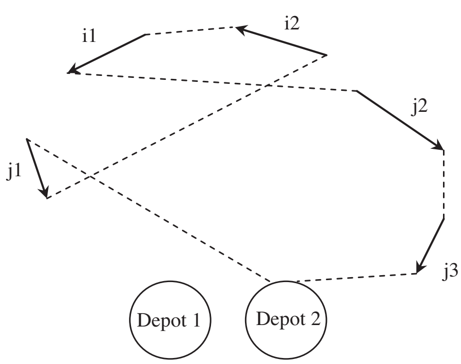

Fig. 6. Clusters *i* and *j* after insertion using sequence Type 5.

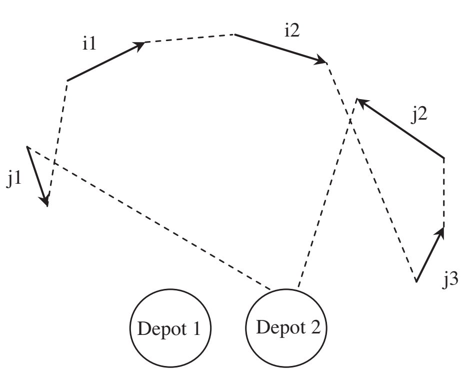

Fig. 7. Clusters *i* and *j* after insertion using sequence Type 6.

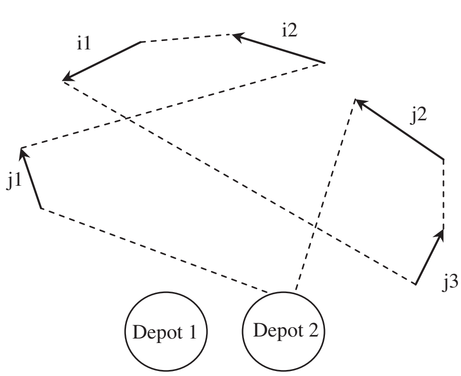

Fig. 8. Clusters *i* and *j* after insertion using sequence Type 7.

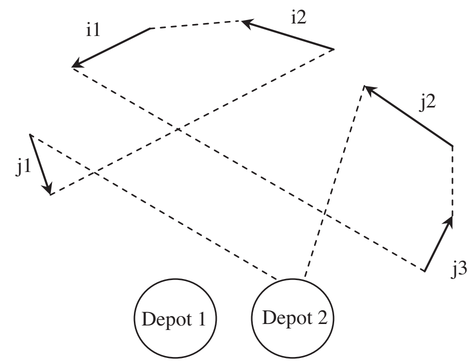

Fig. 9. Clusters *i* and *j* after insertion using sequence Type 8.

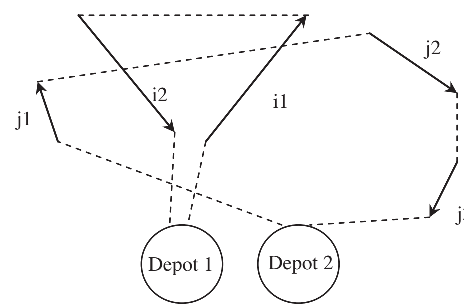

Fig. 10a. Clusters *i* and *j* before insertion.

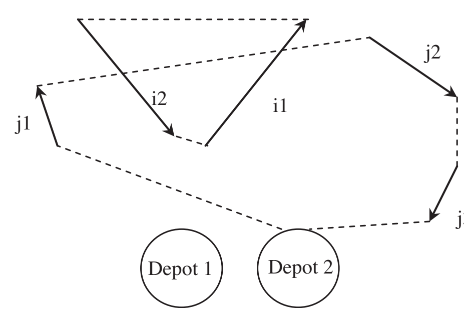

Fig. 10b. Clusters *i* and *j* after converting cluster *i* to a loop.

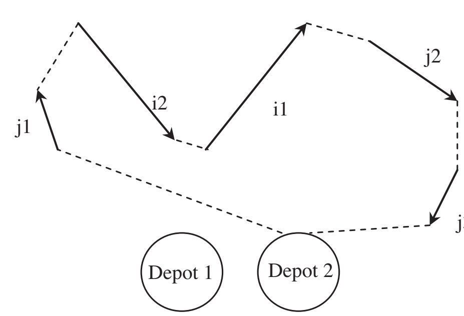

Fig. 10c. The output cluster after removing the longest deadheading path of loop *i* and inserting it to cluster *j*.

initial point of the second part of cluster *j* (point 2). Also, the deadheading paths which connect cluster *i* to depot 1 are removed and the new cluster is constructed. In addition, various directions on which cluster *i* and two separate parts of cluster *j* may be traversed are tested. It means that if B denotes

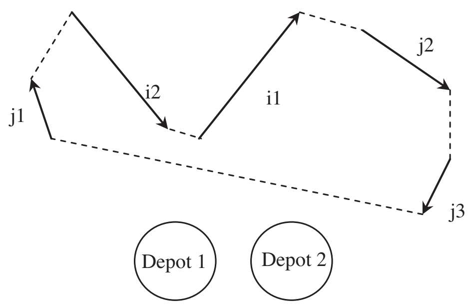

Fig. 10d. Conversion of the output cluster to a loop.

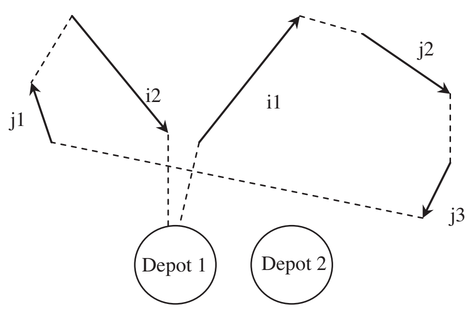

Fig. 10e. Connection of the obtained loop to the closest depot.

cluster *i* and two parts of cluster *j* are denoted by A1 and A2, the following sequences must be examined (B′ is in the opposite direction of B):

(1) Depot 2–A1–B–A2 -Depot 2	(2) Depot 2 - A1′–B–A2–Depot 2
(3) Depot 2–A1–B′–A2–Depot 2	(4) Depot 2–A1–B–A2′–Depot 2
(5) Depot 2–A1′–B′–A2–Depot 2	(6) Depot 2–A1′–B–A2′–Depot 2
(7) Depot 2–A1–B′–A2′–Depot 2	(8) Depot 2–A1′–B′–A2′–Depot 2

Since the problem is defined on a mixed graph, some of these sequences are not permitted and should not be examined. The advantage of evaluating such sequences is that the proposed algorithm could potentially find near optimal solutions by examining a number of permutations. For the clusters depicted in the right hand side of Fig. 2, the combined cluster of sequences (2)–(8) are shown in Figs. 3–9. In this example, it is assumed that all of the required links are edges and could be traversed in two opposite directions.

In addition, to improve Step 2, cluster *i* can be inserted more intelligently. To do so, after removing two deadheading paths which connect cluster *i* to depot 1, we connect the end point of the last required arc in cluster *i* to the starting point of the first required arc in that cluster which creates a loop. Then, the longest deadheading

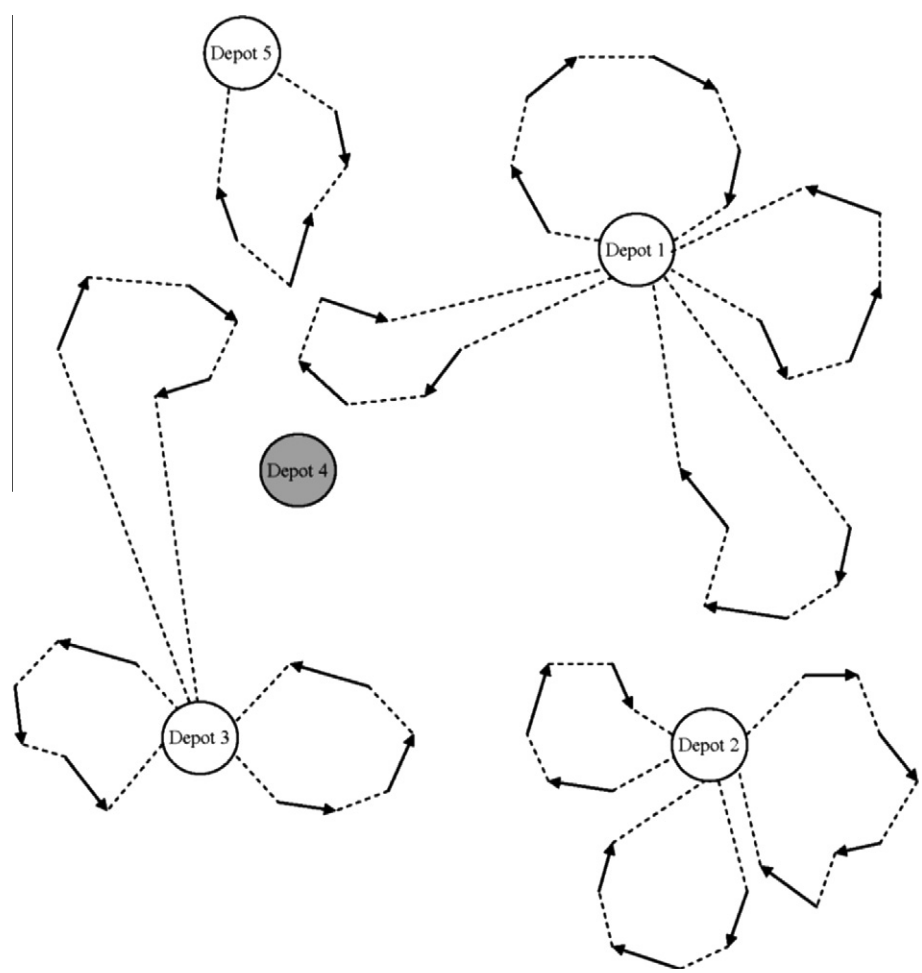

Fig. 11. A feasible solution before applying location allocation heuristic.

path in loop i is removed and the obtained route is inserted into cluster $j$as described earlier in Step 2. Doing so, it is expected to have more cost saving by removing the longest deadheading path from cluster i. If we apply this idea to clusters i and j depicted in the right hand side of Fig. 2, the resulting clusters will be the same as the previous ones depicted in Figs. 2–9. As another example, two other clusters i and j are shown before and after insertion in Figs. 10a, 10b, 10c using sequence of Type 1.
Another point to reduce the transportation cost is to reconnect the combined cluster to a proper depot. This is done by removing its two deadheading connections to depot 1 and converting it to a loop. The resulting loop is then tested for connecting to depot 1 or depot 2 to get the most cost saving. For Example 2, this improvement is shown in Figs. 10d and 10e.

