European Journal of Operational Research 211 (2011) 318–332

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery

Ismail Karaoglan [a], Fulya Altiparmak [b,*], Imdat Kara [c], Berna Dengiz [c]

[a] Department of Industrial Engineering, Selcuk University, Turkey
[b] Department of Industrial Engineering, Gazi University, Turkey
[c] Department of Industrial Engineering, Baskent University, Turkey

ARTICLE INFO

Article history:
Received 20 November 2009
Accepted 3 January 2011
Available online 8 January 2011

Keywords:
Logistics
Location-routing problem
Simultaneous pickup and delivery
Branch-and-cut
Simulated annealing

ABSTRACT

This paper addresses a location-routing problem with simultaneous pickup and delivery (LRPSPD) which is a general case of the location-routing problem. The LRPSPD is defined as finding locations of the depots and designing vehicle routes in such a way that pickup and delivery demands of each customer must be performed with same vehicle and the overall cost is minimized. We propose an effective branch-and-cut algorithm for solving the LRPSPD. The proposed algorithm implements several valid inequalities adapted from the literature for the problem and a local search based on simulated annealing algorithm to obtain upper bounds. Computational results, for a large number of instances derived from the literature, show that some instances with up to 88 customers and 8 potential depots can be solved in a reasonable computation time.

1. Introduction

In today's competitive environment, it is obvious that companies should make strategic and operational decisions in order to optimize and manage the processes in their supply chain more efficiently. One of the most important strategic decisions concerns the design of distribution networks since it offers great potential to reduce costs and to improve service quality. The main elements in designing a distribution network are location and routing decisions. As these elements are interdependent in many distribution networks, the overall system cost can increase if routing decisions are ignored when locating facilities (Salhi and Rand, 1989). The location-routing problem (LRP) overcomes this drawback by simultaneously dealing with location and routing decisions.

The location-routing problem (LRP) deals with determining the location of facilities and the routes of the vehicles for serving the customers under some constraints such as facility and vehicle capacities, route length, etc. to satisfy demands of all customers and to minimize total cost including routing costs, vehicle fixed costs, facility fixed costs and facility operating costs. Some of the application areas of the LRP in practice can be given as food and drink distribution, military equipment location, parcel delivery and telecommunication network design.

After the importance of the considering location and routing decisions simultaneously was stressed by Webb (1968) and Christofides and Eilon (1969), different models and solution approaches for the LRP have been proposed in the literature to formulate and solve distribution network design problems. Laporte (1988) is the first researcher who classifies the LRP models. Min et al. (1998) review the LRP literature using a hierarchical taxonomy based on the problem characteristics such as the number of depots, the presence of capacity, the form of the objective function, etc. More recently, Nagy and Salhi (2007) provide a comprehensive literature review for the LRP models, solution approaches and application areas.

The facility location problem (FLP) and vehicle routing problem (VRP) are two main components of the LRP. Since both problems belong to the class of NP-hard problems, the LRP is also NP-hard problem. Because of its complexity, some mathematical models and exact solution procedures have been developed for a small number of LRP models. Laporte and Nobert (1981, 1988), Laporte et al. (1986) propose mixed integer formulations (MIP) with two-index and exact solution algorithms for the single-depot LRP, multi-depot capacitated LRP and some asymmetric versions of the LRP, respectively. Another MIP formulation with two-index and an exact algorithm for the multi-depot LRP is developed by Laporte et al. (1983). Labbe et al. (2004) propose a branch and cut algorithm for plant cycle location problem, which

* Corresponding author. Tel.: +90 312 231 74; fax: +90 312 230 84 34.
E-mail addresses: ikaraoglan@gazi.edu.tr (I. Karaoglan), fulyaal@gazi.edu.tr (F. Altiparmak), ikara@baskent.edu.tr (I. Kara), bdengiz@baskent.edu.tr (B. Dengiz).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.01.003

is a different version of the LRP arising in telecommunication network design. A multi-echelon version of the LRP with inventory is considered in Ambrosino and Grazia ScutellaI (2005). The authors propose a MIP formulation for the problem and investigate the complexity of the problem on several test problems. Belenguer et al. (2006) develop two new MIP formulations and branch-and-cut algorithms based on these formulations for the LRP. Berger et al. (2007) consider the uncapacitated LRP with route length constraints in their study and they propose a branch-and-price algorithm to solve the problem.

Different heuristic approaches have been also proposed in the literature to solve larger LRPs. Perl and Daskin (1984, 1985), Srivastava and Benton (1990), Srivastava (1993) and Hansen et al. (1994) use classic heuristic approaches. Barreto et al. (2007) propose a cluster analysis based sequential heuristic algorithm. Meanwhile meta-heuristic approaches have been successfully implemented for the problem. Several examples for the application of meta-heuristic approaches can be given as: tabu search (Tuzun and Burke, 1999; Albareda-Sambola et al., 2005), simulated annealing (Wu et al., 2002; Yu et al., 2010), greedy randomized adaptive search procedure (GRASP) (Prins et al., 2006a; Duhamel et al., 2009), memetic algorithms (Prins et al., 2006b), variable neighborhood search algorithms, (Melechovsky and Prins, 2005) and particle swarm optimization (Marinakis and Marinaki, 2008).

All the studies cited above have considered classical VRP in the LRP, i.e. each vehicle starts from a depot, traverses through a number of customers, delivers goods to each customer and returns the same depot. However, in practice, customers can have pickup and delivery demands and they request that both demands should be met at the same time. This kind of problem is known in the literature as vehicle routing problem with simultaneous pickup and delivery (VRPSPD). A number of applications of the VRPSPD can be found in the distribution system of grocery store chains, blood banks, etc. Reverse logistics is also another area in which the planning of vehicle routes takes the form of a VRPSPD. We refer the interested readers to the papers of Berbeglia et al. (2007) and Parragh et al. (2008) for an extensive review about this problem and its variations.

In this paper, we consider LRP with simultaneous pickup and delivery (LRPSPD) which is a general case of the LRP by considering simultaneously pickup and delivery demands of each customer. The LRPSPD is also a general case for the traveling salesman location problem with pickup and delivery (TSPPD) introduced by Mosheiov (1995) in terms of the number of depots to be located and the capacity of vehicles. Finally, the LRPSPD can be considered as a special case of many to many LRP introduced by Nagy and Salhi (1998) in which several customers wish to send goods to others and flows between depots are permitted. Although the LRP has been studied extensively in the literature, the LRPSPD has received no attention from researchers so far. To the best of our knowledge, we are first to address the LRPSPD. In our previous study (Karaoglan et al., 2009a), we have proposed two MIP formulations, which are two-index node-based and flow-based formulations, for the problem and presented several polynomial-size valid inequalities adapted from literature to strengthen the formulations. Our experimental studies revealed that the flow-based formulation with valid inequalities, i.e. strong formulation, can solve most small-size LRPSPD instances up to 30 customers and 5 depots to optimality within two hours of computation time by CPLEX.

Motivated by the successful applications of branch-and-cut algorithms in solving various routing problems such as capacitated vehicle routing problem (e.g. Baldacci et al., 2004), prize collecting traveling salesman problem (e.g. Berube et al., 2009), pickup and delivery traveling salesman problem (e.g. Cordeau et al., 2009) and the LRP (e.g. Belenguer et al., 2006), in this paper we propose a branch-and-cut algorithm based on flow-based formulation with two-index to solve larger LRPSPD instances to optimality. In addition, we present a new set of valid inequalities for the problem adapted from the literature to strengthen linear programming relaxation of the formulation within the branch-and-cut algorithm and a local search based on simulated annealing algorithm to obtain upper bounds. Computational results, for a large number of instances derived from the literature, show that some instances with up to 88 customers and 8 potential depots can be solved within a reasonable computation time.

The contribution of this paper is threefold. First, the well known subtour elimination constraints, generalized large multistar inequalities and Y-capacity constraints in the VRP and LRP literature are generalized for the LRPSPD. Second, an exact algorithm based on branch and cut is proposed to solve LRPSPD instances to optimality. Finally, a heuristic approach based on simulated annealing is developed to improve initial solution and upper bounds found during the search process of the branch and cut algorithm.

The rest of this paper is organized as follows. Problem definition and mathematical formulation are given in Section 2. While Section 3 describes the valid inequalities used to obtain tight bounds, the proposed branch-and-cut algorithm is explained in Section 4. Section 5 reports computational results and conclusions follow in Section 6.

2. Problem definition and mathematical formulation

The location-routing problem with simultaneous pickup and delivery (LRPSPD) can be defined as follows: let $G=(N,A)$ be a complete directed network where $N = N_0 \cup N_C$ is a set of nodes in which N_0 and N_C represent the potential depot nodes and customers, respectively, and $A = \{(i,j):\ i,j \in N\}$ is the set of arcs. Each arc$(i,j) \in N$ has a nonnegative cost (distance) c_{ij} and triangular inequality holds (i.e., $c_{ij} + c_{jk} \geqslant c_{ik}$). A capacity CD_k and a fixed cost FD_k are associated with each potential depot $k \in N_0$. An unlimited fleet of homogeneous vehicles with capacity CV and fixed operating cost FV including the cost of acquiring the vehicles used in the routing is available to serve the customers. Each customer $i \in N_C$ has pickup (p_i) and delivery (d_i) demands, with $0 < d_i, p_i \leqslant CV$. The problem is to determine the locations of depots, the assignment of customers to opened depots and the corresponding vehicle routes with minimum total cost under following constraints:

- Each vehicle is used at most one route.
- Each customer is served by exactly one vehicle.
- Each route begins and ends at the same depot.
- The total vehicle load at any point of the route does not exceed the vehicle capacity.
- The total pickup and total delivery load of the customers assigned to a depot does not exceed the capacity of the depot.

To formulate the LRPSPD, following decision variables are used:

$$x_{ij} = \begin{cases} 1 & \textit{if a vehicle travels directly from node } i \textit{ to node } j \ (\forall i,j \in N) \\ 0 & \textit{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1 & \textit{if depot } k \textit{ is opened } (\forall k \in N_0) \\ 0 & \textit{otherwise} \end{cases}$$

$$z_{ik} = \begin{cases} 1 & \textit{if customer } i \textit{ is assigned to depot } k \ (\forall i \in N_C,\ \forall k \in N_0) \\ 0 & \textit{otherwise} \end{cases}$$

U_{ij}; demand to be delivered to customers routed after node i and transported in arc(i,j) if a vehicle travels directly from node i to node $j(\forall i,j \in N)$, otherwise 0.
V_{ij}; demand to be picked-up from customers routed up to node i (including node i) and transported in arc (i,j) if a vehicle travels directly from node i to node j $(\forall i,j \in N)$, otherwise 0.

