EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
ScienceDirect
www.elsevier.com/locate/ejor

A Profit-Maximization Location-Routing-Pricing Problem: A Branch-and-Price Algorithm

Amir Ahmadi-Javid , Elahe Amiri , Mahla Meskar

PII: S0377-2217(18)30135-8
DOI: 10.1016/j.ejor.2018.02.020
Reference: EOR 14980

To appear in: *European Journal of Operational Research*

Received date: 14 July 2015
Revised date: 7 February 2018
Accepted date: 8 February 2018

Please cite this article as: Amir Ahmadi-Javid , Elahe Amiri , Mahla Meskar , A Profit-Maximization Location-Routing-Pricing Problem: A Branch-and-Price Algorithm, *European Journal of Operational Research* (2018), doi: 10.1016/j.ejor.2018.02.020

Highlights

- This paper studies a location-routing problem with profit in a supply-chain network.
- Customer demands are price-sensitive, and delivered pricing policy is used.
- The problem simultaneously determines location, routing, and pricing decisions.
- Two formulations for the problem are proposed.
- A branch-and-price algorithm is developed to efficiently solve the problem.

A Profit-Maximization Location-Routing-Pricing Problem: A Branch-and-Price Algorithm

Amir Ahmadi-Javid[1], Elahe Amiri, and Mahla Meskar

Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract This paper for the first time considers a profit-maximization location-routing problem with price-sensitive demands. The problem determines the location of facilities, the allocation of vehicles and customers to established facilities, and the pricing and routing decisions in order to maximize the total profit of serving customers. A mixed-integer linear programming model is presented, which can only be used to solve small-size instances with commercial optimization solvers. Then, the model is reformulated as a set-packing model and solved by an efficient branch-and-price algorithm for large-size instances. The proposed algorithm can also be used to solve the more basic location-routing problem with profit where demands are not price-sensitive, which has not been considered by any research earlier. Our numerical study indicates the substantial advantage of the integrated model.

Keywords. Location-Routing Problems (LRP); Vehicle Routing Problems with Profit (VRPP); Price-sensitive demands and delivered pricing; Mixed-Integer Linear Programming (MILP); Branch-and-price and column generation

1. Introduction

Location Routing Problems (LRPs) involve selecting the optimal number and locations of facilities, allocating customers to established facilities and constructing delivery routes. This class of decision problems has been studied since the mid-1970s. Laporte (1988) provided a survey on deterministic LRPs and described various formulations of the problem, as well as the solution methods and computational results used up to 1988. Min et al. (1998) surveyed LRPs and used a two-way classification scheme for LRPs in terms of problem characteristics and solution methods. A detailed review of LRP models, applications and solution methods was also provided by Nagy and Salhi (2007) and Lopez et al. (2013). In recent studies, new variants of LRPs have been put forward, which simultaneously optimize location, routing and other decisions such as inventory decisions. Ahmadi-Javid and Azad (2010) presented such a

[1] Corresponding author ̦s email address: ahmadi_javid@aut.ac.ir

model for the first time in a stochastic supply chain system. Ahmadi-Javid and Seddighi (2013) studied a location-routing problem with production and distribution disruption risks in a supply chain network. For a comprehensive review of LRPs, one can refer to the recent surveys by Prodhon and Prins (2014), and Drexl and Schneider (2015).

LRP models presented in the literature typically minimize some performance metrics. The widely-used measures are cost functions, while some studies consider other metrics, such as the total waiting time of all the customers, the total length of the tour, the maximal waiting time of a typical customer, the average traveled length per customer, the average waiting time per customer, the average response time, the maximum lengths of the tours (Averbakh & Berman, 1994; Averbakh et al., 1994; Jamil et al., 1994. Averbakh & Berman, 1995; Averbakh & Berman, 2002). Moreover, LRP models often assume that every customer is visited exactly once, but there are papers where some customers may not be visited or are visited more than once. For example, in Nagy and Salhi (1998), two visits may be allowed to a customer, or in Averbakh et al. (1994) and Averbakh and Berman (1995) some random customers do not require a visit; while in Albareda-Sambola et al. (2007) some randomly-selected customers are not served (for more details see the review by Lopez et al. (2013)).

To the best of our knowledge, there is little study on LRPs with profit-maximization objectives, where the visit of all customers is not mandatory. However, in the closely related context of Vehicle Routing Problems (VRPs) and traveling saleman problems (TSPs), there are many studies that consider routing problems with no restriction in serving all of the customers and that leave some customers unserved to increase their profit-based objective functions. In fact, it may not be profitable to satisfy all customers as the cost of serving some customers is higher than the revenue of serving them. In the following, we shortly provide an overview of this class of VRPs, which are called Vehicle Routing Problems with Profits (VRPPs).

Dissimilar to the traditional VRPs, VRPPs simultaneously choose a set of customers from potential customers and design the routes to serve the selected customers. In these problems, visiting each customer has a *profit* that represents its relative importance compared to the other customers. Hence, there are two conflicting objectives in these problems: travel-cost minimization and profit maximization. Depending on how these objectives are compromised, three classes of VRPPs have been studied in the literature: Profitable Tour Problems (PTPs), Team Orienteering Problems (TOPs), and Prize Collecting Routing Problems (PCRPs). In PTPs, both objectives are combined in a single objective, which is to maximize the difference between the total collected profit and the total travel cost. In TOPs, the objective is to maximize the total collected profit while the total travel cost does not surpass a given threshold. In PCRPs, the objective is to minimize the total travel cost such that the total collected profit is greater than a predetermined amount. Feillet et al. (2005) surveyed TSPs with profits, a subclass of PTPs, and Vansteenwegen et al. (2011) and Gunawan et al. (2016) reviewed papers on TOPs. For a general survey of VRPPs, one can refer to Archetti et al. (2014) and the references therein. Instances of recent studies on this topic are Vidal et al. (2015), El-Hajj et al. (2016), Archetti et al. (2017), and Gansterer et al. (2017).

Incorporating price-sensitive demands has attracted considerable attention in location-allocation problems and studied in several papers, e.g., Hotelling (1929), Wagner and Falkson (1975), Hansen et al. (1981), Lederer and Thisse (1990), Hanjoul et al. (1990), Hansen et al. (1997), Pفez et al. (2004), Fernعndez et al. (2007), Vogel (2011), Pelegrک et al. (2012), Fernعndez et al. (2014), Ahmadi-Javid and Ghandali (2014), He et al. (2016), and Berger et al. (2017). Moreover, Ahmadi-Javid and Hoseinpour (2015a, b) integrated pricing decisions with location-inventory problems. Several other papers also studied problems involving both inventory and pricing decisions over the last years, see, for example, the review by Chen and Simchi-Levi (2012).

However, the impact of pricing decisions has not been considered in the LRP literature to date. Actually, to the best of our knowledge, there is no study on any type of LRP with profit. This paper, for the first time, studies a Profit-Maximization LRP (PM-LRP) with price-sensitive demands. In this problem, the company uses a spatial pricing policy, under which different delivered prices are charged to customers (or spatial markets or retailers) based on their locations. The mill price (or wholesale price) of the product is the same at all main distribution centers, but the delivered price (or retail price) offered to customers (retailers) who receive their demand at their location may vary by the customersب location, to cover part of the delivery cost borne by the company.

Our PM-LRP can be applied in application areas already mentioned in the location-pricing literature where a fleet of vehicles are used to deliver the demands instead of direct transportation (Hanjoul et al., 1990). In addition, an emerging application area can be for distribution companies and online stores that serve customers who prefer to receive their required goods at their places at higher prices, while they can directly buy them from the existing shopping stores. Even if the market is not segmented, these companies are legally allowed to discriminate delivered prices offered to different zones because the distances of demand zones to their distribution centers are not the same and because people in different zones may have different demand rates depending on factors, such as the accessibility to the physical shopping stores, traffic intensity, social class, etc. Spatial price discrimination is the ability to charge different prices to consumers at different locations (Hoover, 1937; Varian, 1989; Anderson et al., 1989; Fackler & Goodwin, 2001; Lambrecht et al., 2012). Delivered pricing is a type of spatial price discrimination where the price offered to a customer is inclusive of transport charges and is dependent on the customer's location (Carlton, 1983; Espinosa, 1992; Basu et al., 2004). In decision problems under any type of spatial price discrimination, demands at different locations are considered price sensitive, as in our proposed PM-LRP. Another important application of our PM-LRP is for the case that the demands are not price sensitive. In this case, we obtain the most basic LRP with profit, which can be used in designing distribution networks in various industrial areas of VRPPs, reviewed by Archetti et al. (2014).

From a location-analysis perspective, the proposed PM-LRP is an extension of the two uncapacitated and capacitated location problems considered in Hanjoul et al. (1990), and Ahmadi-Javid and Ghandali (2014). From a routing perspective, the most related problem in the VRPP literature is a PTP with multiple capacitated vehicles, which was first studied by Archetti et al. (2009). These related problems are all NPhard, so presenting exact solution algorithm for our PM-LRP seems challenging. Here, a branch-and-price algorithm is developed, which is an extension of the existing algorithms used in the VRPP literature. The nearest work to our paper has been done by Archetti et al. (2013) who presented an algorithm for PTP with a single uncapacitated depot and a fleet of identical capacitated vehicles where a profit was associated with each customer and the objective was to maximize the difference between the collected profit and the traveling cost. We study a similar problem, but in a more complex setting. In our problem, from a given set of heterogeneous capacitated distribution centers (depots), a subset must be selected. Moreover, in our problem demands are price sensitive. Hence, each DC can offer different prices to its allocated customers, which leads to different levels of profits and demands for each customer. One can retrieve the problem studied by Archetti et al. (2013), if in our problem there is only a single uncapacitated DC with zero establishment cost and a single price can be offered to each customer.

To solve our PM-LRP in large scales, the proposed LRP is first formulated as a polynomial-size mixed-integer linear programming model, which can be used only in small sizes. Then, the model is cast as an exponential-size set-packing formulation using the Dantzig-Wolfe decomposition. The resulting model is solved by a branch-and-price algorithm, where the pricing subproblems are variants of the elementary shortest path problem, which can be solved efficiently using a label-setting dynamic programming algorithm. It should be noted that our algorithm can also be used to solve an LRP with profit where demands are not price sensitive. As surveyed above, this LRP has not yet been studied and can be considered the closest extension of VRPPs in the location literature.