Step 3: Given an improvement Matrix I, the insertion that corresponds to the largest improvement is made in the current solution and Matrix I is updated. It means that if entry (i, j) is selected as the best insertion choice, then after insertion, the row and the column corresponding to cluster i will be eliminated and those of cluster j will be updated. The computation efforts needed for this update is limited only to the row and column corresponding to cluster j.

Step 4: The algorithm stops if there is no possible insertion (due to vehicle capacity) or all possible insertions would increase the objective value. Otherwise, continue with Step 3.

There is a main difference between the proposed heuristic and other insertion heuristics in the literature like those presented by Chapleau et al. (1984) and Pearn (1991). Our heuristic is an improvement method starting from a trivial solution while insertion heuristics in the literature are constructive heuristics. A comparable heuristic to the one proposed herein is the Improved Merge heuristic developed by Belenguer et al. (2006). However, a key difference with Improved Merge heuristic is that it only merges some cluster pairs, while the proposed heuristic inserts a pair of smaller clusters into each other with many more possibilities of insertion leading to greater potential for cost saving.

4.2. The proposed location–allocation heuristic

The aim of location allocation heuristic is to improve the objective value by changing selected depots and tour assignments. To provide a better insight about this heuristic, we consider an example and explain it step by step throughout. The steps of location allocation heuristic are as follows:

Step 1: At first, define two following sets:

$$Located\_Depots = \{i \in D | i \text{ is a depot in the current solution}\}$$

$$Strongly\_Located\_Depots = \emptyset$$

For the example depicted in Fig. 11, we have *Located_Depots* = {1,2,3,5}. In the next steps, some depots in the current solution may be replaced by some new depots in order to improve the

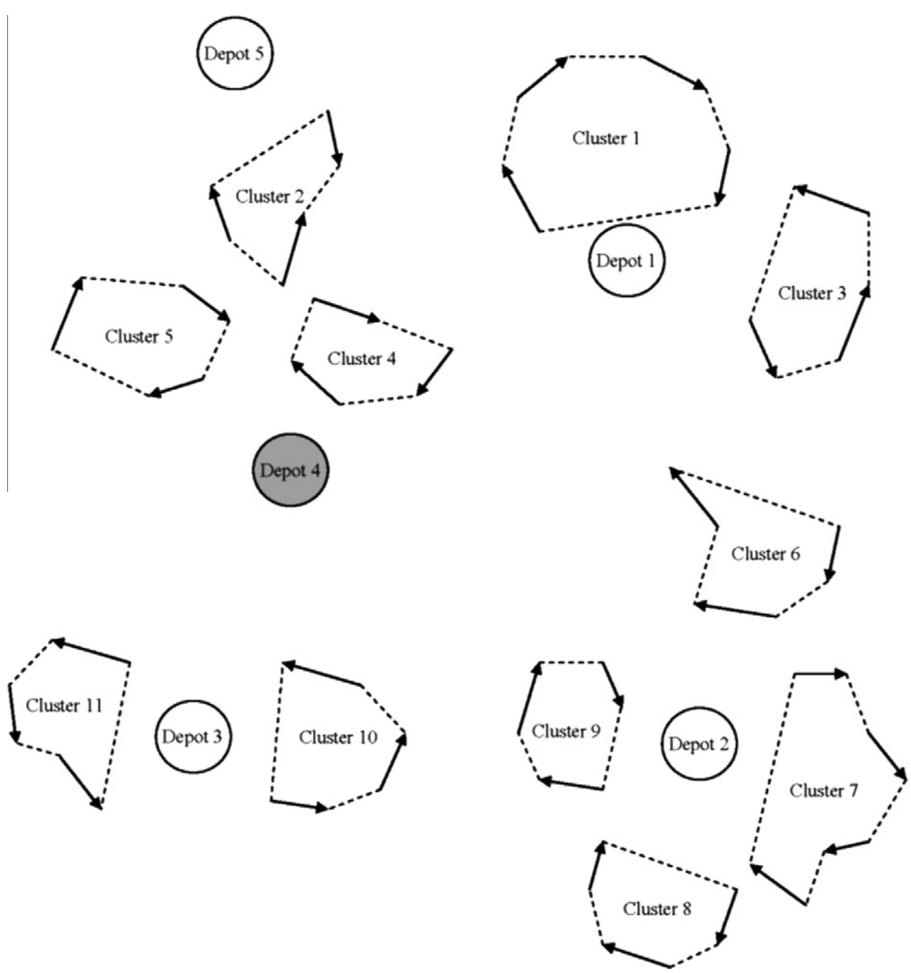

Fig. 12. Solution of Fig. 11 after the 2nd step of location allocation heuristic.

objective value. *Strongly_Located_Depots* includes depots whose replacement by any other depot will not improve the total cost while members of *Located_Depots* are those nodes whose replacement may improve the objective value.*Step 2*: All tours are disconnected from their corresponding depots and are transformed into separate clusters. For the example, the created clusters are shown in Fig. 12. These clusters will be assigned to some depots in the next steps.

Step 3: Select a node j from *Located_Depots* randomly. If the number of current depots is equal to L (maximum allowable number of depots), then eliminate depot j from the current solution. In this example, $L = 4$ and Depot 1 is selected and eliminated.

Step 4: Among all nodes $i \in D$ which have not yet assigned as selected depots, add the one to the current solution which improves the objective value the most. In examining each scenario of adding node i as a depot to the current solution, each cluster k will be assigned to either its previous depot (case I) or to the new depot i (case II). We also let clusters be disconnected from node j and connect to other depots available in

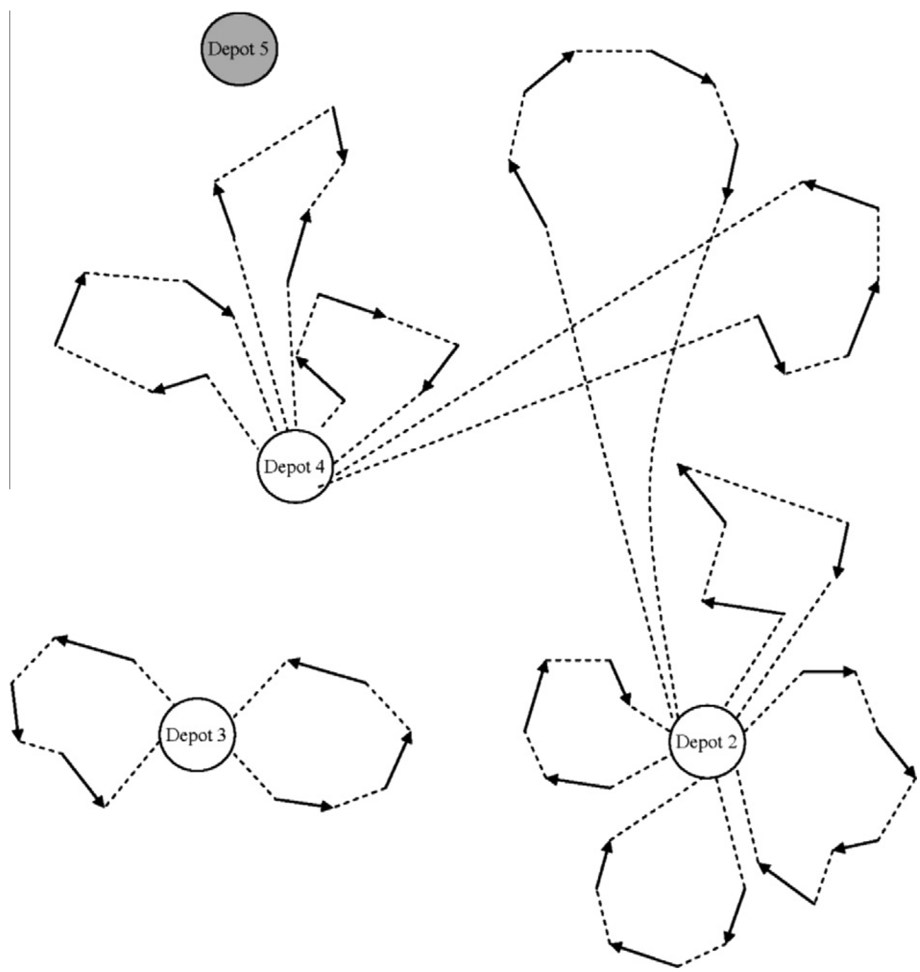

Fig. 13. Solution of Fig. 12 after the 3rd step of location allocation heuristic.

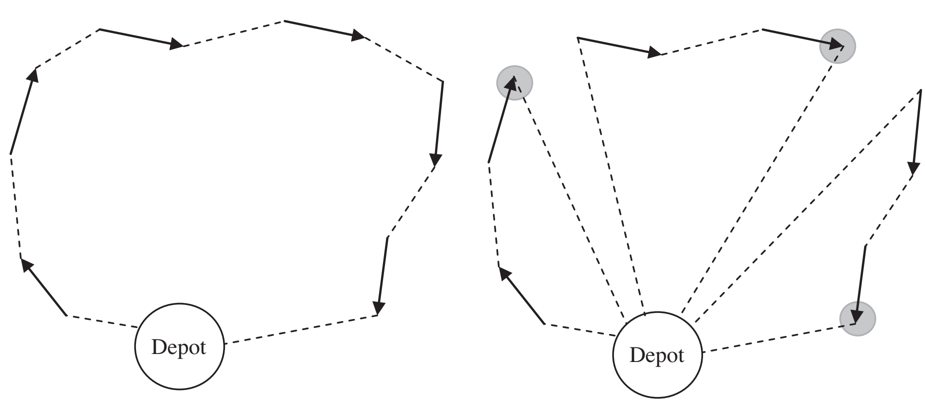

Fig. 14. An example of applying a partial solution generation operator on a single cluster.

the current solution (case III). In this step, if adding node $i \in D$ does not improve the objective value, then no new depot will be added to the solution anymore.