The two-index flow-based formulation of the LRPSPD is given as follows:

$$\min \quad \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} + \sum_{k \in N_0} FD_k y_k + \sum_{k \in N_0} \sum_{i \in N_C} FV x_{ki} \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in N} x_{ij} = 1 \quad \forall i \in N_C \tag{2}$$

$$\sum_{j \in N} x_{ji} = \sum_{j \in N} x_{ij} \quad \forall i \in N \tag{3}$$

$$\sum_{j \in N} U_{ji} - \sum_{j \in N} U_{ij} = d_i \quad \forall i \in N_C \tag{4}$$

$$\sum_{j \in N} V_{ij} - \sum_{j \in N} V_{ji} = p_i \quad \forall i \in N_C \tag{5}$$

$$U_{ij} + V_{ij} \leqslant CV x_{ij} \quad \forall i,j \in N,\ i \neq j \tag{6}$$

$$\sum_{j \in N_C} U_{kj} = \sum_{j \in N_C} z_{jk} d_j \quad \forall k \in N_0 \tag{7}$$

$$\sum_{j \in N_C} U_{jk} = 0 \quad \forall k \in N_0 \tag{8}$$

$$\sum_{j \in N_C} V_{jk} = \sum_{j \in N_C} z_{jk} p_j \quad \forall k \in N_0 \tag{9}$$

$$\sum_{j \in N_C} V_{kj} = 0 \quad \forall k \in N_0 \tag{10}$$

$$U_{ij} \leqslant (CV - d_i) x_{ij} \quad \forall i \in N_C,\ \forall j \in N \tag{11}$$

$$V_{ij} \leqslant (CV - p_j) x_{ij} \quad \forall i \in N,\ \forall j \in N_C \tag{12}$$

$$U_{ij} \geqslant d_j x_{ij} \quad \forall i \in N,\ \forall j \in N_C \tag{13}$$

$$V_{ij} \geqslant p_i x_{ij} \quad \forall i \in N_C,\ \forall j \in N \tag{14}$$

$$\sum_{k \in N_0} z_{ik} = 1 \quad \forall i \in N_C \tag{15}$$

$$\sum_{i \in N_C} d_i z_{ik} \leqslant CD_k y_k \quad \forall k \in N_0 \tag{16}$$

$$\sum_{i \in N_C} p_i z_{ik} \leqslant CD_k y_k \quad \forall k \in N_0 \tag{17}$$

$$x_{ik} \leqslant z_{ik} \quad \forall i \in N_C,\ \forall k \in N_0 \tag{18}$$

$$x_{ki} \leqslant z_{ik} \quad \forall i \in N_C,\ \forall k \in N_0 \tag{19}$$

$$x_{ij} + z_{ik} + \sum_{m \in N_0, m \neq k} z_{jm} \leqslant 2 \quad \forall i,j \in N_C,\ i \neq j,\ \forall k \in N_0 \tag{20}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in N \tag{21}$$

$$z_{ik} \in \{0,1\} \quad \forall i \in N_C,\ \forall k \in N_0 \tag{22}$$

$$y_k \in \{0,1\} \quad \forall k \in N_0 \tag{23}$$

$$U_{ij}, V_{ij} \geqslant 0 \quad \forall i,j \in N \tag{24}$$

where x_{ij} is set to zero when $\max\{d_i + d_j; p_i + p_j; d_j + p_i\} > CV$, $\forall i,j \in N_C, i \neq j$. This restriction guaranties that any incompatible customer pair whose total demands are greater than the vehicle capacity is not appeared in the same route. In this formulation, objective function (1) minimizes the total system cost including transportation, depot and vehicle fixed costs. Constraints (2) and (3) are known as degree constraints. While constraints (2) ensure that each customer must be visited exactly once, constraints (3) guarantee that entering and leaving arcs to each node are equal. Constraints (4) and (5) are flow conservation constraints for delivery and pickup demands, respectively. These constraints eliminate subtour and guarantee that pickup and delivery demands are satisfied for each customer. Constraints (6) imply that total load on any arc must not exceed the vehicle capacity. While constraints (7) ensure that total delivery load dispatching from each depot equals to total delivery demand of customers which are assigned to the corresponding depot, constraints (8) guarantee that the total amount of delivery load returning to the depots must be equal to zero. Similarly, while constraints (9) ensure that total pickup load entering to each depot equals to total pickup demand of customers which are assigned to the corresponding depot, constraints (10) guarantee that the total amount of pickup load dispatching from the depots must be equal to zero. Constraints (11)–(14) are bounding constraints for additional variables. Constraints (15) ensure that each customer must be assigned to only one depot. Constraints (16) and (17) guarantee that total delivery and pickup loads on any depot must not exceed the corresponding depot capacity, respectively. Constraints (18)–(20) forbid the illegal routes i.e. the routes which do not start and end at the same depot. Finally, constraints (21)–(24) are known as integrality constraints which define the nature of the decision variables.

In the given formulation, any integer solution does not contain illegal routes because of the constraint sets (18)–(20) together with (15). The validity of these constraints can be proven by contradiction. Let us consider an infeasible route $P = \{v_k, v_1, v_2, \ldots, v_{s-1}, v_s, v_l\}$ with s arbitrary number of customers where $k, l \in N_0$ and $k \neq l$. Constraints (18) imply that $z_{v_1,k} = 1$. Since $x_{v_1,v_2} = 1$ and $z_{v_1,k} = 1$, constraints (20) ensure that $z_{v_2,k} = 1$. By introducing constraint (20) to all adjacent pairs in route P up to node v_s, it is obtained that $z_{v_s,k} = 1$ which contradict with constraints (15) and (19).

This formulation includes $O((|N_c| + |N_0|)^2)$ binary variables, $O((|N_c| + |N_0|)^2)$ additional variables and $O(|N_c|^2|N_0|)$ constraints.

3. Valid inequalities for the LRPSPD

Valid inequalities are one of the most practical ways to strengthen the linear relaxations of the formulations. These inequalities eliminate some fractional solutions from the solution space such that a stronger lower bound can be obtained for the problem. In this paper, we utilize seven valid inequalities, which were developed for the VRP and FLP in the literature, in our branch-and-cut algorithm. It is worth noting that four out of seven inequalities, which are polynomial-size inequalities, have been adapted to the LRPSPD in our previous study, (Karaoglan et al., 2009a), to strengthen the formulations. Nevertheless, in this paper, we adapt other inequalities to solve the LRPSPD for the first time. Hence, for the sake of the completeness of the paper, all of the inequalities are introduced in this section.

First simple and efficient polynomial-size valid inequality is given as follows:

$$z_{ik} \leqslant y_k \quad \forall i \in N_C,\ \forall k \in N_0 \tag{25}$$

This inequality has been used by Labbe et al. (2004) for plant cycle location problem and still valid for the LRPSPD. This imposes that customer $i \in N_C$ cannot be assigned to the depot $k \in N_0$ if depot k is not open.

Another polynomial-size valid inequality which bounds below the number of routes originating from depots is given as follows:

$$\sum_{k \in N_0} \sum_{i \in N_C} x_{ki} \geqslant r_{LRPSPD}(N_C) \tag{26}$$

where $r_{LRPSPD}(N_C) = \left\lceil \max\left(\sum_{i \in N_C} d_i; \sum_{i \in N_C} p_i\right) \Big/ CV \right\rceil$ and $\lceil \bullet \rceil$ is the smallest integer bigger than $\bullet$. Similar bounding constraint has been used by Achuthan et al. (2003) for the VRP. Validation of this inequality for the LRPSPD is given in Karaoglan et al. (2009a).

Similarly, following polynomial-size constraint which bounds below the number of opened depots is valid for the LRPSPD (Belenguer et al. (2006)):

$$\sum_{k \in N_0} y_k \geqslant y_{\min} \tag{27}$$

where y_{min} denotes the number of opened depots until satisfying $\sum_{k \in S} CD_k \geqslant \max(d(N_C); p(N_C))$ after the depots are listed in non-increasing order of their capacities.

Last polynomial-size valid inequality is given as follow:

$$x_{ij} + x_{ji} \leqslant 1 \quad \forall i, \quad j \in N_C \tag{28}$$

Constraints (28) ensure that any feasible route cannot contain subtour with only two customers. This constraint is a special case of following exponential-size constraints which are derived from capacity and subtour elimination constraints of the VRP (Laporte et al., 1983). Let S be the subset of customers N_c, i.e. $S \subseteq N_c$. The subtour elimination constraints adapted for delivery and pickup demands are given as follows:

$$\sum_{(i,j) \in S} x_{ij} \leqslant |S| - r_{LRPSPD}(S) \quad \forall S \subseteq N_C,\ S > 2 \tag{29}$$

where $r_{LRPSPD}(S)$ is calculated as in constraints (26). The constraints (29) guarantee that the number of vehicles visiting a set of customers is not less than the corresponding lower bound.

Another exponential-size inequality for the LRPSPD is based on the generalized large multistar (GLM) inequalities which have been originally proposed for the VRP (Gouveia, 1995; Letchford and Salazar-Gonzalez, 2006). We have adapted the GLM for delivery and pickup demands as follows:

$$\sum_{i \in S} \sum_{j \in N_C/S} x_{ij} \geqslant 1/CV \left(\sum_{i \in S} d_i + \sum_{i \in S} \sum_{j \in N_C/S} d_j (x_{ij} + x_{ji}) \right) \quad \forall S \subseteq N_C, \quad S \neq \varnothing \tag{30}$$

$$\sum_{i \in S} \sum_{j \in N_C/S} x_{ij} \geqslant 1/CV \left(\sum_{i \in S} p_i + \sum_{i \in S} \sum_{j \in N_C/S} p_j (x_{ij} + x_{ji}) \right) \quad \forall S \subseteq N_C, \quad S \neq \varnothing \tag{31}$$

The constraints (30) and (31) are valid inequalities for the LRPSPD and facet inducing especially when $(\sum_{i \in S} d_i / CV)$ or $(\sum_{i \in S} p_i / CV)$ are close to the next integer because constraints (29) may not be violated in this situation.

The inequalities (25)–(31) adapted to the LRPSPD are based on feasibility requirement. It means that any feasible solution must satisfy these constraints. We now consider another exponential-size inequality which eliminates some feasible solutions based on a special structure of an optimal LRPSPD solution. Similar constraints have been proposed in literature for the VRP and LRP by Achuthan et al. (2003) and Belenguer et al. (2006), respectively. This inequality is given as follows:

$$\sum_{(i,j)\in(S\cup k)} x_{ij} \leqslant |S| + y_k r^*(k,S) \quad \begin{array}{l}\forall k \in N_0,\ \forall S \subseteq N_C, |S| \geqslant 2 \\ \max\left(\sum_{i\in S} d_i;\quad \sum_{i\in S} p_i\right) \leqslant CD_k\end{array} \tag{32}$$

where $r^*(k,S)$ is the optimal number of vehicles for serving the customer set S from depot k. Constraints (32) impose that the number of routes originating from any depot and serving only customer set S must not greater than the optimal number of routes. Validity of this inequality for the LRPSPD is given in Proposition 1.