Unfortunately, we cannot straightforwardly adapt the existing branch-and-price algorithms to solve our problem. Actually, there is no study that provides an exact solution method for a profit-maximization LRP where the objective is to maximize the profit and where visiting all customers are not mandatory. This shows that we have no efficient exact solution method for our LRP with profit even if demands are not price-sensitive. In fact, the price differentiation assumption enormously increases the size of the feasible solution space and increases the complexity level of our problem. Moreover, in our proposed set-packing formulation, each column simultaneously determines one route and all the prices offered to the customers on the route, while the columns of similar set-packing formulations developed for VRPs and LRPs only represent routes. This basically changes the structure of the underlying graph that is used in our label-setting algorithm to manage pricing decisions, which shows why we cannot use the existing algorithms to solve our problem.

The remainder of the paper is organized as follows. Section 2 presents the problem statement and required notation; and models the problem as a polynomial-size mixed-integer linear program and a set-packing formulation. Section 3 uses the latter to propose a branch-and-price algorithm. Section 4 reports the computational experience. Section 5 closes the paper by providing our conclusions and possible future research opportunities.

2. Problem statement and formulation

Our PM-LRP is formally described in Section 2.1. The notation needed throughout the paper is given in Section 2.2, and two formulations of the problem are presented in Sections 2.3 and 2.4.

2.1. Problem statement

In the PM-LRP considered in this paper, a set of customers (or spatial markets or retailers) and a set of main potential Distribution Centers (DCs) are given. The aim of the problem is to determine the set of open DCs, the allocation of customers to open DCs, the pricing decisions, and a set of vehicle routes from open DCs to customers in order to maximize the overall profit.

The *delivered price* (retail price) offered to each customer who receives its demand at its location, is the sum of the mill price (wholesale price) and an additional delivery cost, which is a percentage of the mill price, where the percentage is here called *markup*. Thus, when the mill price is known and fixed, instead of the delivered price, the markup can be considered the unknown element that must be determined by the PM-LRP. To do this, a global set of markup levels is considered, and the demand corresponding to each markup level is assumed to be known for each customer, which may vary for different customers.

The underlying assumptions of the PM-LRP are as follows:

1. A finite number of potential customers and candidate locations for establishing DCs are scattered in a region; their locations are already identified.
2. The distances between any two locations are symmetric and satisfy the triangle inequality.
3. Each customer can be serviced only by one open DC and placed on only one vehicle route, that is, fractional assignment and split delivery is not allowed.
4. Serving all customers are not mandatory. Customers have no priority over the service, and there is no time window for serving any customer.
5. Each DC has a limited distribution capacity, which is known, and the total demand of customers assigned to it must not exceed its capacity.
6. A set of identical vehicles with the same capacity is considered. Each vehicle starts and ends its route at only one DC and can visit any subset of customers whose total demand is less than or equal to the vehicle capacity. Each vehicle can perform at most one route.
7. Vehicles have enough fuel to complete their respective transportation missions and can serve all of the assigned customers without having to re-fuel.
8. Each vehicleبs travel cost is proportional to the length of the path traveled and independent of the vehicle speed, time of the trip, and the weight of the vehicleبs load.
9. A single product is considered, with a known set of markup levels, which are beneficial for the company. The mill price of the product is known and invariable across the network.

10. The spatial pricing policy is followed by the company, under which a finite number of different delivered prices are charged to customers, who receive their demands at their locations.
11. Each customerِs demand is price-sensitive and depends on the markup level considered for delivering the product.
12. The objective is to maximize the overall profit, obtained by subtracting the total cost of establishing DCs, traveling, and purchasing from the total collected revenue gained by selling the purchased products to the customers.

Under the spatial pricing policy, the company offers a specific delivered price to each customer based on its location, so different customers may be charged different markups.

2.2 Notation

2.2.1. Sets

I The set of potential customers

H The set of potential DCs

L The set of markup levels

K_0 The set of available vehicles given by $K_0 = \{v_1, \dots, v_{|K_0|}\}$

2.2.2. Auxiliary sets

In order to model the problem, the following auxiliary sets are also defined:

M The set of all nodes (potential customers and DCs), i.e., $M = I \cup H$

K_h The set of $|K_0|$ virtual vehicles assigned to DC h, $K_h = \{v_1^h, \dots, v_{|K_0|}^h\}$, $h \in H$

K The union of $|H|$ vehicle sets K_h, i.e., $K = \bigcup_{h \in H} K_h$

A The set of all triples (i, j, k) such that vehicle $k \in K$ can travel from node $i \in M$ to node $j \in M$, i.e.,
$A = \{(i, j, k) \colon (i, j \in I, k \in K, i \neq j) \cup (i \in H, j \in I, k \in K_i) \cup \big(i \in I, j \in H, k \in K_j\big)\}$

2.2.3 Parameters

c_{ij} The travel distance between nodes i and j, $i, j \in M$

s The travel cost per unit of distance

Cap^v The vehicle capacity, which is the same for all vehicles

Cap_h^{Dc} The distribution capacity of DC h, $h \in H$

w The mill price per unit of product

p_l The percentage associated with markup level l, $l \in L$

P_l The delivered price per unit of product associated with markup level l, which is computed by $P_l = w(1 + p_l)$, $l \in L$

d_{il} The demand of customer i who is charged markup level l, $i \in I$, $l \in L$

F_h The fixed cost of establishing DC h, $h \in H$

h_k The DC to which that virtual vehicle k is assigned, i.e., $k \in K_{h_k}$, $k \in K$

2.2.4. Decision variables

x_{ijk} A binary variable that becomes 1 if node j is visited immediately after node i by virtual vehicle k, and 0 otherwise, $(i, j, k) \in A$

y_{ikl} A binary variable that equals 1 if node i is selected to be visited by virtual vehicle k at markup level l, and 0 otherwise, $i \in I$, $k \in K$, $l \in L$

t_h A binary variable that takes 1 if DC h is selected to be established, and 0 otherwise, $h \in H$

u_{ik} An auxiliary non-negative variable defined for customer $i \in I$, used in MTZ sub-tour elimination constraints of the route of virtual vehicle k, $i \in I$, $k \in K$

2.3. Polynomial-size formulation

This section presents a mixed-integer linear programming (MILP) model for the proposed PM-LRP. This model can be used only to solve small-sized instances by means of MILP solvers.

As indicated in Section 2.2.2, K_0 denotes the set of $|K_0|$ available identical vehicles, a copy of this set is denoted by K_h that contains identical virtual vehicles assigned to DC h. Let K designate the union of $|H|$ vehicle sets K_h, i.e., $K = \cup_{h \in H} K_h$. Using these additional sets, it is not necessary to define variables for assigning vehicles to DCs, but the total number of selected virtual vehicles must be limited to $|K_0|$ by a constraint in the model. The proposed PM-LRP can now be formulated as the following MILP model:

$$\sum_{k \in K} \sum_{i \in I} \sum_{l \in L} w \times p_l d_{il} \, y_{ikl} - \sum_{h \in H} F_h t_h - \sum_{(i,j,k) \in A} (s \times c_{ij} x_{ijk}) \qquad (1)$$

s.t.

$$\sum_{k \in K} \sum_{l \in L} y_{ikl} \leq 1 \qquad i \in I \qquad (2)$$

$$\sum_{i \in I} x_{hik} - \sum_{i \in I} x_{ihk} = 0 \qquad h \in H, k \in K_h \qquad (3)$$

$$\sum_{i \in I} x_{hik} \leq 1 \qquad h \in H, k \in K_h \qquad (4)$$

$$\sum_{j \in I, j \neq i} x_{ijk} + x_{ih_k k} = \sum_{j \in I, j \neq i} x_{jik} + x_{h_k ik} = \sum_{l \in L} y_{ikl} \qquad i \in I, k \in K \qquad (5)$$

$$\sum_{i\in I}\sum_{l\in L} d_{il}y_{ikl} \leq Cap^{v} \qquad k \in K \tag{6}$$

$$\sum_{i\in I}\sum_{k\in K_h}\sum_{l\in L} d_{il}y_{ikl} \leq Cap_h^{DC}t_h \qquad h \in H \tag{7}$$

$$\sum_{h\in H}\sum_{k\in K_h}\sum_{j\in I} x_{hjk} \leq |K_0| \tag{8}$$

$$1 \leq u_{ik} \leq |I| \qquad i \in I, k \in K \tag{9}$$

$$u_{ik} - u_{jk} + |I|x_{ijk} \leq |I| - 1 \qquad i,j \in I, k \in K, i \neq j \tag{10}$$

$$x_{ijk} \in \{0,1\} \qquad (i,j,k) \in A \tag{11}$$

$$y_{ikl} \in \{0,1\} \qquad i \in I, k \in K, l \in L \tag{12}$$

$$t_h \in \{0,1\} \qquad h \in H \tag{13}$$

The objective (1) is to maximize the total profit, which is the total profit of serving customers, minus the total cost of establishing DCs and transportation. Note that the total profit of serving customers is the total revenue earned by serving them, minus the purchasing cost of the delivered products, i.e.,

$$\sum_{k\in K}\sum_{i\in I}\sum_{l\in L} w(1+p_l)d_{il}\,y_{ikl} - \sum_{k\in K}\sum_{i\in I}\sum_{l\in L} wd_{il}\,y_{ikl} = \sum_{k\in K}\sum_{i\in I}\sum_{l\in L} wp_l d_{il}\,y_{ikl}.$$

Constraints (2) ensure that each customer can be visited at most once. Constraints (3) guarantee that the number of times each vehicle enters a DC is equal to the number of times it leaves it. Constraints (4) ensure that each virtual vehicle can cover at most one route. Constraints (5) imply the connectivity of each route while determining the assignment of customers to virtual vehicles. Constraints (6) are the capacity constraints for the virtual vehicles. Constraints (7) guarantee that the total customer demand served from a single DC does not exceed its capacity. Constraint (8) limits the total number of virtual vehicles used. Constraints (9) and (10) are Miller-Tucker-Zemlin (MTZ) sub-tour elimination constraints introduced in Miller et al. (1960). Note that auxiliary variables u_{ik} $i \in I, k \in K$ are continuous variables. Finally, constraints (11), (12), and (13) force the integrality of the decision variables.

The above model has a polynomial number of variables and constraints, but state-of-the-art MILP solvers such as CPLEX require several hours to solve small-size instances (see Section 4.3 for numerical evaluation of this model). In the next section, an exponential-size reformulation of this model is given, which can be solved in large scales using a branch-and-price algorithm proposed in Section 3.