Returning to the previous example, node 4 is selected as a new depot and the solution after clusters assignment is depicted in Fig. 13. In this example, for clusters 2–5, case II is performed. Assignment of clusters 1 and 6 are done based on case III. The other clusters are assigned through case I. At the end of this step, if no tour is connected to a depot, this depot will be eliminated from the solution. An example of such depot is Depot 5 in Fig. 13.

Step 5: If node *j*, mentioned in Step 3, is added as a depot to the current solution and its tours have not changed as compared to its tours before Step 3, set *Strongly_Located_Depots*?*Strongly_Located_Depots* $\cup$ {*j*} and *Located_Depots*: = *Located_Depots* $-$ {*j*}. Otherwise, set *Strongly_Located_Depots* = $\varnothing$ and *Located_Depots* = {$i \in D$|*i is a depot in the current solution*}. Indeed, *Strongly_Located_Depots* includes nodes whose last eliminations have not reduced the total cost. But whenever an improvement happens, regarding to the change in the current solution, elimination of nodes in *Strongly_Located_Depots* may help algorithm find a better solution.

Step 6: If all depots in the current solution are in *Strongly_Located_Depots*, the algorithm stops. Otherwise, it continues with Step 2.

As explained in Step 3, if the number of depots is less than *L*, the algorithm will allow the solution to open a new depot without eliminating an existing depot *j*. Also, at the end of Step 4, some depots may be eliminated from the current solution. As a result, the heuristic has some flexibility in increasing or decreasing the number of depots during its search. This flexibility is an advantage of the heuristic which provides the capability of determining an appropriate number of depots without using another heuristic. In fact, the above heuristic uses routing and clustering information in order to determine suitable number of depots. In addition, the proposed location heuristic does not get trapped in a cycling repetition since it tries to improve the current solution at every iteration and it will stop if no further significant improvement is possible.

At the end of this step, if the solution has been improved, the iterative search continues with the arc routing heuristic. Otherwise, the iterative loop formed by the routing and location heuristics is stopped and it will be checked whether the number of tours connected to each depot is less than the predetermined maximum number of allowable tours (*P*). If this constraint is not satisfied by the solution, using the insertion heuristics proposed in Section 4.1, some tours will be inserted into each other in order to satisfy this constraint even though it might lead to some increase in cost.

Then, the algorithm goes to the next step in order to decide whether or not the current solution should be replaced by a previously found solution. In this step, if the objective value of the current solution is better than that of the previously found solution, the current solution will be kept. Otherwise, we replace the current solution by the previously found solution with a probability of $1 - \exp(-|\Delta f|/T)$ where $|\Delta f|$ is the absolute difference of the objective values of the two solutions and *T* is the temperature in the SA

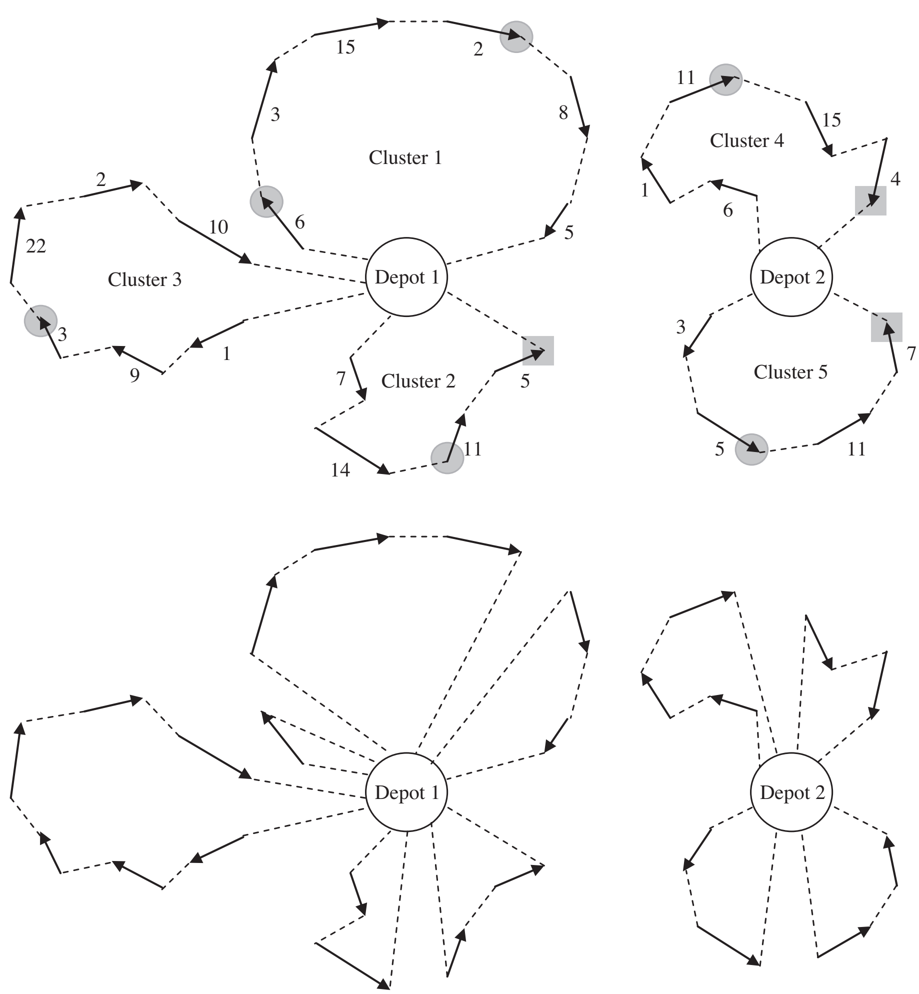

Fig. 15. An example of applying the proposed SA operator on a solution.

Table 1
Evaluation of the quality of the lower bound in solving some *mval* instances for the case of $G = 0$.

File	$\|N\|$	$\|A'UE\|$	$\|A_R\|$	$\|E_R\|$	P	LARP ($L = l$)				
						Optimal method		Lower bound method		Gap (%)
						Cost	Time	Cost	Time	
Mval1a	24	55	20	35	4	28	741	28	0	0
Mval1b	24	51	13	38	5	72	1011	72	0	0
Mval2a	24	44	16	28	4	83	12,086	83	0	0
Mval2b	24	52	12	40	5	53	24,149	53	0	0
Mval3a	24	48	15	33	4	23	11,772	23	0	0
Mval3b	24	45	16	29	5	45	34,119	45	0	0
Mval4a	41	95	26	69	5	95	77,649	95	0	0
Mval4b	41	102	19	83	6	120	54,291	120	0	0
Mval4c	41	103	21	82	7	97	92,419	97	0	0
Mval5a	34	96	22	74	5	72	55,169	72	0	0
Mval5b	34	91	35	56	6	45	67,909	45	0	0
Mval5c	34	98	17	81	7	122	75,121	122	0	0
Mval6a	31	69	22	47	5	61	42,001	61	0	0
Mval6b	31	66	22	44	6	66	28,072	66	3	0
Mval7a	40	86	36	50	5	37	148,038	37	0	0
Mval7b	40	91	25	66	6	66	110,652	66	0	0
Mval8a	30	96	20	76	5	54	47,084	54	0	0
Mval8b	30	91	27	64	6	47	16,482	47	0	0
Mval9a	50	132	32	100	5	67	193,201	67	0	0
Mval9b	50	120	44	76	6	96	169,111	96	0	0
Mval10a	50	138	32	106	5	85	211,376	85	0	0
Mval10b	50	134	33	101	6	149	204,932	149	0	0
	Average					72	76,245	72	0	0

Table 2
Evaluation of the proposed insertion heuristic in solving *mval* data set for the case of $G = 0$.