Proposition 1. $\sum_{(i,j)\in(S\cup k)} x_{ij} \leqslant |S| + y_k r^*(k,S)$ *is a valid inequality for the LRPSPD.*

Proof. Firstly, consider a subproblem including a depot k and serving only two disjoint subsets S_1 and S_2 of customers with two different routes, and merging these two routes is feasible in terms of vehicle load, i.e. total load of vehicle on any arc of route is not greater than vehicle capacity. Given that these conditions and the triangular inequality hold, following inequality is valid for the LRPSPD;

$$\sum_{(i,j)\in(S\cup k)} x_{ij} \leqslant |S| + 1 \quad \begin{array}{l}\forall k \in N_0,\ \forall S \subseteq N_C,\ |S| \geqslant 2 \\ \max\left(\sum_{i\in S} d_i; \sum_{i\in S} p_i\right) \leqslant \min(CV, CD_k)\end{array} \tag{33}$$

where $S = S_1 \cup S_2$. Constraints (33) impose that two routes originating from depot k and satisfying the conditions given above can be merged into a single route. This procedure produces a new route in which total cost is not greater than the previous one. At this point, satisfying the feasibility in terms of vehicle load is the key factor. In the case of classical demand structure (i.e. there are only delivery or pickup demands), $\sum_{i\in S} d_i \leqslant \min(CV, CD_k)$ is necessary and sufficient condition for merging. However, in the case of simultaneous pickup and delivery demands, modified version of this condition (i.e., $\max\left(\sum_{i\in S} d_i,\ \sum_{i\in S} p_i\right) \leqslant \min(CV, CD_k)$) is necessary but not sufficient condition because of the load fluctuation at the customers. Therefore, constraints (33) are modified for the simultaneous pickup and delivery demands as follows:

$$\sum_{(i,j)\in(S\cup k)} x_{ij} \leqslant |S| + r^*(k,S) \quad \begin{array}{l}\forall k \in N_0,\ \forall S_1, S_2 \subseteq N_C, |S| \geqslant 2 \\ \max\left(\sum_{i\in S} d_i; \sum_{i\in S} p_i\right) \leqslant \min(CV, CD_k)\end{array} \tag{34}$$

The constraints (34) can be generalized to more than two routes by relaxing the necessary condition as $\max\left(\sum_{i\in S} d_i; \sum_{i\in S} p_i\right) \leqslant CD_k$. Finally, in order to lift the constraints we obtain constraints (32) by replacing the right hand side of constraints (34) with $|S| + y_k r^*(k,S)$, which are valid for the LRPSP. □

Since there is only one depot and subset of all customers with delivery and pickup demand in constraints (32), the problem under consideration reduces to VRPSPD. Thus, $r^*(k,S)$ is obtained by solving the VRPSPD to optimality. Although the VRPSPD is an NP-hard problem, optimal solution can be obtained in a very short computation time for the small-size instances by using efficient MIP models. In this paper, we implement a flow-based MIP model proposed by Karaoglan et al. (2009b) for the VRPSPD.

4. Branch and cut algorithm

In this section, we describe a branch-and-cut algorithm for the exact solution of the LRPSPD. A branch-and-cut algorithm implements a combination of cutting planes and implicit enumeration to solve any combinatorial optimization problem. The basic idea is based on the identification of the violated inequalities that are valid throughout the enumeration tree. Thus, at each step of the algorithm, violated inequalities, which are identified by solving the separation problem, are added to the formulation and the corresponding linear program is reoptimized.

Fig. 1 presents the steps of the branch-and-cut algorithm developed for the LRPSPD, where $LP(P)$ is the LP-relaxation of model P, S_t^* is the optimal solution of $LP(P)$ at the particular node t, S_{best} is the best feasible solution for the LRPSPD, $f(S_t)$ is the objective function value of the solution at node t, S_{init} is the initial feasible solution, Φ is the set of unexplored nodes of enumeration tree, ns is the number of nodes selected from Φ, s_{max} is the maximum number of the selected nodes.

The main steps of the algorithm can be summarized as follows: *Step 1* obtains an initial solution (S_{init}) and an initial LP, and also initialize the set of unexplored nodes of enumeration tree (Φ). *Step 2* specifies the termination criterion and selects a node with the smallest objective function value from enumeration tree for additional processing. *Step 3* solves the LP-relaxation of the model P and obtains S_t^*. *Steps 4 and 5* prune the current node (t) if S_t^* is infeasible or its objective function value is worse than that of the best feasible solution (S_{best}), respectively. *Step 6* prunes the current node (t) after S_{best} is updated if it is possible. If S_t^* is not integer and the number of selected nodes (ns) from Φ reaches to a predetermined value (s_{max}), a new integer and feasible solution (S_t^{feas}) is obtained from S_t^* by using a special procedure in *Step 7*. If a tailing off (i.e. an improvement on the best solution is not greater than a predetermined value in the last successive pt iterations at the same branching node) does not exist, separation problem is solved for S_t^* in *Step 8*. *Step 9* adds the violated inequalities to the model and the LP relaxation is reoptimized in *Step 3*. *Step 10* creates two new nodes by applying branching rule, if a tailing off exists or new violated inequalities are not identified for S_t^*. The subsequent sections discuss the main building blocks of the branch-and-cut algorithm developed for the LRPSPD in detail.

4.1. Initial solution

At the root node of the branching tree, an initial solution is obtained. The cost of this solution gives an upper bound of the optimal solution value. The procedure to obtain an initial solution in our algorithm can be summarized as follows: after an initial feasible solution is generated by a greedy heuristic, it is improved by SA algorithm and the improved solution is considered as an initial solution to branch-and-cut algorithm.

Algorithm: Branch-and-Cut for the LRPSPD
Input: LRPSPD problem data
Output: Optimal solution (S_{best}) for the LRPSPD

Step 1. Obtain S_{init} and $f(S_{init})$ by using InitFeasSol procedure and SA algorithm, obtain initial $LP(P)$, $S_{best} \leftarrow S_{init}$, $f(S_{best}) \leftarrow f(S_{init})$, $ns \leftarrow 0$ and add the root node to Φ and set the root node as current node (t)
Step 2. If ($\Phi = \varnothing$) or (termination criterion is met) then report S_{best} and stop
else select enumeration node $t \in \Phi$ and $ns \leftarrow ns+1$
Step 3. Solve relaxation of corresponding $LP(P)$ and obtain S_t^*
Step 4. If S_t^* is infeasible then $\Phi \leftarrow \Phi \setminus t$ and goto Step 2
Step 5. If $f(S_t^*) > f(S_{best})$ then $\Phi \leftarrow \Phi \setminus t$ and goto Step 2
Step 6. If (S_t^* is integer) and ($f(S_t^*) < f(S_{best})$) then
$S_{best} \leftarrow S_t^*$, $f(S_{best}) \leftarrow f(S_t^*)$, $\Phi \leftarrow \Phi \setminus t$ and goto Step 2
Step 7. If (S_t^* is not integer) and ($ns = s_{max}$) then $ns \leftarrow 0$ and
obtain S_t^{feas} by using FeasSol procedure and SA algorithm
if $f(S_t^{feas}) < f(S_{best})$ then $S_{best} \leftarrow S_t^{feas}$, $f(S_{best}) \leftarrow f(S_t^{feas})$
Step 8. If a tailing off exists then goto Step 10
else solve separation problem to identify violated inequalities and put them set κ
Step 9. If $\kappa \neq \varnothing$ then add violated inequalities to the $LP(P)$ and goto Step 3
Step 10. Select a non-integer decision variable according to branching rule, create two new nodes and add them to Φ and goto Step 2

Fig. 1. The branch-and-cut algorithm for the LRPSPD.

4.1.1. Generating initial feasible solution

In order to obtain an initial feasible solution, we implement Extended Clarke and Wright Algorithm (ECWA), which is a greedy heuristic. The ECWA is an extension of the well-known Clarke and Wright heuristic (Clarke and Wright, 1964) and firstly introduced by Prins et al. (2006b) for the LRP. The implementation of this heuristic to the LRPSPD can be summarized as follows: initially, a penalty cost (pc_i) arising when the customer is assigned to the second closest depot instead of the closest one is calculated for each customer, and the customers are listed in non-increasing order of their penalty costs. Then, each customer is assigned to the closest depot starting from top of the list. If some of the customers are not assigned to their closest depots because of the depots' capacity, their penalty costs are recalculated considering closest ones with enough capacity and the assignment of these customers to appropriate depots is tried again. These steps are repeated until all customers are assigned. After that depots without customers are closed and a simple route for each customer from whose depot is constructed. Then, each pair of routes (R_1,R_2) in the feasible solution are examined in terms of cost saving obtained by their combination and a pair of nodes providing largest cost saving is combined. This strategy is repeated until no capacity-feasible combination is found.

4.1.2. Improving the initial feasible solution with SA

A straightforward SA is implemented to obtain an initial solution (S_{init}) for the branch-and-cut algorithm. SA, which stems from the simulation of the annealing of solids, is a stochastic search technique that is able to escape local optima using a probability function (Suman and Kumar, 2006). It starts with an initial feasible solution (S'_{init}) and improves it until stopping criterion is met. We employ following four moving strategies, which are well-known in the VRP literature and also extended to the LRP by Prins et al. (2006b), to define neighborhoods for the SA:

- *Insert:* One customer is inserted to a new position, which is in the same route or in two different routes belonging to same depot or two different depots, from its current position.
- *Swap*: Two customers, which are in the same route or in two different routes belonging to same depot or two different depots, are exchanged.
- *Opt*: Two non-consecutive arcs, which are in the same route or in two different routes belonging to same depot or different depots, are deleted. If the deleted arcs are in the same route, two new arcs are created and the path lying between the created arc pair is reversed. If the deleted arcs are in two different routes of the same depot, each route is divided into two parts as starting and terminating part. Then two new arcs are created in such a way that starting and terminating parts of two different routes are connected. Otherwise, i.e. the deleted arcs belong to two different routes of different depots, after two different parts coming from different routes are connected by two new arcs, they are revised such that each route starts and finishes at the same depot.
- *Merge*: Two routes, R_1 and R_2, belonging to same or different depots are selected and merged into new one, R, considering following four alternatives: route R is assigned to the depot r_1 of R_1, to the depot r_2 of R_2, or to another depot r, opened or not. Hence, after four alternatives are evaluated in each merge operation, the one with largest cost saving is applied.

At each iteration of the SA, neighbors of the current solution are generated by using all of the moving strategies and the best one among the neighbors is chosen as a new solution (S_{new}) for the problem. The SA algorithm implements a candidate list strategy in generating the neighbors since searching whole neighborhood of the current solution by a moving strategy is a very time consuming process. According to this strategy, each moving strategy generates randomly a subset of neighbors, LS, that satisfying vehicle and depot capacity constraints and they are gathered in a pool to select a new solution (S_{new}). If the new solution is better than current solution then it is accepted as the current solution, otherwise it is accepted with probability of $\exp(-\Delta s/T_{itr})$ as the current solution. Δs is a relative percent deviation of quality of the new solution from the current solution and calculated by $[(f(S_{new}) - f(S_{cur}))/f(S_{cur})] * 100$. In each iteration of SA, temperature (T_{itr}) is reduced using a geometric cooling schedule, i.e. $T_{itr} = \alpha T_{itr-1}$ where α is a cooling rate. SA stops when the temperature reaches to final temperature.