2.4. Set-packing formulation

In order to solve the problem to optimality for large-size instances, this subsection reformulates the model presented in Section 2.3 as a set-packing formulation using the Dantzig-Wolfe decomposition method (Dantzig & Wolfe, 1960). The new model enables us to develop an efficient branch-and-price algorithm in the next section. To do so, the constraints of the model given in Section 2.3 must be decomposed into a number of independent and connecting constraints in order to construct one master problem and some subproblems. Here, constraints (2), (7), and (8) are considered the connecting constraints used to define the master problem, and constraints (3-6) and (9-12) are the independent constraints.

In the following a *route-price* is defined as any route starting and ending at the same DC and all the prices offered to the customers on that route. Let R be the set of all feasible route-prices, and let R_h be the set of all feasible route-prices, starting and ending at DC $h \in H$. Let us define the following additional notation:

z_r A binary variable that becomes 1 as route-price r in R is selected, and 0 otherwise, $r \in R$

α_{ilr} A binary parameter whose value is 1 if customer i is placed on route-price r and served at markup level l, and 0 otherwise, $r \in R, i \in I\ l \in L$

δ_{ijr} A binary parameter that takes 1 if node j is visited immediately after node i in route-price r, and 0 otherwise, $i, j \in M, r \in R$.

Then, we have the following identities:

$$\sum_{k \in K_h} y_{ikl} = \sum_{r \in R_h} \alpha_{ilr} z_r \qquad h \in H\ i \in I, l \in L \tag{14}$$

$$\sum_{k:\,(i,j,k) \in A} x_{ijk} = \sum_{r \in R} \delta_{ijr} z_r \qquad i, j \in M, \tag{15}$$

which imply

$$\sum_{k \in K_h} x_{hjk} = \sum_{r \in R} \delta_{hjr} z_r, \sum_{k \in K} y_{ikl} = \sum_{r \in R} \alpha_{ilr} z_r$$

$$\sum_{h \in H} \sum_{k \in K_h} \sum_{j \in I} x_{hjk} = \sum_{h \in H} \sum_{j \in I} \sum_{k \in K_h} x_{hjk} = \sum_{r \in R} \left(\sum_{h \in H} \sum_{j \in I} \delta_{hjr} \right) z_r = \sum_{r \in R} z_r$$

$$\sum_{k \in K} \sum_{i \in I} \sum_{l \in L} w \times p_l d_{il}\, y_{ikl} - \sum_{(i,j,k) \in A} \left(s \times c_{ij} x_{ijk} \right)$$

$$= \sum_{i \in I} \sum_{l \in L} w \times p_l d_{il} \left(\sum_{k \in K} y_{ikl} \right) - \sum_{i \in M} \sum_{j \in M} s \times c_{ij} \left(\sum_{k:(i,j,k) \in A} x_{ijk} \right) = \sum_{r \in R} b_r\, z_r$$

where b_r denotes the profit of each route-price $r \in R$, given by

$$b_r = \sum_{i \in I} \sum_{l \in L} \alpha_{ilr} (w p_l d_{il}) - \sum_{i \in M} \sum_{j \in M} \delta_{ijr} s c_{ij}. \tag{16}$$

Hence, by substitution of $\sum_{k \in K} y_{ikl}$ and $\sum_{k:(i,j,k) \in A} x_{ijk}$ from (14) and (15) into the connecting constraints (2), (7), and (8), model (1)-(13) can be reformulated as the following set-packing formulation:

$$Max \sum_{r\in R} b_r\, z_r - \sum_{h\in H} F_h\, t_h \tag{17}$$

s.t.

$$\sum_{r\in R} z_r \left(\sum_{l\in L} \alpha_{ilr}\right) \leq 1 \qquad i\in I \tag{18}$$

$$\sum_{r\in R_h} \left(\sum_{i\in I}\sum_{l\in L} \alpha_{ilr} d_{il}\right) z_r \leq Cap_h^{DC} t_h \qquad h\in H \tag{19}$$

$$\sum_{r\in R_h} z_r \left(\sum_{l\in L} \alpha_{ilr}\right) \leq t_h \qquad h\in H, i\in I \tag{20}$$

$$\sum_{r\in R} z_r \leq |K_0| \tag{21}$$

$$z_r \in \{0,1\} \qquad r\in R \tag{22}$$

$$t_h \in \{0,1\} \qquad h\in H \tag{23}$$

The objective (17) is to maximize the difference between the total collected profit and the total DC establishment cost. Constraints (18) ensure that all customers can be visited at most once. Constraints (19) guarantee that the total customer demand served by the same DC does not exceed its capacity. Constraints (20) are valid inequalities that are added to strengthen the formulation (see Section 4.2 for numerical effectiveness of these inequalities). Constraint (21) ensures that at most $|K_0|$ vehicles are available. Constraints (22) and (23) impose integrality restrictions on the decision variables.

As the number of route-price variables z_r in model (17)-(23) exponentially grows with the problem-instance size, the model cannot directly be solved using MILP solvers. The next section exploits the special structure of this model and presents an algorithm to efficiently solve it. Finally, note that we can generalize the above model to the case where vehicles are not homogeneous following the same method explained above.

3. Branch-and-Price algorithm

This section develops a Branch-and-Price (B&P) algorithm, in which the column generation method is embedded within a Branch-and-Bound (B&B) algorithm, to solve the model (17)-(23).

From now on, the model (17)-(23) and its continuous relaxation are called the Master Problem (MP) and the Linear Master Problem (LMP), respectively. In our B&P algorithm developed for the MP, at each node a Restricted Master Problem (RMP) containing a small subset of variables is first considered, and then its relaxation, called Restricted Linear Master Problem (RLMP), is solved to optimality using an LP solver. Next, a pricing problem is defined based on the resulting solution and its corresponding optimal dual solution. By solving the pricing problem, a set of new *columns* with positive *reduced costs* are iteratively *generated* and added to the current RLMP, and the new RLMP is solved again until no column with positive reduced cost exists (note that our problem has a maximization objective and the reduced cost is a quantity defined for each non-basic variable in the context of the simplex method; see Definition 3.2 in Bertsimas and Tsitsiklis (1997)). If the resulting solution of the RLMP, which is also an optimal solution to LMP, is fractional, some *branching rules* are applied to generate new *child nodes*, where the same column-generation procedure is used until an optimal integer solution is found.

The RLMP corresponding to our model is formulated as follows:

RLMP:

$$\max \sum_{r\in R^*} b_r\, z_r - \sum_{h\in H} F_h\, t_h \tag{24}$$

s.t.

$$\sum_{r\in R^*} z_r \left(\sum_{l\in L} \alpha_{ilr}\right) \le 1 \qquad i \in I \tag{25}$$

$$\sum_{r\in R_h^*} \left(\sum_{i\in I}\sum_{l\in L} \alpha_{ilr} d_{il}\right) z_r \le Cap_h^{DC} t_h \qquad h \in H \tag{26}$$

$$\sum_{r\in R_h^*} z_r \left(\sum_{l\in L} \alpha_{ilr}\right) \le t_h \qquad h \in H, i \in I \tag{27}$$

$$\sum_{r\in R^*} z_r \le |K_0| \tag{28}$$

$$0 \le z_r \qquad r \in R^* \tag{29}$$

$$0 \le t_h \le 1 \qquad h \in H \tag{30}$$

where R^* is the set of initially selected columns, which is dynamically updated in the next iterations of the B&P algorithm. The set R_h^* denotes the subset of columns in R^* whose corresponding routes start at DC h , $h \in H$. Note that the upper bound ﺑ1ﭖ provided by the continuous relaxation of constraints (22) is dominated by constraints (18), so it is eliminated in the RLMP formulation. As a consequence, basic columns of the RLMP solution which are set to ﺑ1 ﭖare not generated again in the pricing problem and the set R^* will be free of any duplicate member.

The master routine of our B&P algorithm is as follows:

Step BP.1. Initialization of B&B tree.

Create a root node. Generate all *two-length columns*, which are defined as any combination of a route with two arcs to serve a single customer and any price that can be offered to the customer. Include the generated columns into R^*.

Step BP.2. Column generation.

Step CG.1. Initialization of column generation. Include all columns previously generated in R^*, except for columns that are infeasible with respect to the branching constraints of the current node. In each node, initialize the *active set* as the set of all DCs apart from those DCs that cannot be established according to restrictions imposed by the branching constraints of the current node, i.e., their corresponding variables must be set to zeroes (the active set is only used in CG.2). In the root node, set the parameter $\bar{a}$ to $\lceil n/8 \rceil$ for all DCs (n denotes the number of customers, $|I|$), while in the other nodes, set the parameter $\bar{a}$ to $\lceil n/4 \rceil$ for all DCs (the parameter $\bar{a}$ will be defined and used only in the heuristic label-setting algorithm given in Section 3.1.2; hence, it appears only in CG.2). Moreover, consider the *critical sets* for all DCs to be empty (critical sets will be defined and used only in the exact label-setting algorithm given in Section 3.1.1; hence, they are considered only in CG.3).

Step CG.2. Generate columns using the heuristic label-setting algorithm.

Step CG.2.1. For each DC that is a member of the active set, using the heuristic label-setting algorithm developed in Section 3.1.2, generate columns with positive reduced costs. Then, add these columns to the set R^*, and solve the updated RLMP (note that both of the heuristic and exact label-setting algorithms generate columns that are feasible with respect to the branching restrictions as these constraints are incorporated into the pricing problem before column generation).

Step CG.2.2. From the active set, remove DCs for which 1) no column has been generated in the last try and 2) their associated parameter $\bar{a}$ has reached to n. Then, double the parameter $\bar{a}$ for those DCs existing in the active set for which 1) no column has been generated in the last try and 2) the parameter $\bar{a}$ is less than n. Go to Step CG.3 if the active set is empty; otherwise go back to Step CG.2.1.

Step CG.3. Generate columns using the exact label-setting algorithm. List all DCs in descending order based on the number of columns generated in the last try. In the ordered DC list, find the first DC for which at least one column can be generated using the exact label-setting algorithm proposed in Section 3.1.1 (when two DCs have the same priority in the list, randomly select one of them). Then, add the newly generated column(s) to the set R^*, solve the updated RLMP, and update the critical set associated with the selected DC. If no such DC can be found, go Step BP.3; otherwise, repeat this step from the beginning.

Step BP.3. Deciding the current node. In the root node, solve the RMP with the present columns in R^* and update the *best lower bound*. At all nodes, if the resulting solution of the RLMP is an integer, fathom the node, save the solution, and update the best lower bound. If the RLMP is infeasible, fathom the node. If the resulting solution of the RLMP is fractional, classify the node as *unexplored* node and save its corresponding upper bound. If the parent node of the current node has another unsolved child node, select it as the current node, and go to Step BP.2 to solve it using column generation. Otherwise, go to the next step.