File	$\|N\|$	$\|A'UE\|$	$\|A_R\|$	$\|E_R\|$	P	Fixed serviced cost	CARP								
							Merge		Insertion		PSRC		IURL		LB
							Cost	Time	Cost	Time	Cost	Time	Cost	Time	
Mval1a	24	55	20	35	4	202	31	6	*29*	20.5	41	0.02	29	0.00	28
Mval1b	24	51	13	38	5	189	*73*	5	81	15	125	0.01	103	0.01	72
Mval1c	24	53	17	36	10	203	143	4.5	*133*	10	224	0.02	154	0.00	106
Mval2a	24	44	16	28	4	241	103	4	*88*	17	168	0.01	133	0.01	83
Mval2b	24	52	12	40	5	298	97	5.5	*97*	15.5	173	0.01	136	0.00	97
Mval2c	24	49	14	35	10	270	*282*	4	*282*	7.5	374	0.01	331	0.00	251
Mval3a	24	48	15	33	4	90	35	4.5	*28*	19.5	43	0.01	38	0.00	25
Mval3b	24	45	16	29	5	83	*64*	4	*64*	12.5	79	0.01	67	0.01	59
Mval3c	24	43	18	25	9	73	*97*	3.5	*97*	6	118	0.01	119	0.00	93
Mval4a	41	95	26	69	5	471	150	18.5	*124*	84.5	228	0.03	213	0.01	109
Mval4b	41	102	19	83	6	496	185	20.5	*171*	81.5	279	0.03	241	0.01	154
Mval4c	41	103	21	82	7	512	163	20.5	*143*	75	316	0.03	228	0.01	118
Mval4d	41	104	21	83	11	531	*270*	20.5	273	56.5	484	0.03	374	0.01	215
Mval5a	34	96	22	74	5	525	145	18.5	*117*	69.5	208	0.03	158	0.01	72
Mval5b	34	91	35	56	6	536	*101*	18	106	58.5	182	0.03	141	0.01	77
Mval5c	34	98	17	81	7	548	*179*	18.5	187	60.5	261	0.03	263	0.01	149
Mval5d	34	92	29	63	11	525	*262*	17	266	40.5	358	0.03	330	0.01	194
Mval6a	31	69	22	47	5	265	88	9.5	*70*	30	127	0.02	102	0.01	61
Mval6b	31	66	22	44	6	251	73	8.5	*72*	25.5	155	0.02	103	0.00	66
Mval6c	31	68	23	45	12	258	*126*	8	132	16.5	268	0.02	186	0.01	107
Mval7a	40	86	36	50	5	327	45	14.5	*40*	50.5	112	0.03	63	0.01	37
Mval7b	40	91	25	66	6	346	115	15	*93*	55.5	161	0.03	145	0.01	66
Mval7c	40	90	28	62	11	337	106	14.5	*102*	35.5	241	0.03	167	0.01	87
Mval8a	30	96	20	76	5	527	103	18.5	*89*	75	139	0.03	124	0.01	54
Mval8b	30	91	27	64	6	484	*62*	17	*62*	61	135	0.03	127	0.01	47
Mval8c	30	83	28	55	11	444	248	13.5	*245*	30.5	398	0.03	318	0.01	173
Mval9a	50	132	32	100	5	391	94	34.5	*91*	144.5	138	0.05	123	0.02	67
Mval9b	50	120	44	76	6	357	157	30	*119*	131.5	195	0.04	145	0.02	96
Mval9c	50	125	42	83	7	372	89	32	*80*	118	157	0.05	126	0.01	56
Mval9d	50	131	38	93	12	405	146	33	*142*	93	290	0.05	217	0.01	109
Mval10a	50	138	32	106	5	545	140	39	*124*	181	190	0.05	160	0.03	89
Mval10b	50	134	33	101	6	504	*182*	37	185	158.5	249	0.05	210	0.02	157
Mval10c	50	136	36	100	7	519	149	38.5	*141*	144	232	0.02	195	0.00	104
Mval10d	50	129	42	87	12	485	*205*	34.5	*205*	94.5	362	0.01	275	0.01	158
# Of dominated cases							7		21		0		0		
Gap (%)			Maximum				101.39%		64.81%		202.70%		170.21%		
			Average				33.67%		24.34%		115.70%		73.01%		

Table 3
Evaluation of the proposed insertion heuristic in solving *Ipr* data set for the case of $G = 0$.

File	$\lvert N\rvert$	$\lvert A'UE\rvert$	$\lvert A_R\rvert$	$\lvert E_R\rvert$	P	Fixed service cost	CARP								
							Merge		Insertion		PSRC		IURL		LB
							Cost	Time	Cost	Time	Cost	Time	Cost	Time	
lpr-a-01	28	94	52	0	2	12,440	1157	5	*1081*	14.5	1160	0.02	1097	0.01	1044
lpr-a-02	53	169	99	5	3	26,242	*2057*	20.5	2238	112.5	2852	0.04	2344	0.02	1810
lpr-a-03	146	469	271	33	8	72,033	4938	186	*4761*	901.5	7050	0.22	6118	0.07	4082
lpr-a-04	195	651	469	34	13	119,701	*8836*	503	9056	3229	13,354	0.42	12,183	0.14	7245
lpr-a-05	321	1056	748	58	20	189,306	*18,555*	1373	19,010	8599	25,847	0.97	22,861	0.30	13,428
lpr-b-01	28	63	45	5	9	13,291	1634	5	*1616*	18.5	1756	0.02	1577	0.00	1544
lpr-b-02	53	117	92	9	3	26,032	3169	20	*2775*	78.5	3490	0.04	2915	0.01	2622
lpr-b-03	163	361	279	26	8	70,890	8997	185.5	*7545*	919	9127	0.18	9020	0.07	6969
lpr-b-04	248	582	493	8	15	114,840	*14,609*	494	14,665	3322.5	19,114	0.38	17,401	0.12	12,092
lpr-b-05	401	876	764	37	22	188,512	26,774	1353.5	*26,266*	8577.5	34,961	0.80	31,190	0.27	21,326
lpr-c-01	28	52	11	39	9	17,797	866	6.5	*844*	25	1100	0.02	909	0.01	842
lpr-c-02	53	101	23	77	5	34,417	2020	28	*1996*	135.5	2512	0.04	2346	0.01	1922
lpr-c-03	163	316	61	241	12	105,132	*7241*	240.5	7314	1218.5	10,631	0.22	9407	0.06	5831
lpr-c-04	277	604	142	362	20	160,685	10,317	687.5	*9906*	2937.5	13,731	0.55	12,476	0.15	7717
lpr-c-05	369	841	416	387	29	244,929	17,132	1589	*16,469*	6799	23,439	1.03	21,129	0.31	12,890
# Of dominated cases							5		10		0		0		
Gap	Max.						38.18%		41.57%		92.49 %		70.25%		
	Ave.						20.43%		17.28%		54.76%		38.18%		

Table 4
Evaluation of the proposed SA in solving *mval* data set for the case of $G = 0$.

File	CARP	LARP ($L = 1$)					LARP ($L = 2$)					LARP ($L = 3$)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
Mval1a	28	28	0	28	0	0.00	28	5	28	6	0.00	28	16	28	16	0.00
Mval1b	72	72	39	72	0	0.00	72	0	72	1	0.00	72	0	72	8	0.00
Mval1c	112	112	900	104	3	7.69	75	900	70	302	7.14	65	900	63	352	3.17
Mval2a	83	83	10	83	0	0.00	83	0	83	3	0.00	83	5	83	1	0.00
Mval2b	97	53	0	53	0	0.00	53	3	53	3	0.00	53	184	53	11	0.00
Mval2c	256	119	900	109	5	9.17	75	900	62	210	20.97	43	193	43	552	0.00
Mval3a	25	23	12	23	0	0.00	23	0	23	4	0.00	23	0	23	10	0.00
Mval3b	59	45	17	45	0	0.00	45	0	45	3	0.00	45	9	45	20	0.00
Mval3c	93	41	900	37	5	10.81	29	900	26	1026	11.54	20	900	16	3600	25.00
Mval4a	109	95	171	95	0	0.00	95	0	95	3	0.00	95	216	95	31	0.00
Mval4b	154	120	713	120	0	0.00	120	32	120	9	0.00	120	311	120	27	0.00
Mval4c	118	102	900	97	0	5.15	100	900	97	6	3.09	97	49	98	47	1.02-
Mval4d	239	125	900	115	10	8.70	101	900	87	2406	16.09	93	900	85	475	9.41
Mval5a	72	72	61	72	0	0.00	72	141	72	3	0.00	72	219	72	16	0.00
Mval5b	77	47	900	45	0	4.44	45	116	45	14	0.00	45	132	45	194	0.00
Mval5c	149	122	23	122	0	0.00	122	33	122	3	0.00	122	35	122	2	0.00
Mval5d	214	93	900	87	10	6.90	66	572	66	1207	0.00	66	900	61	3600	8.20
Mval6a	61	63	900	61	0	3.28	63	900	61	2	3.28	63	900	61	57	3.28
Mval6b	66	66	524	66	3	0.00	62	50	62	9	0.00	62	515	62	20	0.00
Mval6c	113	103	900	94	10	9.57	60	900	52	3057	15.38	47	900	45	1406	4.44
Mval7a	37	37	184	37	0	0.00	37	5	37	14	0.00	37	59	37	28	0.00
Mval7b	66	67	900	66	0	1.52	67	900	66	2	1.52	67	900	66	30	1.52
Mval7c	89	91	900	80	6	13.75	77	900	57	3600	35.09	70	900	56	3600	25.00
Mval8a	54	54	56	54	0	0.00	54	7	54	2	0.00	54	64	54	38	0.00
Mval8b	47	49	900	47	0	4.26	47	739	47	13	0.00	47	508	47	818	0.00
Mval8c	194	149	900	112	4	33.04	90	900	83	127	8.43	83	709	83	84	0.00
Mval9a	67	68	900	67	0	1.49	68	900	67	18	1.49	68	900	67	108	1.49
Mval9b	96	96	900	96	0	0.00	96	828	96	6	0.00	96	649	96	248	0.00
Mval9c	57	54	900	52	1	3.85	54	900	52	57	3.85	54	900	52	1111	3.85
Mval9d	115	95	900	88	10	7.95	89	900	73	3600	21.92	89	900	72	3600	23.61
Mval10a	89	86	900	85	0	1.18	85	312	85	9	0.00	85	416	85	72	0.00
Mval10b	157	154	900	149	0	3.36	154	900	149	26	3.36	153	900	149	118	2.68
Mval10c	104	107	900	96	0	11.46	97	900	96	39	1.04	97	900	95	3600	2.11
Mval10d	164	116	900	98	16	18.37	106	900	77	483	37.66	85	900	74	3600	14.86
Gap	Max.					33.04%					37.66%					25.00%
	Ave.					4.88%					5.64%					3.75%
Marginal location Improvement	Max.	56.54%				41.75%				42.67%						
	Ave.	13.82%				8.41%				4.62%						
Total location Improvement	Max.	56.54%				70.70%				83.20%						
	Ave.	13.82%				19.90%				22.06%						

* Calculated by Belenguer et al. (2006).