4.2. Initial linear program

To initialize the LP model, we relax constraints (11)–(14) and (18)–(20), which are having at least $O(|N_c||N_0|)$ complexity, in addition to integrality constraints of the original formulation. The rationale behind the relaxation of constraints (11)–(14) and (18)–(20) is that their relaxation reduces the size of the LP model and thus helps to solve the model efficiently. Hence, the initial LP model includes constraints (2)–(10), (15)–(17), (24) together with the valid inequalities (26) and (27), which are bounding below the number of routes originating from depots and below the number of opened depots at the root node.

4.3. Separation procedures

We utilize mainly two different separation procedures in order to identify violated inequalities. While polynomial-size inequalities, namely (11)–(14), (18)–(20), (25) and (28), are separated by a straightforward way, we propose greedy constructive heuristics in separation of exponential-size inequalities (29)–(32).

The separation of constraints (11) is carried out by considering each arc (i,j) with $x^*_{ij} > 0$ $i \in N_C, j \in N$ and evaluating the constraint. Other polynomial-size constraints are also separated in a similar way after determining the corresponding arcs for each constraint. Note that, in the separation of the constraints (25), depots with $y_k > 0$ should be considered in addition to corresponding arcs, i.e. (i,k), $i \in N_C$, $k \in N_0$ and $x_{ik} > 0$.

We separate the constraints (29) by means of a greedy constructive heuristic. In each iteration of the heuristic, a node from the set of customer nodes, Ω, is considered as a seed, s. After the set Γ is initialized by this seed, i.e. $\Gamma \leftarrow s$, it is iteratively expanded by a new customer node, t^*, that minimizing the slack of the constraint, and then the constraint violation is checked for $\Gamma \cup t^*$. If the constraint is violated, then it is stored into a set, Θ, consisting of the constraints to be added to the model. When the current set Γ is not expanded without generating a previously generated set, the next seed from Ω is considered and these steps are repeated until $\Omega = \emptyset$. Fig. 2 presents the steps of the heuristic used for the separation of the constraint (29). It is worthy to note that set Ω can be initialized considering all customer nodes in N_c. However, our preliminary experiments show that there is no difference in the set Θ in terms of included constraints. Therefore, in order to eliminate unnecessary computations, Ω is initialized with the half of the customer nodes which are selected randomly.

We also implement the same heuristic given in the Fig. 2 in the separations of the constraints (30) and (31) after changing the definition used in the selection of node t^* with minimum slack value. Following definitions (35) and (36) are used to select node t^* in Step 2.3.1 for the constraints (30) and (31), respectively.

$$t^* \leftarrow \underset{t\in(N_c\setminus\Gamma)}{\arg\min}\left\{slack_t^{30}\,\middle|\,\frac{1}{CV}\left(\sum_{i\in(\Gamma\cup t)} d_i + \sum_{i\in(\Gamma\cup t)}\sum_{j\in(N_c\setminus(\Gamma\cup t))} d_j\left(x^*_{ij}+x^*_{ji}\right)\right) - \sum_{i\in(\Gamma\cup t)}\sum_{j\in(N_c\setminus(\Gamma\cup t))} x^*_{ij}\right\} \tag{35}$$

$$t^* \leftarrow \underset{t\in(N_c\setminus\Gamma)}{\arg\min}\left\{slack_t^{31}\,\middle|\,\frac{1}{CV}\left(\sum_{i\in(\Gamma\cup t)} p_i + \sum_{i\in(\Gamma\cup t)}\sum_{j\in(N_c\setminus(\Gamma\cup t))} p_j(x^*_{ij}+x^*_{ji})\right) - \sum_{i\in(\Gamma\cup t)}\sum_{j\in(N_c\setminus(\Gamma\cup t))} x^*_{ij}\right\} \tag{36}$$

The greedy constructive heuristic used in the separation of constraint (32) differs from the heuristic given in Fig. 2 in terms of initializing the set Γ and calculating the slack value at the selected node t^*. In this approach, after set Γ is initialized with depot k and customer s with $x^*_{ks} > 0$, Γ is successively updated by adding a new customer node t^*considering approximated slack value, which is calculated using the lower bound of the number of optimal routes (r_{LRPSPD} $(\Gamma \cup t)$) instead of $r^*(k, \Gamma \cup t)$. The rationale behind using approximated slack value is that the calculation of ($r_{LRPSPD}(\Gamma \cup t)$) is much easier than that of $r^*(k, \Gamma \cup t)$ such that it helps the reducing computational burden of the branch-and-cut algorithm. After a customer node t^* is selected among candidate nodes, its exact slack value based on $r^*(k, \Gamma \cup t^*)$ is used during the checking the violation of the constraint (32) in order to maintain the optimality of LRPSPD solution. As one depot k and a subset of customers assigned to this depot are considered in each iteration of the greedy heuristic, the problem under consideration is reduced to

Procedure: Seperation of constrainst (29)
Input: Fractional solution of $LP(P)$
Output: Violated constraints
Step 1. Select half of the customers randomly and add them to set Ω and $\Theta \leftarrow \emptyset$
Step 2. Perform following steps until $\Omega = \emptyset$
Step 2.1. Select randomly a customer s from Ω
Step 2.2. $\Gamma \leftarrow s$; $\Omega \leftarrow \Omega \setminus s$
Step 2.3. Perform following steps while $\left(\sum_{i\in\Gamma}\sum_{j\in N_c\setminus\Gamma} x^*_{ij} > 0\right)$
Step 2.3.1. $t^* \leftarrow \underset{t\in(N_c\setminus\Gamma)}{\arg\min}\left\{slack_t^{29}\,\middle|\,|\Gamma\cup t| - r_{LRPSPD}(\Gamma\cup t) - \sum_{(i,j)\in(\Gamma\cup t)} x^*_{ij}\right\}$
Step 2.3.2. $\Gamma \leftarrow \Gamma \cup t^*$ and check the violation of constraint for the set Γ
Step 2.3.3. If violated constraint has been found, then store it into set Θ

Fig. 2. Separation algorithm for the constraint (29).

the VRPSPD. Therefore, we implement a flow-based MIP model, proposed by Karaoglan et al. (2009b) for the VRPSPD, to find $r^*(k, \Gamma \cup t^*)$. Since the VRPSPD is an NP-hard problem, we restrict the number of nodes to be examined in set Γ by $N_{\max}$to obtain an optimal solution in a very short computation time. After our preliminary experiments, $N_{\max}$ is set to 10.

The separation approaches are applied sequentially until one succeeds in our branch-and-cut algorithm. Based on our preliminary experiments, the order in which the separation approaches are executed is determined as follows: we first call the separation approaches for the polynomial-size inequalities, i.e. (11)–(14) and (18)–(20), (25) and (28) to generate violated cuts and these cuts are imposed into the LP model one at a time. If violated cuts for polynomial-size inequalities are not found, the heuristic procedures for the exponential-size inequalities (29)–(32) are called sequentially until reaching a set of violated cuts. After violated cuts are identified and imposed to the LP model, the model is reoptimized.

The objective function value of the LP model is not always improved by adding violated cuts to the LP model. This phenomenon is called tailing off and it can be reduced by forcing the algorithm to branch instead of generating new violated cuts. The branching in our algorithm is forced whenever the improvement in the objective function value of the LP model is not greater than 0.01 in the last five iterations at the same branching node.

4.4. Upper bound

An effective upper bounding procedure can speed up the branch-and-cut algorithm by reducing the number of explored nodes. Thus, we implement a three-stage heuristic approach to generate feasible solutions from fractional solutions. While first stage decides which depots will be opened, next two stages assign each customer to one of the opened depots and construct routes in each depot, respectively. Let $x^*_{ij},\ y^*_k,\ z^*_{ik}$ be optimal fractional solution of any node on enumeration tree. In the first stage, depots are sorted in non-increasing order of y^*_k and then first $y_{\min}$ depots satisfying total delivery and total pickup demands of customers are opened. Second stage selects largest z^*_{ik} value and assigns customer i^* to opened depot k^* if its capacity is sufficient. This assignment process continues until all customers are assigned to opened depots. If there are some customers who are not assigned to any opened depot, new depots selected among closed ones considering y^*_k are opened and those customers are assigned to opened depots considering z^*_{ik}. In the final stage, feasible routes are constructed for each opened depot k using following procedure: firstly, a route R is started by selecting customer i^* with largest x^*_{ki} and then appending a new customer j^* with largest x^*_{ij} to the last customer i^* in the route R continues as long as their total pickup and delivery loads do not exceed vehicle capacity. This step is repeated until all customers are assigned to routes in the depot k. Note that a simple route is constructed for customer j who is not assigned to any routes in the depot k since $x^*_{ij} = 0$. After generating a feasible solution (S'_{feas}), an improved solution (S_{feas}) is obtained by implementing SA algorithm explained in Section 4.1. The main steps of the heuristic approach are given in Fig. 3 where Ω is the set of opened depots and Ψ_k is the set of customers assigned to depot k. To reduce the computation time spent by the upper bounding procedure, it is applied after ten consecutive nodes are evaluated.

4.5. Branching strategy

If the separation algorithms tail off or the solution is not integer, branching is performed. We implement straightforward branching strategy based on variables. Since there are three different binary decision variables in the formulation, we implement following order in branching stage: firstly the algorithm tries to branch on the most fractional y_k variable which is closer to 0.5 and generate two subproblems in such a way that one by fixing $y_k = 1$ and the other by fixing $y_k = 0$. If all the y_k variables are integer, the algorithm branches on the most fractional z_{ik} variable. If all y_k and z_{ik}variables are integer, the most fractional x_{ij} variable is used in branching.

5. Computational study

This section presents the results from our computational experiments which were organized into two stages. The first stage investigates the effects of valid inequalities in the proposed formulation. The second stage evaluates the performance of the proposed branch-and-cut algorithm. Before the computation results, firstly we give a brief information about the test problems.

5.1. Test problems

Since there are no benchmark instances in the literature for the LRPSPD, we have derived LRPSPD instances from two LRP test sets generated by Prodhon (2008) and Barreto (2003) using demand separation approaches proposed by Salhi and Nagy (1999) and Angelelli and Mansini (2002). In both sets, coordinates are given for each customer and the cost between two customers is the Euclidean distance rounded to the real number with four digits.

Barreto's set includes 20 test instances which were taken from the LRP literature or obtained by adding depots to classical VRP instances. While the number of customers varies between 8 and 318, the number of depots is between 2 and 15. We consider the names of original LRP and VRP instances to denote an instance from this set in sequel. The name of each instance includes the information about the name of the author, who generated the instance, the publication year of the instance, the number of customers, $|N_C|$, and the number of potential depots, $|N_0|$ (i.e. *Author-Year-*$|N_C|$x$|N_0|$).

Prodhon's set consists of 28 LRP instances with capacitated routes and depots. The main characteristics of the set are given as follows: the number of customers $|N_c|$ in $\{20, 50, 100, 200\}$, the number of potential depots $|N_0|$ in $\{5, 10\}$, uniform demands in [11,20], the number of clusters *clu* in $\{1, 2, 3\}$ (1 means that all nodes scatter on Euclidean plane), vehicle capacity CV in {a, b} where $a = 70$ and $b = 150$. In the set, depot capacities have been determined in such way that at least two or three depots are opened. In rest of the paper, we utilize following coding structure to denote an instance from this set: $|N_C| - |N_0| - cluCV$.