Step BP.4. Exploring the B&B tree. Choose an unexplored node with the largest upper bound among all unexplored nodes. Then, generate new branching nodes based on the rules and priorities determined in Section 3.2. Randomly select one of its two branching nodes, select it as the current node, and go to Step BP.2 to solve it using column generation. Stop whenever all nodes of the tree are fathomed, and return the integer solution that corresponds to the best lower bound.

The heuristic and exact label-setting algorithms used to solve our pricing problem in Step BP.2 are given in Sections 3.1.1 and 3.1.2, and the branching rules used in Step BP.4. are given in Section 3.2.

3.1 Pricing problem for generating columns

Our *pricing problem* for generating columns with positive reduced costs after solving the current RLMP is decomposable into a set of independent *pricing subproblems*, one for each DC. To present these subproblems, let μ_i, π_h, σ_{ih}, and ω_0 be the non-negative dual variables associated to the inequality constraints of LMP: (18), (19), (20), and (21), respectively. Then, the reduced cost of variable z_r, $r \epsilon R_h$, in the LMP is equal to

$$\tilde{b}_r = b_r - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}\mu_i - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}d_{il}\pi_h - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}\sigma_{ih} - \omega_0. \tag{31}$$

Thus, the optimality condition for any feasible solution of LMP is given by

$$b_r - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}\mu_i - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}d_{il}\pi_h - \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}\sigma_{ih} - \omega_0 \le 0 \tag{32}$$

for $h \in H, r \in R_h$. By multiplying both sides of (32) by -1 and substituting b_r from (16) into (32), the optimality condition can be stated as

$$\sum_{i\in M_h}\sum_{i\in M_h}\delta_{ijr}s \times c_{ij} + \sum_{i\in I}\sum_{l\in L}\alpha_{ilr}(\mu_i + d_{il}\pi_h + \sigma_{ih} - wp_l d_{il}) + \omega_0 \ge 0 \tag{33}$$

for $h \in H, r \in R_h$. Therefore, given the values of μ_i, π_h, σ_{ih}, and ω_0 by solving RLMP, the pricing subproblem corresponding with DC $h \in H$ is presented as follows:

Pricing subproblem associated with DC h:

$$\min \sum_{i\in M_h}\sum_{j\in M_h} x_{ij}s \times c_{ij} + \sum_{i\in I}\sum_{l\in L}(\mu_i + d_{il}\pi_h + \sigma_{ih} - wp_l d_{il})y_{il} + \omega_0 \tag{34}$$

s.t.

$$\sum_{j\in I} x_{hj} = 1 \tag{35}$$

$$\sum_{j\in M_h} x_{ij} = \sum_{j\in M_h} x_{ji} = \sum_{l\in L} y_{il} \qquad i \in I \tag{36}$$

$$\sum_{l\in L} y_{il} \le 1 \qquad i \in I \tag{37}$$

$$\sum_{i\in I}\sum_{l\in L} d_{il} y_{il} \le Cap^{v} \tag{38}$$

$$1 \le u_i \le |I| \qquad i \in I \tag{39}$$

$$u_i - u_j + |I| x_{ij} \le |I| - 1 \qquad i, j \in I, i \ne j \tag{40}$$

$$x_{ij} \in \{0,1\} \qquad i, j \in M_h \tag{41}$$

$$y_{il} \in \{0,1\} \qquad i \in I, l \in L \tag{42}$$

where M_h is designated as a set including DC $h \in H$ and all potential customers, i.e., $M_h = I \cup \{h\}$.

The next two subsections propose the exact and heuristic label-setting algorithms used here for solving the pricing subproblem corresponding with each DC. These algorithms are applied in Steps CG.2 and CG.3 of our B&P algorithm explained earlier in this section.

3.1.1 Exact label-setting algorithm for solving pricing subproblems

The pricing subproblem corresponding with DC $h \in H$, i.e., (34)–(42), is actually an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), which is called here ESPPRC with Price-sensitive Demands (ESPPRC-PD). The ESPPRC-PD simultaneously determines the best elementary path and pricing decisions. In the last years, this has become more and more common to tackle ESPPRCs by means of labeling algorithms developed based on dynamic programming, in column-generation approaches to capacitated routing problems. Here, we similarly solve our ESPPRC-PD (34)–(42) using a label-setting algorithm that extends the ones presented in Feillet et al. (2004) and Righini and Salani (2006, 2008).

This algorithm assigns *labels* to each vertex, which save the required information about (partial) paths from the start node h to that vertex. The labels of each vertex are extended along the outgoing arcs of that vertex (if it is feasible) to generate new labels. This operation is continued until all labels are extended in all feasible ways or they are fathomed by some rules.

In order to improve the efficiency of the label-setting algorithm, we use the Decremental State Space Relaxation (DSSR) acceleration method introduced by Righini and Salani (2008), which is enhanced by the 2-cycle elimination procedure proposed by Houck et al. (1980). Let us briefly describe the idea used in the DSSR method. An *elementary* path satisfies the *elementary condition*, that is, each node cannot be visited more than once. A relaxation of the ESPPRC in which the path may visit some nodes more than once is called Shortest Path Problem with Resource Constraints (SPPRC). In this relaxation, stronger dominance rules can be used such that the SPPRC is solvable in pseudo-polynomial time (Irnich & Desaulniers, 2005). Using this fact, in the DSSR method, the elementary condition, which is necessary for a feasible path in the ESPPRC, is relaxed at the beginning, and the corresponding SPPRC is solved. If the optimal path obtained by solving the SPPRC is elementary, the search procedure is terminated. Otherwise, the customers on the optimal path that are visited twice or more will be added to a set, which is called the *critical set* (for each DC, one critical set is used). The search procedure is continued while the elementary condition is checked for customers contained in the critical set henceforth.

There are two policies for label generation: *mono-directional* (Desrochers, 1988; Feillet et al., 2004) and *bounded bidirectional* (Ahuja et al., 1993; Righini & Salani, 2006). We will use both policies and evaluate their performance.

A detailed description of the steps of our DSSR-based label-setting algorithm for solving the ESPPRC corresponding to DC h under the mono-directional label-generation policy is provided in the following. The changes required under the bounded bidirectional policy will be discussed after.

Step LS.1. Graph construction: For our ESPPRC-PD, based on the instructions explained below, construct a graph where for each customer in I, a set of $|L|$ vertexes are defined, one for each markup level. The set of these $|L|$ virtual vertexes associated with customer i is called *cluster* i, and each vertex l in cluster i is denoted by $i.l$. Moreover, a single vertex for DC h is defined. Each vertex $i.l$ is weighted by demand d_{il}, and vertex h is weighted by 0. Between any two clusters in the graph, all $|L| \times |L|$ possible arcs are considered. Also, between vertex h and each cluster, all $|L|$ possible arcs are considered. Figure 1 schematically indicates the arcs for an instance with $|I| = 3$ and $|L| = 3$. The arcs are also weighted in order to calculate the opposites of reduced costs. Let $c_{i.l}^{j.k}$ be the weight of the arc outgoing from vertex $i.l.$ to vertex $j.k.$, and $c_h^{i.l}$ ($c_{i.l}^h$) be the arc weight of the arc outgoing from vertex h ($i.l$) to vertex $i.l$ (h). These arc weights are defined as follows:

$$c_{i.l}^{j.k} = s \times c_{ij} + d_{jk}\pi_h + \sigma_{jh} + \mu_j - wp_k d_{jk} \quad (43)$$

$$c_{i.l}^{h} = s \times c_{ih} + \omega_0, c_h^{i.l} = s \times c_{hi} \quad (44)$$

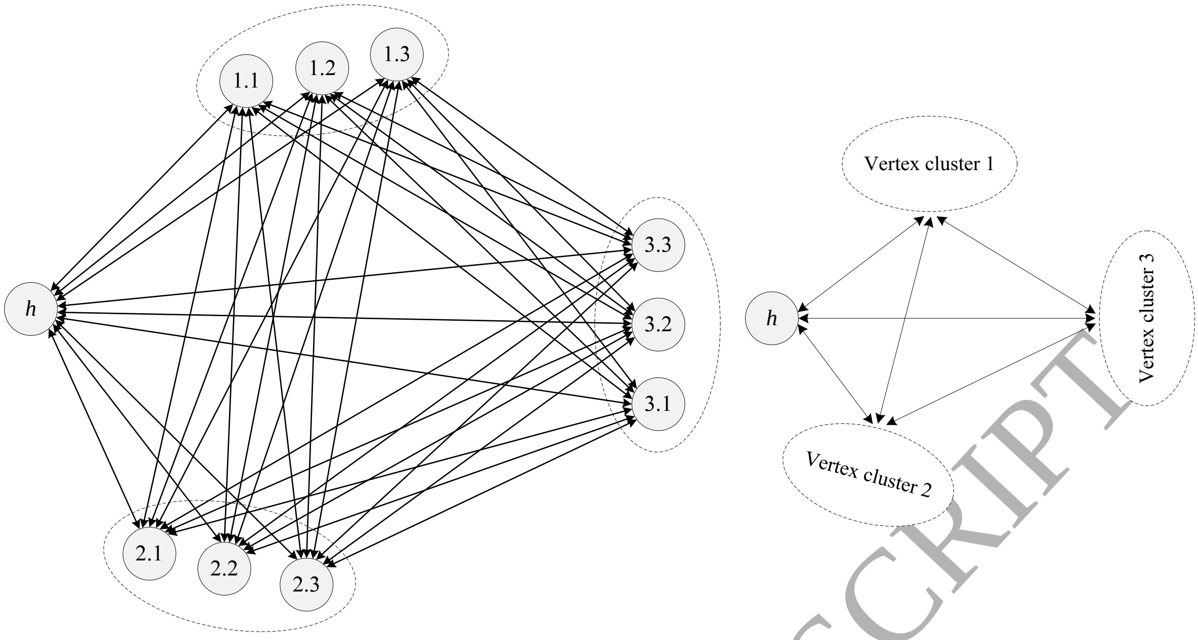

Figure 1. An illustrative example of the graph for an instance of the ESPPRC with $|I| = 3$ and $|L| = 3$.

If there is a vertex in a cluster with less demand delivery and a greater profit in comparison to another vertex in the same cluster, remove the latter from the graph (because the corresponding price cannot appear in an optimal solution).