Table 5
Evaluation of the proposed SA in solving *mval* data set for the case of $G = 0.5\lceil c_{ave} \rceil$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
Mval1a	32	32	0	32	0	0.00	32	5	32	3	0.00	32	0	32	8	0.00
Mval1b	76	76	55	76	0	0.00	76	0	76	0	0.00	76	794	76	3	0.00
Mval1c	116	116	900	108	7	7.41	83	900	78	356	6.41	81	900	75	1214	8.00
Mval2a	87	87	14	87	0	0.00	87	209	87	4	0.00	87	708	87	1	0.00
Mval2b	100	56	0	56	1	0.00	56	5	56	2	0.00	56	0	56	1	0.00
Mval2c	260	123	900	113	2	8.85	84	900	70	562	20.00	55	154	55	3600	0.00
Mval3a	27	25	6	25	0	0.00	25	5	25	1	0.00	25	20	25	0	0.00
Mval3b	61	47	15	47	1	0.00	47	0	47	0	0.00	47	108	47	1	0.00
Mval3c	95	43	900	39	18	10.26	31	900	30	1506	3.33	26	900	22	3600	18.18
Mval4a	113	99	132	99	0	0.00	99	0	99	0	0.00	99	591	99	1	0.00
Mval4b	158	124	562	124	0	0.00	124	744	124	1	0.00	124	265	124	2	0.00
Mval4c	122	106	900	101	0	4.95	106	900	101	7	4.95	106	900	101	23	4.95
Mval4d	243	129	900	119	25	8.40	109	900	95	1438	14.74	101	900	95	769	6.32
Mval5a	76	76	59	76	0	0.00	76	530	76	4	0.00	76	58	76	12	0.00
Mval5b	81	51	900	49	0	4.08	51	900	49	27	4.08	51	900	49	142	4.08
Mval5c	153	126	26	126	0	0.00	126	388	126	0	0.00	126	261	126	1	0.00
Mval5d	218	97	900	91	14	6.59	96	900	74	1111	29.73	93	900	73	3600	27.40
Mval6a	65	67	900	65	0	3.08	67	900	65	1	3.08	67	900	65	43	3.08
Mval6b	70	70	689	70	8	0.00	70	743	70	14	0.00	70	421	70	22	0.00
Mval6c	117	107	900	98	19	9.18	66	900	60	2044	10.00	66	900	57	2722	15.79
Mval7a	41	41	139	41	0	0.00	41	196	41	7	0.00	41	22	41	20	0.00
Mval7b	70	71	900	70	3	1.43	71	900	70	2	1.43	71	900	70	12	1.43
Mval7c	93	95	900	84	5	13.10	84	900	65	2717	29.23	82	900	65	3600	26.15
Mval8a	57	57	56	57	0	0.00	57	22	57	4	0.00	57	4	57	27	0.00
Mval8b	51	53	900	51	0	3.92	53	900	51	7	3.92	53	900	51	509	3.92
Mval8c	198	153	900	116	25	31.90	100	900	91	2901	9.89	99	900	91	1154	8.79
Mval9a	70	71	900	70	0	1.43	71	900	70	34	1.43	71	900	70	80	1.43
Mval9b	99	99	900	99	0	0.00	99	805	99	12	0.00	99	453	99	358	0.00
Mval9c	60	57	900	55	1	3.64	57	900	55	69	3.64	57	900	55	216	3.64
Mval9d	118	98	900	91	22	7.69	92	900	79	3600	16.46	87	900	79	3600	10.13
Mval10a	92	89	900	88	0	1.14	89	900	88	9	1.14	89	900	88	19	1.14
Mval10b	160	157	900	152	0	3.29	157	900	152	13	3.29	157	900	152	98	3.29
Mval10c	107	110	900	99	0	11.11	109	900	99	44	10.10	108	900	99	87	9.09
Mval10d	167	119	900	101	12	17.82	110	900	83	3600	32.53	99	900	83	3600	19.28
Gap	Max.					31.90%					32.53%					27.40%
	Ave.					4.68%					6.16%					5.18%
Marginal location	Max.	55.50%					38.32%					34.52%				
Improvement	Ave.	13.48%					5.99%					2.45%				
Total location	Max.	55.50%					67.69%					78.85%				
Improvement	Ave.	13.48%					17.88%					18.98%				

* Calculated by Belenguer et al. (2006).

algorithm. After doing a number of iterations at each temperature, the temperature is decreased using $T := \alpha \times T$ where α is the cooling rate parameter. This will be continued until the time limit is reached.

4.3. The neighboring solution generating operator

In order to generate a neighboring solution, we have developed an operator which produces a neighboring solution through breaking some clusters to smaller clusters. Breaking down the clusters allows the proposed routing heuristic to insert and combine these smaller clusters in order to find better solutions. An example of applying such operator on a singleton cluster is illustrated in Fig. 14. While there are many possibilities for breaking the clusters, the proposed operator breaks down the cluster so that the total demands of the required arcs in each of the resulting smaller cluster are close to each other. This logic would help the insertion heuristic to combine pairs of small clusters without violating vehicle capacity constraints. This is done as follows. First, a cluster i and two of its endpoints are selected randomly and the summation of demands over the required arcs between these endpoints is denoted by Q_i. For each of the other clusters, two endpoints are selected in such a way that the summation of demands over the required arcs between those endpoints be close to Q_i as much as possible. Hence, for each cluster j, an endpoint $j1$ is chosen randomly and in order to select the second endpoint, it proceeds from $j1$ in cluster j to find an endpoint which would be as far as possible to the first endpoint and the summation of demands over the required arcs between those endpoints be less than or equal to Q_i. An example of applying this operator is depicted in Fig. 15. In this example, cluster 1 and its two endpoints are selected first ($Q_1 = 3 + 15 + 2 = 20$). Then the first endpoints in the other clusters are chosen and highlighted by grey circles. Finally, the second endpoint in each cluster (except for cluster 1) is selected and shown by a grey square. In cluster 3, it is impossible to select the second endpoint in such a way that the total demand is less than Q_1. Thus, cluster 3 is not broken into smaller clusters.

5. Computational results

Two standard problem sets of Mixed CARP known as *mval* and *lpr* have been solved to evaluate the performance of the proposed

Table 6
Evaluation of the proposed SA in solving *mval* data set for the case of $G = \lceil c_{ave} \rceil$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
Mval1a	39	35	0	35	0	0.00	35	1	35	2	0.00	35	0	35	12	0.00
Mval1b	84	80	63	80	0	0.00	80	12	80	1	0.00	80	60	80	2	0.00
Mval1c	123	119	900	111	4	7.21	89	900	84	437	5.95	89	67	84	891	5.95
Mval2a	95	91	16	91	0	0.00	91	174	91	2	0.00	91	785	91	4	0.00
Mval2b	106	59	0	59	8	0.00	59	108	59	1	0.00	59	0	59	1	0.00
Mval2c	267	126	900	116	11	8.62	89	900	76	709	17.11	64	29	64	3600	0.00
Mval3a	30	26	4	26	0	0.00	26	0	26	1	0.00	26	10	26	0	0.00
Mval3b	65	49	35	49	10	0.00	49	0	49	1	0.00	49	0	49	1	0.00
Mval3c	99	45	900	41	22	9.76	35	900	34	1791	2.94	32	504	28	3600	14.29
Mval4a	120	102	152	102	0	0.00	102	0	102	1	0.00	102	48	102	1	0.00
Mval4b	165	127	601	127	1	0.00	127	308	127	1	0.00	127	136	127	1	0.00
Mval4c	129	109	900	104	0	4.81	109	900	104	3	4.81	109	90	104	36	4.81
Mval4d	250	132	900	122	29	8.20	113	900	101	1209	11.88	108	210	101	3600	6.93
Mval5a	83	79	47	79	0	0.00	79	721	79	6	0.00	79	627	79	7	0.00
Mval5b	88	54	900	52	0	3.85	54	900	52	30	3.85	54	0	52	124	3.85
Mval5c	160	129	39	129	0	0.00	129	91	129	1	0.00	129	41	129	6	0.00
Mval5d	225	100	900	94	19	6.38	95	900	80	1453	18.75	95	130	80	3600	18.75
Mval6a	72	70	900	68	0	2.94	70	900	68	3	2.94	70	123	68	38	2.94
Mval6b	77	73	471	73	6	0.00	73	91	73	19	0.00	73	576	73	24	0.00
Mval6c	124	110	900	101	21	8.91	69	900	66	2101	4.55	68	426	66	3600	3.03
Mval7a	48	44	211	44	0	0.00	44	543	44	6	0.00	44	280	44	16	0.00
Mval7b	77	74	900	73	5	1.37	74	900	73	1	1.37	74	106	73	11	1.37
Mval7c	100	98	900	87	7	12.64	91	900	71	2094	28.17	88	642	71	3600	23.94
Mval8a	63	60	76	60	0	0.00	60	22	60	5	0.00	60	0	60	15	0.00
Mval8b	58	56	900	54	0	3.70	56	900	54	3	3.70	56	260	54	476	3.70
Mval8c	205	156	900	119	48	31.09	107	900	97	3025	10.31	107	346	97	2114	10.31
Mval9a	75	73	900	72	0	1.39	73	900	72	45	1.39	73	435	72	69	1.39
Mval9b	105	102	900	102	1	0.00	102	331	102	15	0.00	102	288	102	125	0.00
Mval9c	66	60	900	58	5	3.45	60	900	58	134	3.45	60	362	58	208	3.45
Mval9d	123	100	900	93	34	7.53	100	900	83	3600	20.48	100	515	83	3600	20.48
Mval10a	98	92	900	91	0	1.10	91	594	91	10	0.00	91	792	91	11	0.00
Mval10b	166	160	900	155	1	3.23	160	900	155	12	3.23	160	804	155	67	3.23
Mval10c	113	113	900	102	1	10.78	113	900	102	49	10.78	113	357	102	71	10.78
Mval10d	173	122	900	104	8	17.31	110	900	89	3600	23.60	108	105	89	3600	21.35
Gap	Max.					31.09%					28.17%					23.94%
	Ave.					4.54%					5.27%					4.72%
Marginal location	Max.	55.56%					37.27%					28.09%				
Improvement	Ave.	16.75%					5.38%					1.40%				
Total Location	Max.	55.56%					66.67%					76.03%				
Improvement	Ave.	16.75%					20.57%					21.14%				

* Calculated by Belenguer et al. (2006).

algorithms in solving LARP problems. The *val* instances have originated from directed graphs called *val* instances in Benavent et al. (1992). Belenguer et al. (2006) converted *val* instances to mixed graphs through replacing some edges by arcs. The set *mval* contains 34 networks which includes 24–50 nodes and 51–138 links. In this benchmark, although the name of some instances are similar (for example Mval1a, Mval1b and Mval1c), the corresponding networks are somewhat different. In fact, this kind of naming has been used by Belenguer et al. (2006) to generate the instances some of which have identical graphs with different vehicle capacities but after converting to mixed graphs, the instances with similar names have been differentiated from each other. Generally, for a fixed instance number (for example mval2*x* where *x* could be a, b and c), the vehicle capacity is decreasing from the first instance "a" to the last instances "c". The *lpr* set contains 15 bigger instances including 28–401 nodes and 52–1056 links. This set of instances mimics real street networks. These instances have been categorized to three groups namely a, b, c. The *lpr-a* instances have been used to model modern cities in which the corresponding network is directed and symmetric. Old cities have been modeled through *lpr-b* instances whose graphs are rather directed and anti-symmetric. The *lpr-c* instances are also representatives of low-traffic residential areas with a lot of streets and two-sided streets, i.e., their graphs are undirected. These test problems are available online at http://www.uv.es/~belengue/mcarp/index.html.