We consider test instances up to 100 customers of both test sets (i.e. first 15 and 22 instances from Barreto's and Prodhon's test sets, respectively) in our computational study. In order to generate the delivery and pickup demands of the customers in each test instance, we

Procedure: FeasSol (obtaining a feasible solution at the node of enumeration tree)

Input: $x_{ij}^*, y_k^*, z_{ik}^*$: optimal fractional solution $(\forall i, j \in N, \forall k \in N_0)$;

Output: S'_{feas} : feasible solution

Step 1. Set $\Omega \leftarrow \varnothing$, $\Psi_k \leftarrow \varnothing$, $S'_{feas} \leftarrow \varnothing$

Step 2. Sort the depots according to non-increasing order of y_k^*, open first y_{min} depots and put opened depots to set Ω.

$$y_{min} \leftarrow \min_{S \subseteq N_0} \{|S| \,|\, \textstyle\sum_{k \in S} CD_k \ge max(\sum_{i \in N_C} d_i; \sum_{i \in N_C} p_i)\}$$

Step 3. Perform following steps while $(N_C \setminus \bigcup_{k \in \Omega} \Psi_k) \neq \varnothing$ (assigning customers to one of the opened feasible depot)

Step 3.1. $i^* \leftarrow \infty$

Step 3.2. $i^*, k^* \leftarrow \underset{k \in \Omega, i \in N_C}{\text{argmax}} \left(z_{ik}^* \,\middle|\, i \notin \bigcup_{k \in \Omega} \Psi_k \text{ and } \sum_{j \in (\Psi_k \cup i)} d_j < CD_k \text{ and } \sum_{j \in (\Psi_k \cup i)} p_j < CD_k \right)$

Step 3.3. if $i^* = \infty$ then open depot k among closed ones with largest y_k^*,

$y_{min} \leftarrow y_{min} + 1$, $\Omega \leftarrow \Omega \cup k$ and goto Step 3.2 else $\Psi_{k^*} \leftarrow \Psi_{k^*} \cup i^*$

Step 4. Perform following steps for constructing routes to service all customers for each depot k

Step 4.1. $R \leftarrow \varnothing$

Step 4.2. $i^* \leftarrow arg\,max_{i \in \Psi_k} (x_{ki}^*)$, $R \leftarrow R \cup i^*$ and $x_{ki^*}^* \leftarrow 1$

Step 4.3. $j^* \leftarrow \infty$

Step 4.4. $j^* \leftarrow arg\,max \left\{ x_{i^* j}^* \,\middle|\, \sum_{m \in R \cup j} d_m < CV, \sum_{m \in R \cup j} p_m < CV \text{ and } \sum_{m \in R} p_m + d_m < CV \right\}$

Step 4.5. if $j^* < \infty$ then $R \leftarrow R \cup j^*$, $x_{i^* j^*}^* \leftarrow 1$, $i^* \leftarrow j^*$, $\Psi_{k^*} \leftarrow \Psi_{k^*} \setminus j^*$ and

goto Step 4.3. else $x_{i^* k}^* \leftarrow 1$, $S'_{feas} \leftarrow S'_{feas} \cup R$ and goto Step 4.1

Fig. 3. The algorithm to generate a feasible solution from the fractional LRPSPD solution.

utilize demand separation approaches proposed by Salhi and Nagy (1999) and Angelelli and Mansini (2002). These approaches are briefly defined as follows. In Salhi and Nagy's approach, a ratio $r_i = \min(x_i/y_i; y_i/x_i)$, where x_i and y_i are the coordinates of customer i, is calculated for each customer i, and then the delivery and pickup demands are generated as $d_i = r_i * q_i$ and$p_i = q_i - d_i$, where q_i is the original demand of customer i. We refer to this type of problems as X. Similarly, another type of problem, called Y, is generated by exchanging delivery and pickup demands of each customer. In Angelelli and Mansini's approach, the original demand of each customer i is considered as delivery demand ($d_i = q_i$) and the pickup demand of the corresponding customer is generated as $p_i = \lfloor(1-\gamma)q_i\rfloor$ if i is even and $p_i = \lfloor(1+\gamma)q_i\rfloor$ if i is odd. In this paper, we consider two γ values as 0.2 and 0.8 to generate two different types of problems called Z and W, respectively. As a result, the number of LRPSPD instances generated from Barreto's and Prodhon's test sets by using 4 different separation strategies (X,Y,Z and W type) are equal to 60 and 88 instances, respectively. The interested readers can reach the LRPSPD instances from authors.

5.2. Effects of valid inequalities

Following Cordeau et al. (2009) study, we conduct two experiments to investigate the effect of valid inequalities on LP relaxation of problem. In the experimental study, we utilize test instances derived from Barreto's set.

The first experiment is to show that adding one family of valid inequalities at a time to original formulation improves the LP relaxation bounds. We analyze computational results using lower bound percentage gap for each test instance. The lower bound percentage gap, ΔLB,

Table 1
Effect of valid inequalities added to original formulation.

Instances[a]	Average percentage gap ($\overline{\Delta LB}$)							
	Original	(25)	(26)	(27)	(28)	(29)	(30), (31)	(32)
Srivastava86-8×2	15.71	11.88	15.33	14.09	14.39	8.73	15.71	12.12
Perl83-12×2	20.71	10.55	15.89	10.55	20.15	16.13	20.71	20.66
Gaskell67-21×5	24.43	22.69	22.56	16.06	20.20	13.89	23.79	24.36
Gaskell67-22×5	23.36	17.85	22.98	20.50	15.77	11.89	23.11	23.03
Min92-27×5	35.56	32.30	35.43	33.73	25.34	18.03	35.51	34.02
Gaskell67-29×5	30.07	23.73	29.11	26.84	25.58	17.94	30.06	29.66
Gaskell67-32×5_1	24.43	19.94	22.68	22.72	18.17	11.95	24.43	24.36
Gaskell67-32×5_2	23.28	18.66	21.32	20.91	14.80	10.04	22.67	23.19
Gaskell67-36×5	18.04	7.12	16.19	8.85	18.04	17.16	18.04	18.04
Ch69-50×5	24.49	15.10	23.66	18.69	21.59	20.23	24.49	24.29
Perl83-55×15	25.66	23.57	24.46	18.12	24.92	24.10	25.66	25.54
Christofides69-75×10	28.34	19.47	27.26	24.58	26.28	24.27	28.15	27.91
Perl83-85×7	27.55	26.80	27.30	17.86	26.51	25.58	27.55	27.54
Daskin95-88×8	29.76	31.04	29.48	24.98	34.86	32.13	30.01	29.76
Christofides69-100×10	23.76	16.92	22.61	20.08	20.60	19.26	23.72	23.69
Average	25.01	19.84	23.75	19.90	21.81	18.09	24.91	24.54

[a] Instances are derived from Barreto's set.

is the gap between the LP relaxation bound (LB) produced by a particular formulation and upper bound (UB). It is calculated as $\Delta LB = 100 \times [(UB - LB)/UB]$ where UB is the optimal/best solution obtained by the proposed branch-and-cut algorithm. Table 1 reports the results. Each cell in the table reports average lower bound percentage gap, $\overline{\Delta LB}$, over four instances, which are obtained from original LRP instance by applying four separation strategies (i.e., X, Y, Z and W type). The first column gives the names of the original VRP instances used to produce LRPSPD instances. In the next column, named *Original*, $\overline{\Delta LB}$ for the original formulation is reported. The other columns in

Table 2
Marginal contribution of each family of valid inequality.

Instances*	Average percentage gap ($\overline{\Delta LB}$)								
	Original	Full	(25)	(26)	(27)	(28)	(29)	(30), (31)	(32)
Srivastava86-8×2	15.71	3.21	8.74	4.75	4.65	4.65	8.80	4.65	4.65
Perl83-12×2	20.71	2.54	13.16	5.79	2.94	2.94	5.71	2.94	2.94
Gaskell67-21×5	24.43	5.96	13.86	5.99	12.35	12.35	20.59	12.35	12.35
Gaskell67-22×5	23.36	4.20	11.91	5.02	4.16	4.16	15.75	4.16	4.16
Min92-27×5	35.56	13.23	17.96	13.53	13.67	13.67	30.69	13.67	13.67
Gaskell67-29×5	30.07	10.58	17.98	10.66	10.65	10.65	21.76	10.65	10.65
Gaskell67-32×5_1	24.43	7.71	11.92	8.91	7.74	7.74	18.32	7.74	7.74
Gaskell67-32×5_2	23.28	5.75	9.40	7.26	5.84	5.84	16.86	5.84	5.84
Gaskell67-36×5	18.04	4.42	15.90	7.08	4.42	4.42	4.43	4.42	4.42
Ch69-50×5	24.49	9.15	19.41	10.97	9.04	9.04	12.96	9.04	9.04
Perl83-55×15	25.66	13.79	23.34	15.26	20.92	20.92	22.05	20.92	20.92
Christofides69-75×10	28.34	14.59	24.14	16.04	14.52	14.52	17.30	14.52	14.52
Perl83-85×7	27.55	15.48	25.45	15.69	24.82	24.82	26.57	24.82	24.82
Daskin95-88×8	29.76	21.77	26.76	28.11	29.70	29.53	29.70	29.70	29.70
Christofides69-100×10	23.76	10.55	18.46	12.78	10.45	10.45	14.76	10.45	10.45
Average	25.01	9.53	17.23	11.19	11.72	11.71	17.75	11.72	11.72

* Instances are derived from Barreto's set.

Table 3
Computational results for the instances derived from Barreto's test set by Salhi and Nagy's separation approach (i.e. BSN).