Step LS.2. Starting label generation: Start any feasible elementary path from vertex h and gradually develop it to other vertexes based on the instructions explained below, such that the path finally returns to vertex h, and such that for those customers in the critical set Cr_h at most one vertex from its corresponding cluster is visited (the critical set Cr_h is first set to empty and then will be updated during Step B.P.2). As a partial path gets to a vertex, one label is created. For a partial path getting to vertex $i.l$ (or vertex h at the end), label $(\boldsymbol{E}, S, C, D, i, l)$ (or $(\boldsymbol{E}, S, C, D, h, .)$) is constructed. In this label, $\boldsymbol{E}$ is the set of clusters visited by the path, S is the total number of times that the path visits the clusters in $\boldsymbol{E}$ (for an elementary path it is equal to the number of vertexes in $\boldsymbol{E}$), C denotes the total arc weight of the path, and D is the total vertex weight of the corresponding path.

Create the first label for vertex h as $(\varphi, 0, 0\ ,0, h, .)$ and set the control parameter T to 0. A label is called *untreated* if it has not been used yet by the label-setting algorithm to develop a new label. Note that when a label is extended, it becomes *treated* and it is not considered anymore in the next iterations. In fact, by extending a label, a new *untreated* label is created.

Step LS.3. Label selection: Find all of the untreated labels with parameter $D = T$, and then sort them in ascending order of C. Then, apply Step L.S.4 and Step L.S.5 below for each one of these labels according to the determined order.

Step LS.4. Label extension: Extend each label $(\boldsymbol{E}, S, C, D, i, l)$ found in Step L.S.3 along the outgoing arcs of its corresponding vertex $i.l$ such that the next vertex $j.k$ satisfies $j \notin (Cr \cap E) \cup \{pred(i.l)\}$ where $pred(i.l)$ denotes the predecessor of vertex $i.l$ on the partial path corresponding to the label (the two-cycle elimination rule). When an untreated label $(\boldsymbol{E}, S, C, D, i, l)$ is extended to a new label whose corresponding vertex is $j.k$ (h), define the new label as $(\boldsymbol{E}', S', C', D', j, k)$ (or $(\boldsymbol{E}', S', C', D', h, .)$) with

$$\boldsymbol{E}' = \boldsymbol{E} \cup \{j\}$$

$$S' = S + 1$$

$$D' = D + d_{jk}$$

$$C' = \begin{cases} C + c_{i.l}^{j.k} & j \neq h \\ C + c_{i.l}^{h} & j = h \end{cases}.$$

Step LS.5. Label elimination and storage: Maintain a newly generated label $(\boldsymbol{E}, S, C, D, i, l)$, unless it can be eliminated by one of the following rules:

i) Feasibility rule. Eliminate the label if the resource constraint $D \leq Cap^v$ is violated (this constraint guarantees that the total demand of the corresponding route satisfies the vehicle capacity).

ii) Bounding rule. Eliminate the label if the following condition holds:

$$LB = C - w \times \max_{l \in L}\{p_l\} \times (Cap^v - D) + c_{i.l}^{h} + \omega_0 > 0.$$

(The quantity LB is a lower bound for the total arc weight of any complete path that can be extended from the current label $(\boldsymbol{E}, S, C, D, i, l)$, which corresponds to the ideal case where the remaining capacity of the vehicle can be sold at the maximum price and the path can directly return to DC h to have the minimum travel cost; note that the distances satisfy the triangle inequality, and for any $j \in I, k \in L$ we have $\mu_j + d_{jk}\pi_h + \sigma_{jh} \geq 0$ as dual variables μ_j, π_h, and σ_{jh} are all non-negative).

iii) Dominance rule. Eliminate the label if there is another label $(\boldsymbol{E}', S', C', D', i, k)$ with the following properties:

$$i)\boldsymbol{E}' \cap Cr_h \subseteq \boldsymbol{E}^* \cap Cr_h$$

$$ii)S' \leq S$$

$$iii)D' \leq D$$

$$iv)C' \leq C$$

$$v)pred(i.k) \in \boldsymbol{E}^*$$

where at least one of the first four relations must be held strictly. The set $\boldsymbol{E}^*$ denotes the union of the set $\boldsymbol{E}$ and the set of unreachable clusters for the label, where a cluster is considered *unreachable* for the label if the vehicle capacity constraint is certainly violated by visiting the cluster at any price level (the reason is that if such a label $(\boldsymbol{E}', S', C', D', i, k)$ exists, then extending it results in better columns).

Step LS.6. Stopping label generation: Set the parameter T to the minimum value of the total demands of the paths corresponding to the untreated labels. If there is no untreated label or parameter T reaches Cap^v, include all labels $(\boldsymbol{E}, S, C, D, i, k)$ with $i = h$ and $C < 0$ into the set $\Gamma(h)$ (recall that for each completed label $(\boldsymbol{E}, S, C, D, h, k)$, the quantity C is the opposite of the reduce cost of the column determined by that label). Otherwise, go back to Step LS.3.

Step LS.7. Updating the critical set: Among the labels in the set $\Gamma(h)$, find the label $(\boldsymbol{E}, S, C, D, h, .)$ with minimum C. If this label does not determine an elementary path, find those customers on the path that are visited more than once and add them to the critical set Cr_h, and go back to Step LS.2. Otherwise, terminate the algorithm and return columns associated with the labels in $\Gamma(h)$ whose corresponding paths are elementary.

In the above algorithm, the mono-directional label-generation policy is used. Under the bounded bidirectional label-generation policy, labels are extended both forwardly from the initial vertex and backwardly from the terminal vertex unless half of a critical resource with monotonic consumption along paths has been consumed (in our case, the vehicle capacity is considered the critical resource). Then, full paths are built by joining pairs of forward and backward labels. This policy may reduce the computational effort required for label generation. Now we describe how our algorithm can be adapted under this policy.

In each one of our pricing subproblems, source and destination vertexes are the same DC h, and traveling costs are symmetric. Thus, there is no need to backwardly extend labels from the destination vertex, and we can only merge each pair of forward labels satisfying *feasibility* and *uniqueness* conditions, explained below, to generate all columns with positive reduced cost.

The feasibility condition checks that the sum of the demands on a merged route is not greater than the vehicle capacity, and that the vertexes in the critical sets are not located on the routes corresponding to the both labels. The uniqueness condition checks that a merged route is saved only once. To algorithmically control this, two labels are merged if the absolute deviation between the total demands of their corresponding half routes cannot be reduced, by two new routes obtained by excluding the last vertex from one route and attaching it to the other route (for more details see Righini and Salani (2006)).

Steps L.S.1-L.S.5 and L.S.7 of the label-setting algorithm under the bounded bidirectional policy remain intact, but Step L.S.6 must be modified as follows:

Step LS.6 (under bounded bidirectional label-generation policy). Set the parameter T to the minimum value of the total demands of the paths corresponding to untreated labels. If there is no untreated label or parameter T is strictly greater than $Cap^v/2$, create new labels by joining each pair of forward labels that satisfy both feasibility and uniqueness conditions, and include all new labels $(\boldsymbol{E}, S, C, D, i, .)$ with $i = h$ and $C < 0$ into the set $\Gamma(h)$. Otherwise, go back to Step L.S.3.

3.1.2 Heuristic label-setting algorithm for solving pricing subproblems

To increase the speed of the column-generation algorithm, the ESPPRC problem can first be solved heuristically by considering a subgraph of the graph constructed in Step L.S.1 to decrease its size. Let us now explain our heuristic in the following. This heuristic is iteratively used in Step CG.2 of our B&P algorithm.

Assume that $\bar{a} \leq n$ is a given positive integer defined for each DC h. The heuristic label-setting algorithm searches for columns with positive reduced costs in a smaller graph in which each vertex has at most $\bar{a} \times |L|$ outgoing arcs. Figure 2 presents how this graph reduction can be applied for the illustrative example depicted in Figure 1 for $\bar{a} = 2$. When running Step CG.2, the parameter $\bar{a}$ is here set to a greater value, if no column with positive reduced cost can be generated by the pricing subproblem associated with DC h. This reduction can accelerate generating some columns with positive reduced costs in the first iterations.

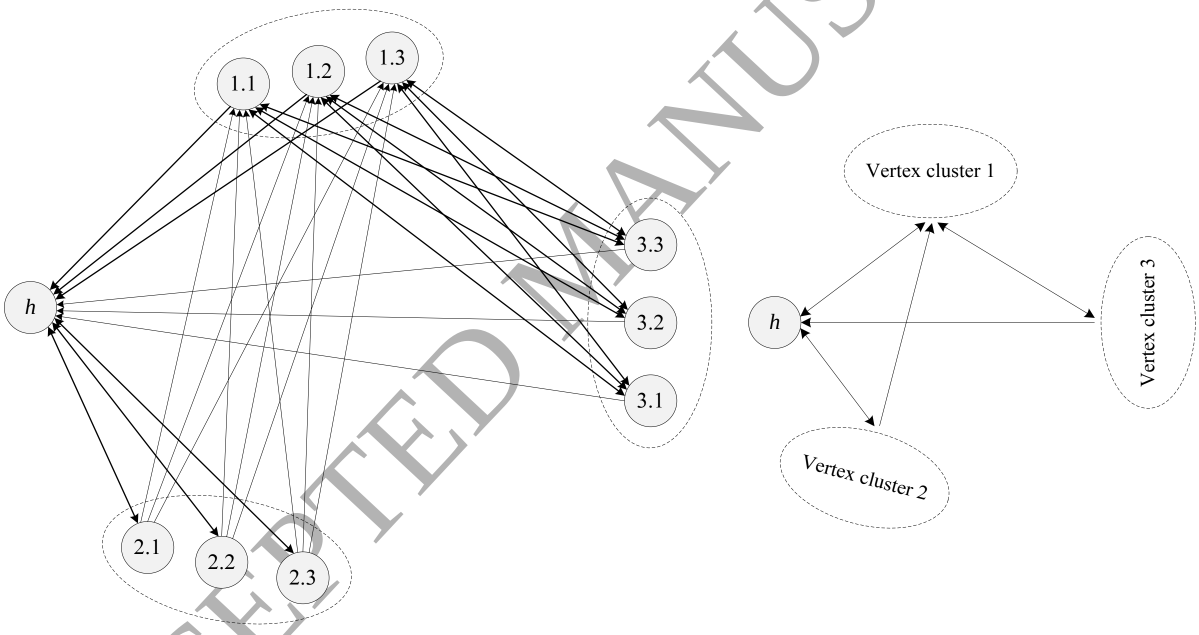

Figure 2. A graph reduction for the example given in Figure 1 for $\bar{a} = 2$.