To convert these Mixed CARP instances to LARP instances, we have considered all nodes in each instance to be in the set of potential depots and the values of P (maximum number of tours) have been fixed to the values set by Gouveia et al. (2010) as presented in Tables 2 and 3. The proposed algorithms have been coded in MATLAB 7.9 and run on a computer with 3.0 gigahertz processor and 4.0 gigabyte of RAM. The time limit of 15 minute and 1 hour were set, respectively, for solving *mval* and *lpr* instances when solved using the proposed SA algorithm. The initial temperature was set to 20 and 60 for *mval* and *lpr* instances, respectively, and it was updated every 10,000 iterations. The cooling rate was set to 0.99. To obtain lower bounds, the mixed integer programming model presented in Section 3.3 has been implemented and solved using CPLEX 12.1 with a time limit of 1 hour for all instances. It must be noted that this model has been solved as a mixed integer programming not a linear programming.

An important point in solving all these instances is that the total service cost for the required links is a constant in any solution. We have removed such a big constant from the objective function in

Table 7
Evaluation of the proposed SA in solving *mval* data set for the case of $G = 2\lceil c_{ave} \rceil$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
Mval1a	53	42	0	42	1	0.00	42	790	42	3	0.00	42	410	42	4	0.00
Mval1b	100	88	77	88	1	0.00	88	173	88	0	0.00	88	17	88	0	0.00
Mval1c	137	126	900	118	14	6.78	103	900	98	539	5.10	103	900	98	1303	5.10
Mval2a	111	99	12	99	0	0.00	99	70	99	0	0.00	99	121	99	0	0.00
Mval2b	118	65	4	65	12	0.00	65	4	65	2	0.00	65	212	65	2	0.00
Mval2c	281	133	900	123	18	8.13	103	74	90	1870	14.44	85	8	85	3600	0.00
Mval3a	36	29	8	29	1	0.00	29	186	29	0	0.00	29	26	29	1	0.00
Mval3b	73	53	39	53	6	0.00	53	143	53	1	0.00	53	18	53	1	0.00
Mval3c	107	49	900	45	26	8.89	43	900	42	2981	2.38	43	900	40	3600	7.50
Mval4a	134	109	198	109	4	0.00	109	10	109	1	0.00	109	217	109	2	0.00
Mval4b	179	134	501	134	5	0.00	134	897	134	0	0.00	134	116	134	1	0.00
Mval4c	143	116	900	111	1	4.50	116	900	111	6	4.50	116	900	111	10	4.50
Mval4d	264	139	900	129	37	7.75	133	900	115	762	15.65	133	900	115	3600	15.65
Mval5a	97	86	32	86	0	0.00	86	215	86	8	0.00	86	326	86	5	0.00
Mval5b	102	61	900	59	1	3.39	61	900	59	37	3.39	59	619	59	99	0.00
Mval5c	174	136	189	136	0	0.00	136	48	136	0	0.00	136	127	136	0	0.00
Mval5d	239	107	900	101	16	5.94	107	900	94	1065	13.83	107	900	94	3600	13.83
Mval6a	86	77	900	75	0	2.67	77	900	75	4	2.67	75	518	75	6	0.00
Mval6b	91	80	158	80	8	0.00	80	96	80	36	0.00	80	150	80	26	0.00
Mval6c	138	117	900	108	18	8.33	87	900	80	2033	8.75	87	900	80	3600	8.75
Mval7a	62	51	301	51	0	0.00	51	262	51	3	0.00	51	170	51	8	0.00
Mval7b	91	81	900	80	3	1.25	81	900	80	1	1.25	81	900	80	1	1.25
Mval7c	114	105	900	94	5	11.70	101	900	85	1903	18.82	101	900	85	3600	18.82
Mval8a	75	66	89	66	1	0.00	66	11	66	6	0.00	66	0	66	5	0.00
Mval8b	72	63	900	61	0	3.28	63	900	61	7	3.28	63	900	61	9	3.28
Mval8c	219	163	900	126	38	29.37	122	900	111	3138	9.91	122	900	111	3138	9.91
Mval9a	85	78	900	77	0	1.30	78	900	77	51	1.30	78	900	77	47	1.30
Mval9b	117	108	900	108	3	0.00	108	515	108	17	0.00	108	549	108	20	0.00
Mval9c	78	66	900	64	1	3.13	66	900	64	113	3.13	66	900	64	149	3.13
Mval9d	133	105	900	98	39	7.14	105	900	93	3600	12.90	105	900	93	3600	12.90
Mval10a	110	98	900	97	0	1.03	98	900	97	11	1.03	98	900	97	12	1.03
Mval10b	178	166	900	161	0	3.11	166	900	161	7	3.11	166	900	161	13	3.11
Mval10c	125	119	900	108	5	10.19	119	900	108	48	10.19	119	900	108	38	10.19
Mval10d	185	128	900	110	13	16.36	116	900	101	3600	14.85	116	900	101	3600	14.85
Gap	Max.					29.37%					18.82%					18.82%
	Ave.					4.24%					4.43%					3.97%
Marginal location	Max.	55.23%					25.64%					17.48%				
Improvement	Ave.	21.15%					3.57%					0.69%				
Total location	Max.	55.23%					63.35%					69.75%				
Improvement	Ave.	21.15%					23.67%					23.98%				

* Calculated by Belenguer et al. (2006).

order to show more realistic results on the gap between the lower and upper bounds.

5.1. Evaluation of the Lower bounding method

To evaluate the quality of lower bounds obtained by the model presented in Section 3.3, The *mval* instances are solved by the mathematical model presented in Section 3.1 for the case of $L = 1$ and the lower bound values are compared to the optimal values. In solving the model, it assumed that the location cost is equal to 0 ($G = 0$). The model is also allowed to search for optimal solutions with a time limit of 72 hours. To speed up finding optimal solutions, we have provided the lower bound value for the exact model by imposing the constraint of *Objective Function* $\geqslant$ *LB*. The results of those instances that have been solved to optimality within the given time limit are reported in Table 1. In all reported results in Table 1, the lower bound values are equal to the optimal values which demonstrate the quality of solutions to the lower bounding model.

5.2. Evaluation of the proposed insertion heuristic for the Mixed CARP

In this section, we evaluate the ability of the proposed insertion heuristic in solving Mixed CARP instances without combining it with the neighboring solutions generation heuristic described in Section 4.3. This heuristic is compared with three other routing heuristics developed by Belenguer et al. (2006). The first one is called Improved Merge (IM) heuristic that is briefly described at the end of Section 4.1. The second heuristic, called PSRC, is a modified version of the classical path scanning heuristic. Belenguer et al. (2006) showed that PSRC would dominate the classical path scanning heuristic when is used for solving most of CARP benchmarks. The third one, named IURL, is a combination of Ulusoy algorithm (Ulusoy, 1985) and another modified version of path scanning called PSRL.

Since all of these heuristics involve some randomness in some of their steps, each instance has been solved 50 times using each algorithm and the total runtime and the objective values associated with their best solutions are reported in Tables 2 and 3. These algorithms have been coded as described by Belenguer et al. (2006), but for the PSRC and IURL algorithms, we found that the quality of solutions reported in the original paper is better. Therefore, in Tables 2 and 3, we have reported the published results for PSRC and IURL. On the other hand, our IM implementation performs better than the original code and we have reported our results for IM.

In these tables, the first six columns give some information about the problem instances. Fixed service costs are also reported

Table 8
Evaluation of the proposed SA in solving *lpr* data set for the case of $G = 0$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
lpr-a-01	1044	1044	40	1044	0	0.00	1044	384	1044	0	0.00	1044	517	1044	1	0.00
lpr-a-02	1810	1801	3600	1724	0	4.47	1801	3600	1724	5	4.47	1790	3600	1724	1	3.83
lpr-a-03	4122	4315	3600	3953	17	9.16	4315	3600	3834	3600	12.55	4181	3600	3834	3600	9.05
lpr-a-04	7651	6989	3600	6024	365	16.02	6495	3600	5724	3600	13.47	6492	3600	5646	3600	14.98
lpr-a-05	16,193	12,249	3600	9673	3600	26.63	10,931	3600	9403	3600	16.25	10,869	3600	9346	3600	16.30
lpr-b-01	1544	1483	0	1483	0	0.00	1483	0	1483	0	0.00	1483	0	1483	0	0.00
lpr-b-02	2622	2624	3600	2622	0	0.08	2624	3600	2622	2	0.08	2622	38	2622	2	0.00
lpr-b-03	6988	6935	3600	6474	14	7.12	6796	3600	6402	387	6.15	6592	3600	6402	3600	2.97
lpr-b-04	12,614	11,620	3600	10,713	124	8.47	11,184	3600	10,485	3600	6.67	11,184	3600	10,456	3600	6.96
lpr-b-05	23,259	20,018	3600	17,498	3600	14.40	19,023	3600	17,131	3600	11.04	18,998	3600	17,100	3600	11.10
lpr-c-01	842	798	35	798	0	0.00	798	1944	798	14	0.00	798	406	798	653	0.00
lpr-c-02	1922	1638	3600	1635	1	0.18	1635	1102	1635	88	0.00	1635	3600	1630	3600	0.31
lpr-c-03	6500	5503	3600	4642	3600	18.55	5106	3600	4326	3600	18.03	5106	3600	4292	3600	18.97
lpr-c-04	8569	9448	3600	6306	3600	49.83	8752	3600	6261	3600	39.79	8174	3600	6233	3600	31.14
lpr-c-05	15,008	15,938	3600	10,117	3600	57.54	13,662	3600	10,098	3600	35.29	13,063	3600	9976	3600	30.94
	Max.					57.54%					39.79%					31.14%
	Ave.					14.16%					10.92%					9.77%
Marginal location	Max.	24.36%					14.28%					6.60%				
Improvement	Ave.	4.94%						3.84%				1.24%				
Total location	Max.	24.36%					32.50%					32.88%				
Improvement	Ave.	4.94%						8.54%				9.74%				

* Calculated by Belenguer et al. (2006).

in 7th columns so that interested readers could compare the experimental results presented herein with other published results. Lower bound values are also obtained by Gouveia et al. (2010). For each instance, the lowest cost among all solutions found by different methods is highlighted by italic values. As shown by "# of dominated cases" in Table 3, the proposed Insertion heuristic dominates other heuristics in 21 cases out of 34 instances. In 7 cases, IM finds better solutions and for the other 6 instances, both heuristics lead to the same objective values. The final row of Table 2 shows that the average gap between the lowest cost found by Insertion heuristic and its lower bound (LB) is significantly less than those rendered by other heuristics. Table 3 shows similar results for *lpr* problems. The proposed heuristic finds better solutions in 10 out of 15 instances, and achieves a lower average gap.