Instances	DSS	UB	B&C				B&C1				CPLEX	
			Gap	Nodes	Cuts	SolTime	Gap	Nodes	Cuts	SolTime	Gap	SolTime
Srivastava86-8×2	X	625.43	0.00	0	20	0.38	0.00	0	20	0.27	0.00	0.41
	Y	625.43	0.00	3	16	0.26	0.00	3	16	0.20	0.00	0.67
Perl83-12×2	X	242.41	0.00	2	13	0.60	0.00	2	14	0.40	0.00	0.78
	Y	242.41	0.00	2	12	0.55	0.00	4	12	0.36	0.00	3.47
Gaskell67-21×5	X	454.48	0.00	153	54	13.34	0.00	416	79	32.75	0.00	2243.03
	Y	454.48	0.00	141	43	8.98	0.00	290	53	16.13	0.00	1984.16
Gaskell67-22×5	X	629.51	0.00	3494	203	109.62	0.00	3494	203	99.65	0.00	1440.88
	Y	629.51	0.00	3310	175	64.99	0.00	1896	129	41.30	0.00	980.93
Min92-27×5	X	2998.80	0.00	1078	121	91.87	0.00	1378	139	126.64	6.75	14400.00
	Y	2998.80	0.00	1440	114	190.62	0.00	1159	104	130.73	7.27	14400.00
Gaskell67-29×5	X	490.34	0.00	12	33	6.63	0.00	12	33	5.97	1.13	14400.00
	Y	490.34	0.00	33	32	6.57	0.00	33	32	5.40	1.80	14400.00
Gaskell67-32×5_1	X	563.47	0.00	90	87	18.77	0.00	217	194	58.66	1.37	14400.00
	Y	563.47	0.00	120	92	27.31	0.00	153	156	58.76	1.55	14400.00
Gaskell67-32×5_2	X	507.03	0.00	53	54	13.72	0.00	37	59	14.87	0.00	720.88
	Y	507.03	0.00	4	39	7.87	0.00	4	39	7.15	0.00	461.29
Gaskell67-36×5	X	494.86	0.00	8	94	18.36	0.00	227	168	56.33	1.97	14400.00
	Y	494.86	0.00	8	117	18.13	0.00	72	123	29.55	2.49	14400.00
Christofides 69-50×5	X	576.69	2.40	2087	792	14400.00	2.65	3019	1225	14400.00	4.95	14400.00
	Y	578.97	2.69	2681	840	14400.00	3.00	2697	1428	14400.00	5.14	14400.00
Perl83-55×15	X	985.56	2.94	422	778	14400.00	4.09	356	710	14400.00	20.52	14400.00
	Y	988.06	3.14	502	743	14400.00	4.31	297	518	14400.00	20.70	14400.00
Christofides 69-75×10	X	888.16	10.33	127	542	14400.00	12.01	140	673	14400.00	17.26	14400.00
	Y	884.00	8.90	129	593	14400.00	8.98	130	643	14400.00	17.53	14400.00
Perl83-85×7	X	1381.57	5.88	80	649	14400.00	5.88	80	749	14400.00	26.06	14400.00
	Y	1368.64	4.96	96	655	14400.00	4.96	101	673	14400.00	25.40	14400.00
Daskin95-88×8	X	399.17	11.20	0	34	14400.00	18.71	0	0	4826.98	N/A	14400.00
	Y	401.98	0.00	0	45	14068.40	1.05	0	25	14400.00	N/A	14400.00
Christofides 69-100×10	X	880.18	7.45	14	279	14400.00	7.89	16	234	14400.00	13.90	14400.00
	Y	880.89	6.17	100	339	14400.00	5.59	64	344	14400.00	13.90	14400.00
Average			2.20	540	254	5768.90	2.64	543	193	5463.74		9861.22

the table give $\overline{\Delta LB}$s for the original formulation with an extra family of valid inequalities in the column caption introduced. The last row in table reports the average over all instances for each column.

As seen in Table 1, the original formulation has large lower bound percentage gaps which range from 15.71% to 35.56%. The average percentage gap is around 25%. The performance of the original formulation is improved by adding a family of valid inequalities at a time. The inequality (29) is the most efficient one and its inclusion in the original formulation reduces the average percentage gap to 18.09% and improvement is around 7% on average. The inequalities (25) and (29) are also efficient ones and they improve the percentage gaps around 5.17% and 5.11% on average, respectively. Another efficient inequality (28) leads to improvement on the percentage gaps around 3.20%. The inequalities (26) and (32) in the original formulation provide slightly improvement on the percentage gaps around 1.26% and 0.47% on average, respectively.

In the second experiment, we investigate the marginal contributions of each family of inequalities. The column named *Full* in Table 2 gives average percentage gap ($\overline{\Delta LB}$) of the linear relaxation of the full formulation which is obtained by adding all the inequalities to the original formulation. Following columns report the gaps obtained by removing a family of inequalities from the full formulation. The removed inequalities are denoted in the column caption. As seen in Table 2, the full formulation improves the lower bounds according to the original formulation and the average percentage gap reduces to 9.53%. Removing inequalities (25) and (29) increase the percentage gaps around 7.70% and 8.22% on average, respectively. It is also possible to observe smaller but a significant increase (around 2% on average) on the percentage gaps by removing other inequalities. These results reveal that all valid inequalities should be in the formulation.

5.3. Results for the branch-and-cut algorithm

We have coded the branch-out-cut algorithm in C++ by using LP/MIP solver CPLEX 11.1 Callable Library. All experiments have been performed on Intel Xeon 3.16 GHz equipped with 1 GB RAM computer and a time limit of 4 hours has been imposed on each instance.

Regarding to the SA algorithm, which is used to improve the upper bounds obtained in the nodes of the enumeration tree, we have implemented following parameter values selected based on our previous experience: the initial temperature (T_o) is taken as 665 in which an inferior solution (inferior by 70% relative to current solution) is accepted with a probability of 0.90, the final temperature (T_f) is taken as 0.15 such that a solution which is inferior by 1% relative to current solution is accepted with a probability of 0.1% and the number of neighbors (LS) generated by each moving strategy is set to number of customers, $|N_C|$, of the instance.

It is important to note that a comparison of B&C with a state-of-the-art algorithm is not possible since, to the best our knowledge, there is no competing approach in this research area. Thus, the performance of our approach has been compared to that of LP/MIP solver CPLEX

Table 4
Computational results for the instances derived from Barreto's test set by Angelelli and Mansini's separation approach (i.e. BAM).

Instances	DSS	UB	B&C				B&C1				CPLEX	
			Gap	Nodes	Cuts	SolTime	Gap	Nodes	Cuts	SolTime	Gap	SolTime
Srivastava86-8×2	W	873.58	0.00	0	1	0.00	0.00	0	1	0.00	0.00	0.03
	Z	806.06	0.00	0	3	0.00	0.00	0	3	0.00	0.00	0.03
Perl83-12×2	W	243.98	0.00	0	12	0.57	0.00	0	12	0.41	0.00	2.37
	Z	243.98	0.00	0	11	0.65	0.00	0	11	0.46	0.00	3.48
Gaskell67-21×5	W	528.42	0.00	2078	351	290.18	0.00	5747	788	1405.25	0.00	12777.80
	Z	513.30	0.00	701	151	89.43	0.00	2244	485	487.41	0.00	7564.93
Gaskell67-22×5	W	653.80	0.00	1	12	3.76	0.00	3	13	3.87	0.00	34.30
	Z	653.80	0.00	0	10	2.66	0.00	3	15	3.29	0.00	50.87
Min92-27×5	W	3142.02	0.00	137	86	19.66	0.00	183	97	30.89	1.79	14400.00
	Z	3142.02	0.00	181	76	18.12	0.68	13278	1372	14400.00	2.38	14400.00
Gaskell67-29×5	W	592.10	0.00	6	45	6.76	0.00	9	47	10.92	3.51	14400.00
	Z	592.10	0.00	444	110	100.91	0.00	1074	318	489.22	10.02	14400.00
Gaskell67-32×5_1	W	696.38	0.00	13009	617	5694.09	0.00	10899	725	5225.26	7.27	14400.00
	Z	643.37	0.00	1219	308	259.67	0.00	9554	1069	4999.97	4.29	14400.00
Gaskell67-32×5_2	W	595.27	0.00	8	80	15.68	0.00	3582	364	304.35	3.20	14400.00
	Z	564.33	0.00	87	100	30.90	0.00	194	137	59.93	0.00	8179.27
Gaskell67-36×5	W	540.37	0.00	18	165	27.20	0.00	1962	403	657.10	1.67	14400.00
	Z	540.37	0.00	40	148	46.46	0.00	258	291	151.88	2.55	14400.00
Christofides 69-50×5	W	708.37	4.52	1056	1169	14400.00	4.67	953	1297	14400.00	8.53	14400.00
	Z	701.91	3.67	835	1266	14400.00	3.74	920	1522	14400.00	11.21	14400.00
Perl83-55×15	W	1330.85	20.93	540	1398	14400.00	22.56	490	1271	14400.00	23.90	14400.00
	Z	1338.30	20.81	387	1492	14400.00	22.93	344	1363	14400.00	24.41	14400.00
Christofides 69-75×10	W	1177.65	17.55	120	628	14400.00	17.55	157	715	14400.00	19.13	14400.00
	Z	1108.82	12.81	194	688	14400.00	14.62	182	663	14400.00	15.65	14400.00
Perl83-85×7	W	1901.09	23.56	90	728	14400.00	23.53	95	735	14400.00	25.67	14400.00
	Z	1893.28	23.41	148	679	14400.00	23.42	119	700	14400.00	25.44	14400.00
Daskin95-88×8	W	533.37	21.96	0	0	14400.00	21.60	0	0	14400.00	N/A	14400.00
	Z	487.81	15.57	0	19	14400.00	14.87	0	34	14400.00	N/A	14400.00
Christofides 69-100×10	W	1079.41	12.88	16	541	14400.00	12.74	40	521	14400.00	19.29	14400.00
	Z	1038.26	9.26	19	420	14400.00	9.26	30	480	14400.00	16.25	14400.00
Average			6.23	712	377	5980.23	6.41	1744	515	6701.01		11033.77

11.1 and a new variant of B&C, called B&C1. We run CPLEX under default settings directly on the flow-based formulation of the LRPSPD in Section 2. The B&C1 is derived by excluding SA algorithm from the B&C. Note that SA algorithm is implemented only to improve the initial feasible solution in the B&C1. The rationale behind this implementation is to assess the effectiveness of SA algorithm on the performance of B&C. These two approaches have also been run on Intel Xeon 3.16 GHz equipped with 1 GB RAM computer for a time limit of 4 hours.

Tables 3–6 depict computational results for the B&C, B&C1 and CPLEX on LRPSPD instances derived from Barreto's and Prodhon's sets by applying Salhi and Nagy's and Angelelli and Mansini's demand separation approaches, respectively. The first column in the tables is the same as in previous tables. The column labeled *DSS* denotes demand separation strategy. The column labeled *UB* reports the upper bound value for the corresponding instance. It is important to note that the upper bound shows the optimal value for the instances which are solved to optimality within 4 hours. Successive four columns summarize computational results for the B&C. The column labeled *Gap* reports the percentage gap. The column labeled *Nodes* gives the number of nodes explored by the enumeration tree. The column labeled *Cuts* presents the number of violated inequalities added to the model. The column labeled *Soltime* reports the CPU time in seconds required to run the algorithm. The next four columns present computational results for the B&C1. The last two columns report percentage gap and CPU time in seconds for the CPLEX. Note that the percentage gaps for each algorithm are computed as $100[(UB^* - LB^*)/UB^*]$ where UB^* is the best upper bound obtained by any of three algorithms and LB^* is the lower bound found by the corresponding algorithm.

Table 5
Computational results for the instances derived from Prodhon's test set by Salhi and Nagy's separation approach (i.e. PSN).