In the heuristic label-setting algorithm, we do not use the DSSR method, that is, we do not generate any labels that correspond to non-elementary partial paths, so the critical sets include all customers in the heuristic. Moreover, the first and last conditions of dominance rule are relaxed. As a consequence of these relaxations, a greater number of labels are fathomed, but we may lose the guarantee of solving the ESPPRC to optimality.

Steps L.S.2, L.S.3, L.S.6, and L.S.7 of the exact label-setting algorithm similarly exist in the heuristic label-setting algorithm after setting each critical set to the set of all customers (note that under bounded bidirectional label-generation policy, Step L.S.6 must be modified along the lines mentioned above). The other steps require modifications. Step L.S.1 is adjusted as follows:

Step L.S.1 (in heuristic label-setting algorithm). Apply Step L.S.1 of the exact label-setting algorithm. Additionally, for each cluster i $\in I$, find $\bar{\alpha}$ clusters with the least c_{ij}^{min} values, keep their arcs outgoing to cluster i, and remove the other arcs, where $c_{ij}^{min} = \min_{l,k\in L}\{c_{i.l}^{j.k}\}$. Note that the parameter $\bar{\alpha}$ is updated during the column-generation procedure (see Step CG.2.2).

Moreover, only the first sentence of Step L.S.4 is changed to

> بExtend each label $(\boldsymbol{E}, S, C, D, i, l)$ selected in Step L.S.3 along the outgoing arcs of its corresponding vertex $i.l$ such that the next vertex $j.k$ satisfies $j \notin \boldsymbol{E}$.پ,

and the rest remains the same. In Step L.S.5, the conditions checked by the dominance rule are restricted to

$$ii)\; S' \leq S$$

$$iii)\; D' \leq D$$

$$iv)\; C' \leq C.$$

3.2 Branching rules

If the optimal solution of the LMP (which is found by solving the current RLMP) is fractional, a branching decision is required. Let $\hat{z}$ be the optimal fractional solution of the current RLMP, and le.t R^* stand for the set of columns that have already been considered in the current RLMP. Then, the following branching scheme with six hierarchical levels is applied to create two new child nodes:

BR-1. If there is a DC $h \in H$ with a fractional variable t_h, apply the dichotomy branching by enforcing $t_h = 1$ on one branch and $t_h = 0$ on the other.

Explanation of BR-1. In one branch, enforcing $t_h = 1$ does not change the pricing subproblems since they are independent of t_h. In the other branch, we force DC h to be closed, so there is no need to solve the pricing subproblem corresponding to DC h.

BR-2. If there is a DC $h \in H$ whose number of vehicles, denoted by $\hat{n}_h$, is fractional, then consider the constraint $n_h \leq \lfloor \hat{n}_h \rfloor$ on one branch, and $n_h \geq \lceil \hat{n}_h \rceil$ on the other, where n_h is a non-negative variable that satisfies the constraint

$$n_h = \sum_{r\in R_h^*} \hat{z}_r. \tag{45}$$

Explanation of BR-2. To add each of the above branching constraints such that the pricing problem is not affected, we need to impose the constraint using an auxiliary variable n_h (Appleget & Wood, 2000). Let

β_h be the dual variable associated with the constraint (45) for DC h. Then, the objective function of the pricing subproblem corresponding to DC h is updated as follows:

$$\sum_{i\in M_h}\sum_{j\in M_h} sc_{ij}\,x_{ij} + \sum_{i\in I}\sum_{l\in L}(\mu_i + d_{il}\pi_h + \sigma_{ih} - wp_l d_{il})y_{il} + \omega_0 + \beta_h \qquad (46)$$

Since β_h is independent of the customers on the route, the pricing-subproblem structure remains unchanged. Moreover, for the pricing subproblem corresponding to DC h, the quantity LB used in the dominance rule in Step LS.5 and the weight of the arc outgoing from vertex $i.l$ to vertex h are modified by for any $i \in I, l \in L$

$$LB = C - w \times \max_{l\in L}\{p_l\} \times (Cap^v - D) + c_{i.l}^h + \omega_0 + \beta_h. \qquad (47)$$

$$c_{i.l}^h = s \times c_{ih} + \omega_0 + \beta_h. \qquad (48)$$

BR-3. If there is a customer $o \in I$ that is served partially, i.e., the value of $\sum_{r\in R^*}(\sum_{l\in L}\alpha_{olr}\,\hat{z}_r)$ is fractional, first remove the constraint (25) corresponding to customer o, and instead consider $\sum_{r\in R^*} z_r(\sum_{l\in L}\alpha_{olr}) = 1$ on one branch and $\sum_{r\in R^*} z_r(\sum_{l\in L}\alpha_{olr}) = 0$ on the other.

Explanation of BR-3. Since in the child node, the inequality (25) corresponding to customer o is replaced by an equality constraint, its associated dual variable μ_o can become negative, so in the pricing subproblem corresponding to *any* DC $h \in H$, the quantity LB used in the dominance rule in Step LS.5 is given by for any $i \in I, l \in L$

$$LB = C - w \times \max_{l\in L}\{p_l\} \times (Cap^v - D) + c_{i.l}^h + \omega_0 + min\{0, \mu_o\}. \qquad (49)$$

In addition, we remove the cluster corresponding to customer o in the second child node.

BR-4. If there is a customer $o \in I$ that is served partially by a DC $h \in H$, i.e., the value of $\sum_{r\in R_h^*}(\sum_{l\in L}\alpha_{olr}\,\hat{z}_r)$ is fractional, add the constraint $\sum_{r\in R_h^*} z_r(\sum_{l\in L}\alpha_{olr}) = 1$ on one branch and $\sum_{r\in R_h^*} z_r(\sum_{l\in L}\alpha_{olr}) = 0$ on the other.

Explanation of BR-4. Let μ_{oh} be the dual variable associated with $\sum_{r\in R_h^*} z_r(\sum_{l\in L}\alpha_{olr}) = 1$. In the pricing subproblem corresponding with DC h, we add μ_{oh} to the weight of any arc entering cluster o as follows:

$$c_{i.l}^{o.k} = s \times c_{io} + d_{ok}\pi_h + \sigma_{oh} + \mu_o - wp_k d_{ok} + \mu_{oh}, i \in I, l \in L.$$

Dual variable μ_{oh} can become negative, so in the pricing subproblem associated with DC h, quantity LB used in the dominance rule in Step LS.5 is modified for any $i \in I, l \in L$

$$LB = C - w \times \max_{l\in L}\{p_l\} \times (Cap^v - D) + c_{i.l}^h + \omega_0 + min\{0, \mu_{oh}\}. \qquad (50)$$

In the other child node, we delete the cluster corresponding to customer o and its related arcs in the pricing subproblem associated with DC h.

BR-5. If there is a customer $o \in I$ that is served partially with a delivered price $k \in L$, i.e., the value of $\sum_{r\in R^*} z_r\alpha_{okr}$ is fractional, add the constraint $\sum_{r\in R^*} z_r\alpha_{okr} = 1$ on one branch and $\sum_{r\in R^*} z_r\alpha_{okr} = 0$ on the other.

Explanation of BR-5. Let μ_{ok} be the dual variable associated to $\sum_{r\in R^*} z_r\alpha_{okr} = 1$. In the pricing subproblems, we add μ_{ok} to the weight of any arc entering cluster $o.k$ i.e.,

$$c_{i.l}^{o.k} = s \times c_{io} + d_{ok}\pi_h + \sigma_{oh} + \mu_o - wp_k d_{ok} + \mu_{ok}.$$

As dual variable μ_{ok} can become negative, in the pricing subproblem corresponding to *any* DC $h \in H$, the quantity LB used in the dominance rule in Step LS.5 is modified as for any $i \in I, l \in L$

$$LB = C - w \times \max_{l\in L}\{p_l\} \times (Cap^v - D) + c_{i.l}^h + \omega_0 + min\{0, \mu_{ok}\}. \qquad (51)$$

In the second child node, we remove the vertex $o.k$ and its related arcs from the graph used in the pricing subproblem of *any* DC $h \in H$.

BR-6. If the total flow on some arc (i,j) is fractional, i.e., $\sum_{r\in R^*} \delta_{ijr}\hat{z}_r$ is fractional for some $i, j \in I$ (the symbol $(i,j) \in r$ means that the arc (i,j) is a part of the route corresponding to the column r), then consider $\sum_{r\in R^*} \delta_{ijr} z_r = 1$ on one branch and $\sum_{r\in R^*} \delta_{ijr} z_r = 0$ on the other.

Explanation of BR-6. This branching rule was first suggested by Desrochers and Soumis (1989), which is a modified version of the well-known rule given by Ryan and Foster (1981).To keep the structure of the pricing subproblems unchanged, the branching constraint is not explicitly added to the RLMP. For one child node, all existing columns whose corresponding routes include arc $(i,k){:}\, k \neq j$ are deleted from R^*. Then, the constraints (25) corresponding to $i, j \in I$ are removed, and constraints

$$\textstyle\sum_{r\in R^*} z_r\left(\sum_{l\in L} \alpha_{ilr}\right) = 1 \text{ and } \sum_{r\in R^*} z_r\left(\sum_{l\in L} \alpha_{jlr}\right) = 1$$

are added to the RLMP. During the label generation, a label $(\boldsymbol{E}, S, C, D, i, l)$ is extended only along the arcs outgoing from cluster i to cluster j and we delete all the arcs from other clusters $k \neq i$ to cluster j. Observing the fact that the above two constraints are structurally similar to those considered in BR.3, the quantity LB used in the dominance rule in Step LS.5 is modified in a similar way explained in BR.3.

In the other child node, while running the label-generation procedure, we control that a label $(\boldsymbol{E}, S, C, D, i, l)$ is not extended along the arcs outgoing from cluster i to cluster j. Moreover, all existing columns whose corresponding routes include arc (i,j)` are deleted from R^*.

The above branching rules together ensure that solving the model can be continued until all nodes of the B&B tree are fathomed. Moreover, branching rules are presented in the order of priority, and whenever there are multiple candidates for the branching rules at the same priority level, the one whose fractional part is the closest to 0.5 is chosen. Note that modifications forced by branching rules are inherited from the parent nodes across the B&B tree.

4. Computational experiments

This section is organized as follows: Section 4.1 explains how our problem instances (test problems) are generated. Section 4.2 reports computational results of the B&P algorithm under both mono-directional and bounded bidirectional label-generation policies. This subsection also assesses the impact of valid inequalities proposed in Section 2.3. Section 4.3 provides numerical results to compare the polynomial-size model and set-packing formulation. Section 4.4 evaluates the effect of differential pricing. Finally, Section 4.5 analyzes the value of integrating location, pricing and routing decisions.