5.3. Evaluation of the proposed SA algorithm

Computational results related to the evaluation of the proposed SA algorithms are presented in Tables 4–11. All of the obtained solutions in these tables are available in http://www.practical-optimization.com/shhd/LARP.rar. The *mval* and *lpr* instances have been solved for various settings of location cost. Since routing cost is a function of deadheading cost, to provide a tradeoff between location cost and routing cost, we have set the depot opening cost as $G = \alpha \times c_{ave}$ where α is a constant and c_{ave} is the average deadheading cost of the arcs in the corresponding network. The notation x denotes the smallest integer value greater than or equal to x. In the following tables, for each setting of location cost, different cases of maximum number of locations have been examined (L = 1, 2, 3). In addition to solution cost of location problems, all of these tables include the best known objective values available in the literature for the CARP associated with the related test problems. In Tables 4–11, the row of "Marginal Location Improvement" shows the cost saving achieved by allowing one more depot to be opened, comparing with the previous scenario using this ratio: $(SA\_Cost_{LARP(L)} - SA\_Cost_{LARP(L-1)})/SA\_Cost_{LARP(L-1)}$. In this row, the values under column $L = 1$ present maximum and average cost saving obtained by changing the location of the only available depot as compared with the solution cost of CARP. Moreover, in the last row of these tables, "Total Location Improvements" are reported which mean the total cost savings (in percentage) achieved by solving LARP when compared with CARP results using the ratio: $(SA\_Cost_{LARP(L)} - SA\_Cost_{CARP})/SA\_Cost_{CARP}$.

Tables 4–7 show computational results of solving *mval* instances. Table 4 shows that in the case of $G = 0$, the average gap between the SA solutions cost and their corresponding lower bounds is less than 6% for different cases of L = 1, 2 and 3. In the "Marginal Location Improvement" row, the cost saving under the column $L = 1$ shows that the average cost saving due to changing the location of the central depot is 13.82% for *mval* problems. Moreover, the maximum saving from location improvement happens to be 56.54%. This potential saving justifies the significance of location decision in such an application which has been the focus of this paper. In Table 4, the average and maximum value of "Total Location Improvement" over *mval* instances are 13.82% and 56.54% for $L = 1$, 19.9% and 70.07% for $L = 2$, and 22.06% and 83.20% for $L = 3$, respectively. These observations are again evidences of potential improvement by considering location decisions in this application. Similar computational results for $G = \alpha \times c_{ave}$ (α = 0.5, 1, 2) have been presented in Tables 5–7. The "Gap" row in these tables shows that as location cost increases from $G = 0.5c_{ave}$ in Table 5 to $G = 2c_{ave}$ in Table 7, the average values of Gap decrease. It can be justified by the fact that greater values of location cost cause to greater values in the denominator of the formula by which Gap values are calculated. It is also noticeable that in these tables, the average of "Total location improvements" for $L = 3$ are more than 18%. It shows that the proposed SA finds high quality solutions when nonzero location cost is assumed. The other fact that we can see in Tables 4–7 is that as the vehicle capacity decreases for a fixed instance number (for example Mval2x) from instance "a" to instances "b" and "c", the computational time of LB and SA, generally increases suddenly. An example such behavior can be in Table 4 under the column $L = 3$ for instances Mval3a, Mval3b and Mval3c. The lower bound time for these instance are 10, 20

Table 9
Evaluation of the proposed SA in solving *mval* data set for the case of $G = 0.5\lceil c_{ave} \rceil$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
lpr-a-01	1104	1104	76	1104	0	0.00	1104	152	1104	0	0.00	1104	2636	1104	0	0.00
lpr-a-02	1880	1871	3600	1794	0	4.29	1871	3600	1794	7	4.29	1871	3600	1794	1	4.29
lpr-a-03	4191	4384	3600	4022	70	9.00	4384	3600	3972	3600	10.37	4384	3600	3972	3600	10.37
lpr-a-04	7735	7073	3600	6108	323	15.80	6878	3600	5892	3600	16.73	6812	2503	5892	3600	15.61
lpr-a-05	16,274	12,482	3600	9906	3600	26.00	10,960	3600	9869	3600	11.05	10,960	3600	9869	3600	11.05
lpr-b-01	1641	1580	3	1580	1	0.00	1580	0	1580	4	0.00	1580	0	1580	1	0.00
lpr-b-02	2722	2724	3600	2722	1	0.07	2722	1034	2722	3	0.00	2722	336	2722	7	0.00
lpr-b-03	7077	7024	3600	6563	16	7.02	7024	3600	6563	901	7.02	6791	2335	6563	3600	3.47
lpr-b-04	12,704	11,710	3600	10,803	111	8.40	11,428	3600	10,665	3600	7.15	11,428	3600	10,665	3600	7.15
lpr-b-05	23,357	20,275	3600	17,755	3600	14.19	19,455	3600	17,645	3600	10.26	19,455	3600	17,645	3600	10.26
lpr-c-01	1003	959	23	959	0	0.00	959	0	959	11	0.00	959	2953	959	574	0.00
lpr-c-02	2080	1796	3600	1793	0	0.17	1796	3600	1793	99	0.17	1796	3600	1793	1209	0.17
lpr-c-03	6655	5658	3600	4797	3600	17.95	5601	3600	4636	3600	20.82	5588	3600	4636	3600	20.53
lpr-c-04	8693	9572	3600	6430	3600	48.86	8921	3600	6430	3600	38.74	8646	3600	6430	3600	34.46
lpr-c-05	15,144	16,074	3600	10,253	3600	56.77	13,936	3600	10,253	3600	35.92	13,937	3600	10,253	3600	35.93
	Max.					56.77					38.74					35.93
	Ave.					13.90					10.84					10.22
Marginal location	Max.	23.30%					13.30%					3.32%				
Improvement	Ave.	4.66%					2.84%					0.51%				
Total location	Max.	23.30%					32.65%					32.65%				
Improvement	Ave.	4.66%					7.34%					7.84%				

* Calculated by Belenguer et al. (2006).

Table 10
Evaluation of the proposed SA in solving *lpr* data set for the case of $G = \lceil c_{ave} \rceil$.

File	CARP	LARP (L = l)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
lpr-a-01	1224	1164	57	1164	0	0.00	1164	116	1164	0	0.00	1164	964	1164	0	0.00
lpr-a-02	2019	1940	3600	1863	1	4.13	1940	3600	1863	3	4.13	1940	3600	1863	1	4.13
lpr-a-03	4329	4453	3600	4091	185	8.85	4453	3600	4091	3600	8.85	4453	3600	4091	3600	8.85
lpr-a-04	7902	7156	3600	6191	333	15.59	7041	3600	6058	3600	16.23	7033	3600	6058	3600	16.09
lpr-a-05	16,436	12,715	3600	10,139	3600	25.41	11,249	3600	10,139	3600	10.95	11,249	3600	10,139	3600	10.95
lpr-b-01	1834	1676	3	1676	1	0.00	1676	0	1676	3	0.00	1676	0	1676	1	0.00
lpr-b-02	2922	2824	3600	2822	1	0.07	2822	112	2822	3	0.00	2822	89	2822	3	0.00
lpr-b-03	7254	7112	3600	6651	23	6.93	7112	3600	6651	3600	6.93	6883	3600	6651	3600	3.49
lpr-b-04	12,883	11,799	3600	10,892	117	8.33	11,600	3600	10,843	3600	6.98	11,600	3600	10,843	3600	6.98
lpr-b-05	23,553	20,531	3600	18,011	3600	13.99	19,644	3600	18,011	3600	9.07	19,391	3600	18,011	3600	7.66
lpr-c-01	1324	1119	41	1119	0	0.00	1119	11	1119	13	0.00	1119	30	1119	370	0.00
lpr-c-02	2395	1953	3600	1950	0	0.15	1953	3600	1950	90	0.15	1953	3600	1950	1612	0.15
lpr-c-03	6965	5813	3600	4952	3600	17.39	5760	3600	4946	3600	16.46	5760	3600	4946	3600	16.46
lpr-c-04	8941	9696	3600	6554	3600	47.94	8843	3600	6554	3600	34.93	8810	3600	6554	3600	34.42
lpr-c-05	15,415	16,209	3600	10,388	3600	56.04	14,614	3600	10,388	3600	40.68	14,021	3600	10,388	3600	34.97
Gap	Max.					56.04%					40.68%					34.97%
	Ave.					13.65%					10.36%					9.61%
Marginal location	Max.	22.64%					11.53%					4.06%				
Improvement	Ave.	7.34%					2.58%					0.60%				
Total location	Max.	22.64%					31.56%					31.56%				
Improvement	Ave.	7.34%					9.77%					10.34%				

* Calculated by Belenguer et al. (2006).

and 3600 s respectively. This behavior can be justified with regard to the available vehicle capacity. As discussed earlier in Section 5, the available vehicle capacity decreases from instance "a" to instances "b" and "c". This shows that as the vehicle capacity is more limited, the instance become more difficult and finding high quality solutions satisfying vehicle capacity constraint takes more time. In this case, the vehicle capacities for instances of Mval3a, Mval3b and Mval3c are set to 80, 50, and 20, respectively.