Instances	DSS	UB	B&C				B&C1				CPLEX	
			Gap	Nodes	Cuts	SolTime	Gap	Nodes	Cuts	SolTime	Gap	SolTime
20-5-1a	X	16816.50	0.00	2641	195	84.51	0.00	3101	207	153.08	4.45	14400.00
	Y	16816.00	0.00	2048	160	73.77	0.00	10003	417	496.34	4.41	14400.00
20-5-1b	X	9167.14	0.00	0	1	1.86	0.00	0	1	0.44	0.00	0.65
	Y	9167.14	0.00	0	1	0.27	0.00	0	1	0.38	0.00	0.46
20-5-2a	X	17814.70	0.00	459	152	27.93	0.00	961	317	118.38	1.95	14400.00
	Y	17814.70	0.00	390	104	16.02	0.00	754	256	74.81	1.94	14400.00
20-5-2b	X	10257.30	0.00	0	15	1.61	0.00	14	21	2.86	0.00	22.70
	Y	10257.30	0.00	0	20	1.38	0.00	0	20	1.38	0.00	29.25
50-5-1a	X	16350.00	0.17	1835	823	14400.00	0.23	1060	1114	14400.00	1.72	14400.00
	Y	16355.20	0.22	1800	1161	14400.00	0.24	1380	1290	14400.00	1.76	14400.00
50-5-1b	X	13132.90	0.04	3069	499	14400.00	0.09	3060	1011	14400.00	2.57	14400.00
	Y	13132.90	0.04	6964	364	14400.00	0.07	5519	802	14400.00	0.23	14400.00
50-5-2a	X	26395.60	0.10	1251	1309	14400.00	1.25	1250	1272	14400.00	2.73	14400.00
	Y	26392.70	0.09	1602	1049	14400.00	1.24	1854	1642	14400.00	2.71	14400.00
50-5-2b	X	22268.50	0.00	440	140	213.75	0.02	1502	1070	14400.00	2.68	14400.00
	Y	22268.50	0.00	3086	210	1147.76	0.03	1660	939	14400.00	2.66	14400.00
50-5-3a	X	11624.20	0.24	3224	1278	14400.00	0.25	3565	1904	14400.00	0.47	14400.00
	Y	11626.60	0.25	4066	1214	14400.00	0.28	2745	2017	14400.00	0.50	14400.00
50-5-3b	X	8472.39	0.12	6584	495	14400.00	0.12	4901	1022	14400.00	0.36	14400.00
	Y	8469.87	0.03	6584	495	14400.00	0.09	4680	1062	14400.00	0.33	14400.00
100-5-1a	X	102388.00	0.18	37	1101	14400.00	0.18	30	945	14400.00	1.19	14400.00
	Y	102381.00	0.17	34	912	14400.00	0.17	50	852	14400.00	1.18	14400.00
100-5-1b	X	94884.00	0.12	50	511	14400.00	0.12	28	569	14400.00	1.11	14400.00
	Y	94878.80	0.12	60	487	14400.00	0.11	50	639	14400.00	1.10	14400.00
100-5-2a	X	105655.00	0.11	87	898	14400.00	0.11	77	881	14400.00	7.10	14400.00
	Y	105655.00	0.11	89	552	14400.00	0.11	90	559	14400.00	7.48	14400.00
100-5-2b	X	97213.80	0.08	61	625	14400.00	1.10	40	593	14400.00	9.62	14400.00
	Y	97206.00	0.07	60	741	14400.00	1.10	34	642	14400.00	8.06	14400.00
100-5-3a	X	56552.10	0.17	24	889	14400.00	0.17	5	857	14400.00	1.55	14400.00
	Y	56581.90	0.22	35	980	14400.00	0.22	5	1033	14400.00	1.60	14400.00
100-5-3b	X	50224.60	0.11	116	484	14400.00	0.11	30	708	14400.00	1.75	14400.00
	Y	50220.80	0.09	104	597	14400.00	0.09	40	679	14400.00	1.80	14400.00
100-10-1a	X	109785.00	1.07	51	713	14400.00	1.62	60	742	14400.00	3.02	14400.00
	Y	109787.00	1.07	37	785	14400.00	1.62	50	806	14400.00	2.98	14400.00
100-10-1b	X	102430.00	0.08	71	417	14400.00	0.08	65	409	14400.00	2.44	14400.00
	Y	102426.00	0.07	96	372	14400.00	0.07	50	406	14400.00	2.43	14400.00
100-10-2a	X	155190.00	32.08	64	725	14400.00	32.08	67	714	14400.00	32.95	14400.00
	Y	107521.00	1.97	60	728	14400.00	1.96	54	672	14400.00	3.01	14400.00
100-10-2b	X	99140.10	1.08	63	363	14400.00	1.08	41	331	14400.00	5.31	14400.00
	Y	99138.40	1.08	63	392	14400.00	1.08	60	370	14400.00	5.31	14400.00
100-10-3a	X	100702.00	1.11	40	874	14400.00	1.11	58	716	14400.00	8.72	14400.00
	Y	99913.60	0.33	49	819	14400.00	0.33	33	913	14400.00	4.33	14400.00
100-10-3b	X	93450.50	0.07	30	267	14400.00	0.07	47	393	14400.00	4.23	14400.00
	Y	93475.00	0.10	58	340	14400.00	0.09	23	385	14400.00	16.64	14400.00
Average			0.98	1079	574	11162.93	1.11	1116	732	11801.08	3.78	13092.11

Table 6
Computational results for the instances derived from Prodhon's test set by Angelelli and Mansini's separation approach (PAM).

Instances	DSS	UB	B&C				B&C1				CPLEX	
			Gap	Nodes	Cuts	SolTime	Gap	Nodes	Cuts	SolTime	Gap	SolTime
20-5-1a	W	26457.70	0.00	1890	294	232.48	0.02	8234	5537	14400.00	0.94	14400.00
	Z	26461.30	0.00	1059	180	75.24	0.02	16931	3502	14400.00	1.41	14400.00
20-5-1b	W	18718.80	0.00	0	17	3.22	0.00	0	17	1.32	5.02	14400.00
	Z	18703.00	0.00	0	18	0.94	0.00	0	18	1.36	4.87	14400.00
20-5-2a	W	27988.70	0.00	701	162	37.79	0.00	6758	711	1020.52	1.90	14400.00
	Z	27980.60	0.00	1283	177	73.81	0.00	8684	486	1050.79	1.69	14400.00
20-5-2b	W	17125.50	0.00	106	35	7.09	0.00	293	61	13.87	5.21	14400.00
	Z	17120.50	0.00	222	30	4.40	0.00	222	30	5.25	5.53	14400.00
50-5-1a	W	33786.20	12.28	300	992	14400.00	3.18	268	1435	14400.00	5.32	14400.00
	Z	32804.50	0.30	528	1626	14400.00	10.50	110	1122	14400.00	2.94	14400.00
50-5-1b	W	26541.40	0.21	631	922	14400.00	0.21	410	1053	14400.00	3.06	14400.00
	Z	26530.80	0.18	500	877	14400.00	0.19	353	970	14400.00	3.27	14400.00
50-5-2a	W	42860.40	2.54	252	1344	14400.00	2.54	660	1769	14400.00	3.52	14400.00
	Z	41825.80	0.16	1080	1425	14400.00	0.17	1070	1744	14400.00	1.57	14400.00
50-5-2b	W	35661.90	0.13	1601	818	14400.00	0.14	973	935	14400.00	1.83	14400.00
	Z	35648.40	0.11	2743	1091	14400.00	0.12	509	575	14400.00	2.05	14400.00
50-5-3a	W	23533.10	4.65	232	1285	14400.00	4.65	960	1296	14400.00	5.34	14400.00
	Z	23501.50	0.26	1070	2260	14400.00	2.10	813	960	14400.00	4.48	14400.00
50-5-3b	W	17182.00	0.21	1541	1392	14400.00	2.26	1228	1180	14400.00	5.76	14400.00
	Z	17174.10	0.13	2050	1173	14400.00	0.13	736	1093	14400.00	5.24	14400.00
100-5-1a	W	160277.00	1.36	70	1453	14400.00	1.37	50	1360	14400.00	1.76	14400.00
	Z	159200.00	0.73	20	1418	14400.00	0.73	42	1373	14400.00	26.37	14400.00
100-5-1b	W	145697.00	0.78	80	918	14400.00	0.78	83	788	14400.00	0.85	14400.00
	Z	145650.00	0.76	30	687	14400.00	0.76	40	656	14400.00	0.91	14400.00
100-5-2a	W	168371.00	28.20	30	1258	14400.00	28.20	18	1317	14400.00	28.29	14400.00
	Z	172418.00	30.38	3	1151	14400.00	29.90	13	1145	14400.00	32.04	14400.00
100-5-2b	W	153827.00	29.57	62	867	14400.00	29.57	80	831	14400.00	32.00	14400.00
	Z	153776.00	29.55	10	905	14400.00	29.55	13	936	14400.00	29.77	14400.00
100-5-3a	W	114517.00	3.65	42	901	14400.00	3.65	57	902	14400.00	3.81	14400.00
	Z	112467.00	1.89	24	1260	14400.00	1.90	65	841	14400.00	2.01	14400.00
100-5-3b	W	100006.00	2.13	90	601	14400.00	2.13	100	563	14400.00	2.50	14400.00
	Z	98972.50	1.11	60	566	14400.00	1.11	65	627	14400.00	1.45	14400.00
100-10-1a	W	224445.00	1.41	60	773	14400.00	5.87	71	884	14400.00	14.49	14400.00
	Z	222434.00	0.53	50	749	14400.00	5.03	46	799	14400.00	17.04	14400.00
100-10-1b	W	209084.00	0.10	120	625	14400.00	0.10	110	566	14400.00	17.28	14400.00
	Z	208958.00	0.04	81	521	14400.00	0.04	127	466	14400.00	15.22	14400.00
100-10-2a	W	218395.00	22.30	30	929	14400.00	22.95	54	829	14400.00	24.70	14400.00
	Z	218243.00	22.42	40	911	14400.00	22.42	54	928	14400.00	23.81	14400.00
100-10-2b	W	159507.00	1.81	60	381	14400.00	1.74	99	443	14400.00	5.24	14400.00
	Z	202371.85	22.78	60	381	14400.00	22.78	110	494	14400.00	23.97	14400.00
100-10-3a	W	210355.00	20.88	7	1148	14400.00	20.87	7	1160	14400.00	21.26	14400.00
	Z	166404.22	2.44	7	1148	14400.00	1.89	10	1243	14400.00	2.80	14400.00
100-10-3b	W	196017.00	21.58	14	481	14400.00	21.82	10	493	14400.00	22.02	14400.00
	Z	152088.46	1.37	14	481	14400.00	0.76	50	717	14400.00	1.04	14400.00
Average			6.12	429	833	11791.71	6.41	1150	1020	12483.94	9.58	14400.00

When Tables 3–6 are examined, it is seen that among the 148 instances considered, 55 instances are solved to optimality by the B&C and 13 of these instances are solved at the root node of enumeration tree. Meantime, 93 instances are still unsolved after 4 hours of computation. For these instances, the final percentage gap between the lower bound and the UB is 7.45% on average. It is also important to note that while 14 unsolved instances have a final gap under 0.1%, the number of unsolved instances having a final gap under 1% equals to 42. Moreover, all the best bounds reported in column UB have been found by B&C.