The proposed algorithm was coded in C++, and RLMPs and the polynomial-size model are solved by CPLEX 12.3. Computations were performed on a computer with a 1.7 GHz CPU and 4GB of RAM, running on a 64-bit Windows operating system.

4.1 Sets of problem instances

As our problem is newly introduced and there is no standard dataset in the literature for it, the performance of the proposed B&P algorithm is evaluated using two set of adapted problem instances. The first set is based on the 14 LRP instances available in Barreto (2003), and the second set is based on 7 instances taken from Archetti et al. (2013) that were created from benchmark instances for the VRP proposed by Christofides et al. (1979).

In both sets, the customersبdemands are modified to be dependent on delivered prices. To do so, the demand of customer i at markup level l is computed by

$$d_{il} = f(P_l) = d_i(a - bP_l)^{1/v}, P_l < a/b \tag{52}$$

where d_i is the demand value for customer i in the original instance, and P_l is the delivered price of the product at markup level l. The quantities a, b, and v are positive parameters of the demand function, which is a negatively sloped function of the delivered price (Greenhut et al., 1975). Depending on the values selected for parameter v, the demand curves would be convex, linear, or concave. Here, the parameters a, b, and v have been set to 10.1, 1.5, and 0.25 in the first set of instances; and 10.74, 1.68, and 0.25 in the second set, respectively. As defined in Section 2.2.3, delivered price P_l is given by $P_l = w(1 + p_l)$, in which w is the final cost (wholesale price) of one product unit, which is here set to \$5; and where p_l is the markup percentage at level l. (markup level l). For the first set of instances, the parameter s varies with the vehicle capacity Cap^v as follows: for $Cap^v \leq 100$, $s = 2$ (except for the cases P06-50×5×6, P06-50×5×11, P06-50×10×6, P06-50×10×11, P07-50×5×6, P07-50×5×11, P07-50×10×6 and P07-50×10×11 where s is set to 1.25 to have non-trivial solutions); for $100 < Cap^v \leq 1000$, $s = Cap^v/80$; for $1000 < Cap^v$, $s = Cap^v/240$.

For each original instance in both data sets, two instances with 6 and 11 markup levels ranging from 0.1 to 0.2 are considered in the interval, i.e., 0.1, 0.12,ه,0.2 and 0.1, 0.11,ه,0. 2. We need more adjustments to make the instances used by Archetti et al. (2013) applicable for our problem because they are used to test a

VRP with single depot, not an LRP. For each new instance, we consider four available vehicles with capacity 100, 5 and 10 DCs where their locations are determined randomly. The capacity and fixed cost of establishment of each DC are set to 200 and 40, respectively. Tables 1 and 2 summarize the metadata for the two adjusted benchmark sets used here. The full data of all instances are freely available at the link: http://bit.ly/EJOR-D-15-0159-data

Table 1. The metadata of the first set of problem instances generated based on Barreto (2003).

Instance name	Name of original instance	# of Customers $\lvert I\rvert$	# of DCs $\lvert H\rvert$	# of Markup levels $\lvert L\rvert$	DC Capacity Cap_h^{DC}	Fixed cost of establishment DCs	Vehicle Capacity Cap^v
C-50×5×6 C-50×5×11	Christofides69 -50×5	50	5	6 11	10000	40	160
C-75×10×6 C-75×10×11	Christofides69 -75×10	75	10	6 11	10000	40	140
C-100×10×6 C-100×10×11	Christofides69 -100×10	100	10	6 11	10000	40	200
Pe-85×7×6 Pe-85×7×11	Perl83-85×7	85	7	6 11	850	373	160
Pe-55×15×6 Pe-55×15×11	Perl83-55×15	55	15	6 11	550	240	120
Pe-12×2×6 Pe-12×2×11	Perl83-12×2	12	2	6 11	280	100	140
G-21×5×6 G-21×5×11	Gaskell67-21×5	21	5	6 11	15000	50	6000
G-22×5×6 G-22×5×11	Gaskell67-22×5	22	5	6 11	15000	50	4500
G-29×5×6 G-29×5×11	Gaskell67-29×5	32	5	6 11	15000	50	4500
G-1-32×5×6 G-1-32×5×11	Gaskell67-32×5	11	5	6 11	35000	50	8000
G-2-32×5×6 G-2-32×5×11	Gaskell67-32×5-2	32	5	6 11	35000	50	11000
G-36×5×6 G-36×5×11	Gaskell67-36×5	36	5	6 11	15000	50	250
M-134×8×6 M-134×8×11	Min92-134×8	134	8	6 11	3000	268	850
M-27×5×6 M-27×5×11	Min92-27×5	27	5	6 11	3000	272	2500

Table 2. The metadata of the second set of problem instances generated based on Archetti et al. (2013).

Instance name	Name of original instance	# of Customers $\lvert I\rvert$	# of DCs $\lvert H\rvert$	# of Markup levels $\lvert L\rvert$	DC Capacity Cap_h^{DC}	Fixed cost of establishment DCs	Vehicle Capacity Cap^v
P03-100×5×6 P03-100×5×11	P03-5	100	5	6 11	200	40	100
P03-100×10×6 P03-100×10×11	P03-10	100	10	6 11	200	40	100
P06-50×5×6 P06-50×5×11	P06-5	50	5	6 11	200	40	100
P06-50×10×6 P06-50×10×11	P06-10	50	10	6 11	200	40	100
P07-75×5×6 P07-75×5×11	P07-5	75	5	6 11	200	40	100
P07-75×10×6 P07-75×10×11	P07-10	75	10	6 11	200	40	100
P09-150×5×6 P09-150×5×11	P09-5	150	5	6 11	200	40	100
P09-150×10×6 P09-150×10×11	P09-10	150	10	6 11	200	40	100
P10-199×5×6 P10-199×5×11	P10-5	199	5	6 11	200	40	100
P10-199×10×6 P10-199×10×11	P10-10	199	10	6 11	200	40	100
P13-120×5×6 P13-120×5×11	P13-5	120	5	6 11	200	40	100
P13-120×10×6 P13-120×10×11	P13-10	120	10	6 11	200	40	100
P14-100×5×6 P14-100×5×11	P14-5	100	5	6 11	200	40	100
P14-100×10×6 P14-100×10×11	P14-10	100	10	6 11	200	40	100

4.2 Computational results for evaluating the B&P algorithm

Table 3 and Table 4 summarize the computational results for assessing the B&P algorithm on the first and second sets of problem instances used here, respectively. All instances are solved to optimality in both tables. In Table 3, each instance is solved by the algorithm under three different settings: M, MV, and BV. Letters M and B, respectively, stand for the cases that mono-directional and bounded bidirectional label-generation policies are used in the column generation phase; and letter V refers to the case that valid inequalities proposed in Section 2.4 are considered in RLMPs. In Table 4, each instance is only solved by the algorithm under two settings M and MV as the setting BV is found inefficient for our problem (see the discussion below). After solving the root node of the B&P algorithm, solving the RMP with the present columns in R^* using the CPLEX leads to a feasible solution whose objective value is a lower bound for the problem, denoted by LB_{root}. In both tables, the next columns present LB_{root}, relative improvement of the exact solution over LB_{root}, the run time required for solving the root node, the upper bound $\mathrm{UB}_{\mathrm{root}}$ provided by solving the RLMP at the root node, the total run time for solving each problem instance, the percent of the total run time consumed in column generation, and the number of nodes of the B&B tree.

Table 3 indicates that settings M and MV have the least total run times among the three algorithm settings in 85.71% and 14.29% of instances, respectively. In addition, in 53.57% of cases, setting M results to less total run times, compared to setting BV; and setting MV always outperforms setting BV. This reveals that in both datasets, using the bounded bidirectional policy is not an accelerator for label-generation (a similar observation was reported before in Archetti et al. (2013) for a VRPP). Hence, in Table 4, the results are only reported for settings M and MV for sake of shortness. In Table 4, the setting MV similarly performs better and has less total run times in 92.86% of the problem instances, rather than setting M. Hence, from Tables 3 and 4, one may conclude that the best algorithm setting is MV.

As expected, run times often increase as the number of markup levels or the number of potential DCs increase, excepting for two pairs of instances: P13-120×10×11 and P13-120×10×6, and P14-100×10×11 and P14-100×5×11. On average 97.71% of the total run time was consumed by column generation. Both Tables 3 and 4 confirm that adding the valid inequalities (20) significantly decreases the run times by reducing the number of explored nodes in 96% of instances, while it increases the time required for solving the root node in 77% of cases.

Moreover, the results approve that the lower bound provided by solving the root node is optimal in 89% of instances. The relative improvements obtained by solving the instances to optimality compared to the solutions obtained at the root nodes, are at most 16.60% and 2.84% under settings M and MV, respectively. Thus, a very promising *heuristic* for our problem can be to apply the column generation only to the root node under setting MV and to solve the resulting RMP with CPLEX, which can save 69.76% of run time on average.

Table 3. Numerical results for evaluating the proposed B&P algorithm on the first set of problem instances under different settings.