Computational result of solving *lpr* instances are presented in Tables 8–11. As shown in Table 8, in the scenario of G = 0, the average gaps for *lpr* instances with 1, 2 and 3 depots are 14.16%, 10.92% and 9.77%, respectively. In this table, the average of "Marginal Location Improvement" in the case of L = 1, 2 and 3 are 4.94%, 3.84%, 1.24%, respectively. It shows that, as it is expected, when the allowable number of locations increases, the marginal saving decreases. In Table 8, the average and maximum value of "Total

Table 11
Evaluation of the proposed SA in solving *lpr* data set for the case of $G = 2[c_{ave}]$.

File	CARP	LARP (L = 1)					LARP (L = 2)					LARP (L = 3)				
	Best known objective value*	SA		LB		Gap (%)	SA		LB		Gap (%)	SA		LB		Gap (%)
		Cost	Time	Cost	Time		Cost	Time	Cost	Time		Cost	Time	Cost	Time	
lpr-a-01	1464	1284	87	1284	0	0.00	1284	280	1284	0	0.00	1284	17	1284	0	0.00
lpr-a-02	2297	2079	3600	2002	0	3.85	2079	3600	2002	0	3.85	2079	3600	2002	1	3.85
lpr-a-03	4605	4591	3600	4229	206	8.56	4591	3600	4229	3600	8.56	4591	3600	4229	3600	8.56
lpr-a-04	8236	7323	3600	6358	348	15.18	7303	3600	6358	3600	14.86	7232	3600	6358	3600	13.75
lpr-a-05	16,760	12,573	3600	10,604	3600	18.57	11,600	3600	10,604	3600	9.39	11,600	3600	10,604	3600	9.39
lpr-b-01	2220	1869	4	1869	2	0.00	1869	0	1869	5	0.00	1869	0	1869	1	0.00
lpr-b-02	3322	3024	3600	3022	3	0.07	3024	3600	3022	9	0.07	3022	178	3022	4	0.00
lpr-b-03	7608	7289	3600	6828	63	6.75	7289	3600	6828	3600	6.75	7277	3600	6828	3600	6.58
lpr-b-04	13,241	11,978	3600	11,071	93	8.19	11,978	3600	11,071	3600	8.19	11,978	3600	11,071	3600	8.19
lpr-b-05	23,945	20,410	3600	18,524	2291	10.18	19,735	3600	18,524	3600	6.54	19,735	3600	18,524	3600	6.54
lpr-c-01	1966	1440	51	1440	7	0.00	1440	128	1440	8	0.00	1440	7	1440	12	0.00
lpr-c-02	3025	2268	3600	2265	17	0.13	2268	3600	2265	112	0.13	2268	3600	2265	314	0.13
lpr-c-03	7585	6123	3600	5262	3600	16.36	6123	3600	5262	3600	16.36	6123	3600	5262	3600	16.36
lpr-c-04	9437	9944	3600	6801	3600	46.21	9626	3600	6801	3600	41.54	9626	3600	6801	3600	41.54
lpr-c-05	15,957	16,480	3600	10,659	3600	54.61	14,800	3600	10,659	3600	38.85	14,800	3600	10,659	3600	38.85
Gap	Max.					54.61%					41.54%					41.54%
	Ave.					12.58%					10.34%					10.25%
Marginal location Improvement	Max.	26.75%				10.19%						0.97%				
	Ave.	11.59%				1.65%						0.08%				
Total location Improvement	Max.	26.75%				30.79%						30.79%				
	Ave.	11.59%				13.11%						13.18%				

* Calculated by Belenguer et al. (2006).

Location Improvement" over *lpr* instances are 8.54% and 32.50% for $L = 2$, and 9.74% and 32.88% for $L = 3$, respectively. These observations show that the proposed SA algorithm works well in solving large instances. Computational results of solving *lpr* set with non-zero location cost are presented in Tables 9–11. In all of these tables, the average values of gap for scenarios of $L = 1,2$ and 3 are less than 14%. Furthermore, the average amount of "Total location improvements" for $L = 3$ are more than 7% in these tables.

6. Conclusion

In this paper, the mathematical formulation for a multi-depot location arc routing problem is developed. A relaxation of this model over the set of tours has been solved to get a lower bound for the objective function. Furthermore, a new insertion algorithm and a location–allocation heuristic have been developed within a simulated annealing framework to solve benchmark problems. Extensive computational results show that the proposed algorithm finds good quality solutions for multi-depot LARP in reasonable time. It is also shown that the potential cost saving resulting from adding location decisions to the capacitated arc routing problem is significant.

References

Aksen, D., Altinkemer, K., 2008. A location–routing problem for the conversion to the "click-and-mortar" retailing: the static case. European Journal of Operational Research 186 (2), 554–575.

Albareda-Sambola, M., Fernandez, E., Laporte, G., 2007. Heuristic and lower bound for a stochastic location–routing problem. European Journal of Operational Research 179 (3), 940–955.

Almeida, M.T., Mourao, M.C., 2000. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. European Journal of Operational Research 121 (2), 420–434.

Amado, L., Mourao, M.C., 2005. Heuristic method for a mixed capacitated arc routing problem: a refuse collection application. European Journal of Operational Research 160 (1), 139–153.

Assad, A.A., Golden, B.L., 1995. Arc routing methods and applications. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8. North-Holland, Amsterdam, pp. 375–483.

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

Bartolini, E., Cordeau, J.F., Laporte, G., in press. Improved lower bounds and exact algorithms for the capacitated arc routing problem. Journal of Mathematical Programming 1–44.

Belenguer, J.M., Benavent, E., 1998. The capacitated arc routing problem: valid inequalities and facets. Computational Optimization and Applications 10 (2), 165–187.

Belenguer, J.M., Benavent, E., 2003. A cutting plane algorithm for the capacitated arc routing problem. Computers and Operations Research 30 (5), 705–728.

Belenguer, J.M., Benavent, E., Labadi, N., Prins, C., Reghioui, M., 2010. Split-delivery capacitated arc-routing problem: lower bound and metaheuristic. Transportation Science 44 (2), 206–220.

Belenguer, J.M., Benavent, E., Lacomme, P., Prins, C., 2006. Lower and upper bounds for the mixed capacitated arc routing problem. Computers and Operations Research 33 (12), 3363–3383.

Benavent, E., Campos, V., Corberan, A., Mota, E., 1992. The capacitated arc routing problem: lower bounds. Networks 22 (7), 669–690.

Beullens, P., Muyldermans, L., Cattrysse, D., Van Oudheusden, D., 2003. A guided local search heuristic for the capacitated arc routing problem. European Journal of Operational Research 147 (3), 629–643.

Chapleau, L., Ferland, J.A., Lapalme, G., 1984. A parallel insert method for the capacitated arc routing problem. Operations Research Letters 3 (2), 95–99.

De Rosa, B., Ghiani, G., Importa, R., Musmanno, R., 2002. The arc routing and scheduling problem with transshipment. Transportation Science 36 (3), 301–313.

Feillet, D., Dejax, P., Gendreau, M., 2005. Profitable arc tour problem: solution with a branch-and-price algorithm. Transportation Science 39 (4), 539–552.

Ghiani, G., Guerriero, F., Improta, G., Musmanno, R., 2005. Waste collection in southern Italy: solution of a real-life arc routing problem. International Transactions in Operational Research 12 (2), 135–144.

Ghiani, G., Laporte, G., 1999. Eulerian location problems. Networks 34 (4), 291–302.

Ghiani, G., Laporte, G., 2001. Location-arc routing problems. Opsearch 38 (2), 151–159.

Golden, B.L., Wong, R.T., 1981. Capacitated arc routing problems. Networks 11 (3), 305–315.

Gouveia, L., Mourao, M.C., Pinto, L.S., 2010. Lower bounds for the mixed capacitated arc routing problem. Computers and Operations Research 37 (4), 692–699.

Haastrup, P., Maniezzo, V., Mattarelli, M., Mazzeo Rinaldi, F., Mendes, I., Paruccini, M., 1998. A decision support system for urban waste management. European Journal of Operational Research 109 (2), 330–341.

Laporte, G., 1988. Location–routing problems. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 163–198.

Laporte, G., Musmanno, R., Vocaturo, F., 2010. An adaptive large neighborhood search heuristic for the capacitated arc-routing problem with stochastic demands. Transportation Science 44 (1), 125–135.
Levy, L., Bodin, L., 1989. The arc oriented location routing problem. INFOR 27 (1), 73–95.
Liu, T., Jiang, Z., Chen, F., Liu, R., Liu, S., 2008.Combined location-arc routing problems: a survey and suggestions for future research. In: IEEE International Conference on Service Operations and Logistics, and Informatics. IEEE, Washington, DC, pp. 2336–2341.
Mansini, R., GraziaSperanza, G., 1998. A linear programming model for the separate refuse collection service. Computers and Operations Research 25 (7), 659–673.
Muyldermans, L., 2003. Routing, districting and location for arc traversal problems. A Quarterly Journal of Operations Research 1 (2), 169–172.
Nagy, G., Salhi, S., 2007. Location–routing: issues, models and methods. European Journal of Operational Research 177 (2), 649–672.
Pearn, W.L., 1991. Augment-insert algorithm for the capacitated arc routing problem. Computers and Operations Research 18 (2), 189–198.
Prins, C., Prodhon, C., Ruiz, A., Soriano, P., WolflerCalvo, R., 2007. Solving the capacitated location–routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. Transportation Science 41 (4), 470–483.
Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location–routing problem. European Journal of Operational Research 210 (2), 204–212.
Ulusoy, G., 1985. The fleet size and mix problem for capacitated arc routing. European Journal of Operational Research 22 (3), 329–337.
Wøhlk, S., 2008. An approximation algorithm for the capacitated arc routing. The Open Operational Research Journal 2, 8–12.