With respect to the results for the instances derived from Barreto's test set (see Tables 3 and 4), we can observe that all instances up to 36 customers and 5 depots are solved to optimality within around 5 minutes, with the exception of the instance Gaskell67-32×5_1 derived by separation strategy *W*, which needs 1.5 hours to reach optimal solution. It is also worth noting that the B&C obtains the optimal solution for an instance with 88 customers and 8 depots derived by separation strategy *Y* within 4 hours. Meanwhile, 23 out of 60 instances are not solved to optimality and the percentage gaps for these instances change between 2.40% and 23.56%. Note that 12 instances of them have percentage gaps lower than 10%.

From Tables 5 and 6, where results for the instances derived from Prodhon's test set are reported, it is seen that the instances with 20 customers and 5 depots are solved to optimality within 4 minutes. Also two instances with 50 customers and 5 depots (denoted by 50-5-2b), which are obtained by applying Salhi and Nagy's separation approach (i.e. *X* and *Y* strategies), are solved to optimality within 20 minutes. For the unsolved instances (70 out of 88 instances), the percentage gaps are between 0.04% and 32.04%. Meantime, only 11 out of 70 instances have percentage gaps greater than 5%.

It is also worth noting that the instances derived from Barreto's and Prodhon's test sets by using Angelelli and Mansini's demand separation approach (denoted as BAM and PAM, respectively) are more difficult to solve than those derived from same test sets by Salhi and Nagy's demand separation approach (denoted as BSN and PSN, respectively). While average gaps are 2.20% and 0.98% for the BSN and PSN (see Tables 3 and 5), respectively, this value increases to 6.23% and 6.12% for the BAM and PAM (see Tables 4 and 6), respectively. The number of cuts and explored nodes for BAM and PAM are also greater than those of BSN and PSN, respectively. For example, while the average number of cuts for the BAM is 377, it is equal to 254 for the BSN. This is expected result when we consider that the pickup and delivery demands generated by Angelelli and Mansini's demand separation approach (i.e. Z and W strategies) are always bigger than those obtained by Salhi and Nagy's demand separation approach (i.e. X and Y strategies) as explained in Subsection 5.1. To conclude, difficulty of solving of an instance increases when pickup and delivery demands of customers for an instance are close to the vehicle capacity.

To assess the effect of the SA algorithm on the performance of the B&C, we compare our algorithm B&C with the B&C1 which is obtained by removing SA algorithm used to improve upper bounds during the search process. From Tables 3–6, followings are observed: removing SA algorithm leads to a worse average gap (4.14% versus 3.88%), to a large average CPU time (9113 s versus 8676 s), to a large average number of nodes explores (1139 versus 690) and to reduction on the number of instances solved to optimality (49 instead of 55) over 148 instances.

Finally, we compare results of B&C with those obtained by CPLEX. As seen in Tables 3–6, B&C outperforms CPLEX in terms of bound quality and computational effort. CPLEX solves only 23 out of 148 instances to optimality and this is achieved with larger CPU times (12086 s in average). 34 out of 125 unsolved instances have percentage gaps greater than 10%. Meanwhile, CPLEX does not find feasible integer solutions within a time limit for four instances generated from Daskin95-88$\times$8. Furthermore, none of instances derived from Prodhon's test set by Angelelli and Mansini's separation approach (PAM) is solved to optimality by CPLEX within a time limit.

6. Conclusion

In this paper, we have considered a general case of the location-routing problem called location-routing problem with simultaneously pickup and delivery, LRPSPD, and proposed for the first time a branch-and-cut algorithm for the exact solution of the LRPSPD. We have adapted several valid inequalities, which were developed for the vehicle routing and facility location problems in the literature, in the implementation of the proposed branch-and-cut algorithm. We have also developed separation algorithms for these inequalities and a heuristic algorithm based on simulated annealing (SA) to improve solutions found during the search process of the algorithm. Finally, we have presented computational results conducted on 148 new test instances derived from the location-routing problem instances. A comparison with CPLEX shows that the proposed branch-and-cut algorithm is a viable approach to solve small and medium size LRPSPD instances. Furthermore, the computational results reveal that simulated annealing helps the branch and cut algorithm in finding high quality solutions by reducing computation time and exploring fewer nodes of tree.

In terms of future research directions, the proposed branch-and-cut algorithm can be modified to take into account more realistic aspects of the LRPSPD such as dynamic environment and stochastic demands. The proposed branch and cut algorithm implements straightforward branching scheme (i.e. branching most fractional variable), and four polynomial- and three exponential-size valid inequalities. New valid inequalities can be developed for the problem. Meantime, it would be worthwhile to investigate the effects of new valid inequalities and different branching schemes, such as closest to integer, farthest from integer, or constraint branching, etc., on the performance of the algorithm.

Acknowledgements

The authors are indebted to the anonymous referees for their helpful comments and suggestions. This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No: 108E069.

References

Achuthan, N.R., Caccetta, L., Hill, S.P., 2003. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. Transportation Science 37 (2), 153–169.
Albareda-Sambola, M., Diaz, J.A., Fernandez, E., 2005. A compact model and tight bounds for a combined location-routing problem. Computers and Operations Research 32, 407–428.
Ambrosino, D., Grazia ScutellaI, M., 2005. Distribution network design: New problems and related models. European Journal of Operational Research 165 (3), 610–624.
Angelelli, E., Mansini, R., 2002. The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Lecture Notes in Economics and Mathematical Systems. Springer, Germany, pp. 249–267.
Baldacci, R., Hadjiconstantinou, E., Mingozzi, A., 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research 52 (5), 723–738.
Barreto, S.S., 2003. Available from: <http://sweet.us.pt/~iscf143>.
Barreto, S.S., Ferreira, C., Paixao, J., Santos, B.S., 2007. Using clustering analysis in a capacitated location-routing problem. European Journal of Operational Research 179, 968–977.
Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Wolfler Calvo, R., 2006. A branch and cut method for the capacitated location-routing problem. International Conference on Service Systems and Service Management, pp. 1541–1546.
Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15, 1–31.
Berger, R.T., Coullard, C.R., Daskin, M., 2007. Location-routing problems with distance constraints. Transportation Science 41 (1), 29–43.
Berube, J.F., Gendreau, M., Potvin, J.Y., 2009. A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem. Networks. doi:10.1002/net.
Christofides, N., Eilon, S., 1969. An algorithm for the vehicle dispatching problem. Operational Research Quarterly 20 (3), 309–318.
Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581.
Cordeau, J.F., Dell'Amico, M., Iori, M., 2009. Branch-and-cut for the pickup and delivery traveling salesman problem with FIFO loading. Computers and Operations Research. doi:10.1016/j.cor.2009.08.003.
Duhamel, C., Lacomme, P., Prins, C., Prodhon, C., 2009. A GRASP $\times$ ELS approach for the capacitated location-routing problem. Computers and Operations Research. doi:10.1016/j.cor.2009.07.004.
Gouveia, L., 1995. A result on projection for the vehicle routing problem. European Journal of Operational Research 85, 610–624.
Hansen, P.H., Hegedahl, B., Hjortkjaer, S., Obel, B., 1994. Heuristic solution to the warehouse location-routing problem. European Journal of Operational Research 76 (1), 111–127.

Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B., 2009a. Formulations for a Location-Routing Problem with Simultaneous Pickup and Delivery. Research Paper. <http://w3.gazi.edu.tr/~fulyaal/Papers/LRPSPD_MIPFormulations.pdf>.
Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B., 2009b. Vehicle Routing Problem with Simultaneous Pickup and Delivery: Mixed Integer Programming Formulations and Comparative Analyses. Research Paper. <http://w3.gazi.edu.tr/~fulyaal/Papers/VRPSPD_MIPFormulations.pdf>.
Labbe, M., Rodríguez-Martin, I., Salazar-González, J.J., 2004. A branch-and-cut algorithm for the plant-cycle location problem. Journal of the Operational Research Society 55 (5), 513–520.
Laporte, G., Nobert, Y., 1981. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6, 224–226.
Laporte, G., Nobert, Y., Pelletier, P., 1983. Hamiltonian location problems. European Journal of Operational Research 12, 82–89.
Laporte, G., Nobert, Y., Arpin, D., 1986. An exact algorithm for solving a capacitated location-routing problem. Annals of Operations Research 6, 293–310.
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., Nobert, Y., 1988. Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science 22 (3), 161–172.
Letchford, A.N., Salazar-Gonzalez, J.J., 2006. Projection results for vehicle routing. Mathematical Programming 105 (2-3), 251–274.
Marinakis, Y., Marinaki, M., 2008. A particle swarm optimization algorithm with path relinking for the location routing problem. Journal of Mathematical Modelling and Algorithms 7, 59–78.
Melechovsky, J., Prins, C., 2005. A metaheuristic to solve a location routing problem with non-linear costs. Journal of Heuristics 11, 375–391.
Min, H., Jayaraman, V., Srivastava, R., 1998. Combined location-routing problems: A synthesis and future research directions. European Journal of Operational Research 108 (1), 1–15.
Mosheiov, G., 1995. The pickup delivery location problem on networks. Networks 26, 243–251.
Nagy, G., Salhi, S., 1998. The many-to-many location-routing problem. TOP 6, 261–275.
Nagy, G., Salhi, S., 2007. Location-routing: Issues, models and methods. European Journal of Operational Research 177 (2), 649–672.
Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. Journal für Betriebswirtschaft 58 (1), 21–51.
Perl, J., Daskin, M.S., 1984. A unified warehouse location-routing methodology. Journal of Business Logistics 5 (1), 92–111.
Perl, J., Daskin, M.S., 1985. A warehouse location-routing problem. Transportation Research 19 (5), 381–396.
Prins, C., Prodhon, C., Wolfler Calvo, R., 2006a. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR 4, 221–238.
Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006b. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. EvoCOP2006, 183–194.
Prodhon, C., 2008. <http://prodhonc.free.fe/homepage>.
Salhi, S., Rand, G.K., 1989. The effect of ignoring routes when locating depots. European Journal of Operational Research 45, 150–156.
Salhi, S., Nagy, G., 1999. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society 50, 1034–1042.
Srivastava, R., Benton, W.C., 1990. The location-routing problem: Consideration in physical distribution system design. Computers and Operations Research 6, 427–435.
Srivastava, R., 1993. Alternate solution procedures for the location-routing problem. Omega 21 (4), 497–506.
Suman, B., Kumar, P., 2006. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society 57 (10), 1143–1160.
Tuzun, D., Burke, L.I., 1999. A Two-phase tabu search approach to the location routing problem. European Journal of Operational Research 116 (1), 87–99.
Webb, M.H.J., 1968. Cost functions in the location of depots for multiple delivery journeys. Operational Research Quarterly 19 (3), 311–320.
Wu, T.H., Low, C., Bai, J.W., 2002. Heuristic solutions to multi-depot location-routing problems. Computers and Operations Research 29 (10), 1393–1415.
Yu, V.F., Lin, S.W., Lee, W., Ting, C.J., 2010. A Simulated Annealing Heuristic for the Capacitated Location Routing Problme. Computers and Industrial Engineering 58 (2), 288–299.