Instance	Optimal value	Algorithm setting	LB_{root}	LB_{root} error bound %	LB_{root} run time (Seconds)	UB_{root}	Total run time (Seconds)	% of CG run time	# of nodes
C-50×5×6	276.76	MV	276.76	0.00	3	276.76	3	100.00	1
		BV	276.76	0.00	29	276.76	29	100.00	1
		M	276.76	0.00	2	367.67	63	93.65	19
C-50×5×11	294.16	MV	294.16	0.00	9	294.16	9	88.89	1
		BV	294.16	0.00	1209	294.16	1209	100.00	1
		M	294.16	0.00	6	382.28	171	98.83	17
C-75×10×6	288.39	MV	287.43	0.33	7	288.99	22	90.91	3
		BV	288.39	0.00	66	288.99	215	98.60	3
		M	280.75	2.65	5	376.16	359	98.05	41
C-75×10×11	301.41	MV	301.41	0.00	17	305.26	84	96.43	3
		BV	301.41	0.00	479	305.26	3041	99.87	3
		M	292.64	2.91	13	373.18	887	99.21	39
C-100×10×6	344.00	MV	344.00	0.00	406	344.44	1066	99.25	3
		BV	344.00	0.00	4279	344.44	5596	99.84	3
		M	339.32	1.36	40667	458.22	71624	99.99	145
C-100×10×11	350.19	MV	350.19	0.00	1733	350.19	1734	99.71	1
		BV	350.19	0.00	5760	350.19	5761	99.88	1
		M	347.28	0.83	767	459.27	50075	99.99	85
Pe-85×7×6	53.40	MV	52.51	1.66	223	137.43	1921	99.17	15
		BV	52.51	1.66	1113	137.43	9344	99.90	15
		M	52.51	1.66	119	180.61	1351	99.70	17
Pe-85×7×11	68.85	MV	68.85	0.00	1949	141.84	4085	99.27	15
		BV	68.85	0.00	8176	141.84	17158	99.82	15
		M	57.42	16.60	329	188.04	2006	99.30	17
Pe-55×15×6	107.35	MV	107.35	0.00	255	132.30	672	96.88	13
		BV	107.35	0.00	4320	132.30	14868	99.76	13
		M	93.99	12.45	8	172.72	179	96.65	17
Pe-55×15×11	117.39	MV	117.39	0.00	243	142.44	1333	97.75	19
		BV	117.39	0.00	3614	142.44	15992	99.79	19
		M	109.05	7.11	23	174.57	645	97.83	23
Pe-12×2×6	71.08	MV	70.86	0.31	1	84.30	1	100.00	5
		BV	71.08	0.00	7	84.30	10	90.00	5
		M	70.86	0.31	1	87.29	1	100.00	5
Pe-12×2×11	96.66	MV	96.66	0.00	1	98.67	3	33.33	7
		BV	96.66	0.00	28	98.67	41	100.00	7
		M	96.66	0.00	1	99.16	2	50.00	7
G-21×5×6	17859.00	MV	17859.00	0.00	3	17859.00	3	66.67	1
		BV	17859.00	0.00	70	17859.00	70	100.00	1
		M	17859.00	0.00	2	17932.20	8	75.00	7
G-21×5×11	18391.90	MV	18391.90	0.00	4	18391.90	4	100.00	1
		BV	18391.90	0.00	105	18391.90	105	100.00	1
		M	18391.90	0.00	5	18463.40	23	91.30	11
G-22×5×6	8927.72	MV	8927.72	0.00	8	8927.72	8	87.50	1
		BV	8927.72	0.00	242	8927.72	242	100.00	1
		M	8927.72	0.00	3	9068.46	21	95.24	9
G-22×5×11	9097.83	MV	9097.83	0.00	13	9097.83	13	92.31	1
		BV	9097.83	0.00	4470	9097.83	4470	100.00	1
		M	9097.83	0.00	19	9238.24	123	98.37	9
G-29×5×6	11264.90	MV	11165.00	0.89	250	11325.80	246	98.37	3
		BV	11165.00	0.89	1075	11325.80	1008	99.70	3
		M	11112.70	1.35	111	11416.20	2372	99.66	23
G-29×5×11	11324.40	MV	11315.60	0.08	781	11370.20	4791	99.71	7

Instance	Optimal value	Algorithm setting	LB_{root}	LB_{root} error bound %	LB_{root} run time (Seconds)	UB_{root}	Total run time (Seconds)	% of CG run time	# of nodes
		BV	11315.60	0.08	3357	11370.20	19809	99.98	7
		M	11271.71	0.47	1161	11460.52	18634	90.25	25
G-1-32×5×6	23670.60	MV	23670.60	0.00	31	23670.60	31	100.00	1
		BV	23670.60	0.00	668	23670.60	668	100.00	15
		M	23670.60	0.00	31	23775.40	299	99.67	11
G-1-32×5×11	23994.10	MV	23994.10	0.00	138	24003.40	724	99.72	5
		BV	23994.10	0.00	401	24003.40	2084	99.95	5
		M	23994.10	0.00	138	24149.40	18301	99.96	15
G-2-32×5×6	29843.30	MV	29843.30	0.00	239	29917.50	810	99.75	3
		BV	29843.30	0.00	601	29917.50	2349	99.91	3
		M	29843.30	0.00	881	30005.50	2739	99.85	9
G-2-32×5×11	30543.70	MV	30543.70	0.00	4841	30543.70	4843	99.92	1
		BV	30543.70	0.00	19490	30543.70	19491	99.98	1
		M	30543.73	0.00	4157	30631.00	Out of memory	-	-
G-36×5×6	300.67	MV	300.67	0.00	3	300.67	3	66.67	1
		BV	300.67	0.00	47	300.67	47	100.00	1
		M	295.20	1.82	2	446.14	61	96.72	47
G-36×5×11	319.92	MV	319.92	0.00	7	322.62	129	98.45	17
		BV	319.92	0.00	65	322.62	875	99.31	17
		M	315.62	1.34	5	450.98	223	95.52	49
M-134×8×6	72.77	MV	72.77	0.00	27	72.77	27	92.59	1
		BV	72.77	0.00	93	72.77	93	97.85	1
		M	72.77	0.00	31	644.49	919	99.67	15
M-134×8×11	164.44	MV	164.44	0.00	81	164.44	81	97.53	1
		BV	164.44	0.00	282	164.44	282	99.29	1
		M	164.44	0.00	102	704.13	3079	99.84	15
M-27×5×6	2927.16	MV	2927.16	0.00	1	2927.16	1	100.00	1
		BV	2927.16	0.00	3	2927.16	3	100.00	1
		M	2927.16	0.00	1	3210.10	6	83.33	5
M-27×5×11	3543.58	MV	3543.58	0.00	1	3543.58	1	100.00	1
		BV	3543.58	0.00	4	3543.58	4	100.00	1
		M	3543.58	0.00	1	3802.43	9	100.00	5

Table 4. Numerical results for evaluating the proposed B&P algorithm on the second set of problem instances under different settings.

Instance	Optimal value	Algorithm setting	LB_{root}	LB_{root} error bound %	LB_{root} run time (Seconds)	UB_{root}	Total run time (Seconds)	% of CG run time	# of nodes
P03-100×5×6	68.75	MV	68.75	0.00	48	72.99	369	99.10	7
		M	68.75	0.00	45	84.55	871	99.85	11
P03-100×5×11	77.02	MV	77.02	0.00	313	77.87	1432	99.91	7
		M	77.02	0.00	153	88.35	2361	99.92	9
P03-100×10×6	91.85	MV	91.85	0.00	84	94.23	731	99.09	7
		M	91.85	0.00	125	110.93	4049	99.85	21
P03-100×10×11	101.74	MV	101.74	0.00	391	102.30	2597	99.64	5
		M	99.12	2.57	395	116.13	8028	99.94	15
P06-50×5×6	71.22	MV	71.22	0.00	17	73.92	54	97.53	5
		M	68.54	3.77	17	80.39	226	99.71	11

Instance	Optimal value	Algorithm setting	LB_{root}	LB_{root} error bound %	LB_{root} run time (Seconds)	UB_{root}	Total run time (Seconds)	% of CG run time	# of nodes
P06-50×5×11	83.48	MV	83.48	0.00	67	84.04	142	99.53	3
		M	81.75	2.06	91	90.02	953	99.86	9
P06-50×10×6	80.62	MV	80.62	0.00	29	81.03	76	99.12	3
		M	77.94	3.33	23	91.39	387	99.66	9
P06-50×10×11	89.79	MV	89.79	0.00	316	89.94	656	99.80	3
		M	89.79	0.00	163	97.05	1950	99.86	11
P07-75×5×6	70.72	MV	70.72	0.00	166	73.73	481	99.31	5
		M	65.76	7.02	53	85.08	960	99.72	15
P07-75×5×11	76.03	MV	76.03	0.00	642	78.11	1803	99.85	5
		M	75.88	0.19	207	87.78	4004	99.90	13
P07-75×10×6	97.16	MV	97.16	0.00	163	98.19	647	99.59	3
		M	89.42	7.97	71	109.97	1777	99.89	15
P07-75×10×11	105.20	MV	105.20	0.00	731	105.52	2635	99.90	5
		M	100.74	4.24	424	112.89	6283	99.94	11
P09-150×5×6	51.08	MV	51.08	0.00	125	52.99	1056	99.68	11
		M	51.08	0.00	111	64.95	1558	99.87	9
P09-150×5×11	57.92	MV	57.92	0.00	523	58.91	1625	99.84	5
		M	57.92	0.00	577	68.10	2757	99.95	5
P09-150×10×6	97.01	MV	94.26	2.84	454	99.70	4511	99.85	9
		M	96.67	0.35	546	115.58	9735	99.93	17
P09-150×10×11	108.79	MV	108.79	0.00	5819	109.20	12289	99.85	5
		M	108.32	0.43	2012	121.10	27612	99.98	13
P10-199×5×6	106.10	MV	106.10	0.00	2059	106.12	3028	99.91	3
		M	104.57	1.44	723	119.11	8849	99.94	11
P10-199×5×11	109.60	MV	109.60	0.00	2890	109.97	15014	99.03	5
		M	106.92	2.44	2505	120.45	26334	99.95	11
P10-199×10×6	115.65	MV	115.65	0.00	1775	115.79	10153	99.94	5
		M	115.65	0.00	1033	124.22	20452	99.97	17
P10-199×10×11	118.67	MV	118.67	0.00	12629	119.21	36853	99.89	5
		M	118.67	0.00	3673	128.58	114766	99.98	21
P13-120×5×6	35.11	MV	35.11	0.00	807	35.80	2083	99.84	7
		M	35.11	0.00	644	54.46	2359	99.83	9
P13-120×5×11	41.90	MV	41.90	0.00	3608	41.96	7449	99.93	5
		M	41.80	0.24	2693	55.16	11191	99.95	7
P13-120×10×6	35.11	MV	35.11	0.00	1271	36.04	12545	99.86	11
		M	34.04	3.05	496	64.64	10839	99.92	23
P13-120×10×11	43.89	MV	43.89	0.00	6289	43.89	6289	99.89	1
		M	43.89	0.00	2013	66.28	19568	99.96	9
P14-100×5×6	69.25	MV	69.25	0.00	85	71.04	569	99.65	7
		M	67.48	2.56	97	73.00	971	99.86	7
P14-100×5×11	75.45	MV	75.45	0.00	405	75.96	2219	99.91	7
		M	75.40	0.07	492	76.71	3059	99.89	7
P14-100×10×6	100.38	MV	100.38	0.00	150	101.04	918	99.13	7
		M	99.60	0.77	57	101.95	761	99.82	7
P14-100×10×11	104.83	MV	104.83	0.00	210	105.65	2207	99.73	5
		M	104.83	0.00	240	106.37	2038	99.93	5

4.3 Evaluating polynomial-size model

This subsection compares the B&P algorithm with an alternative method that uses CPLEX to directly solve the polynomial-size model (1)-(13); the latter is here called *the CPLEX method*. For the first set of instances, Table 5 summarizes the numerical results required to compare the B&P algorithm under setting MV and CPLEX method. The columns of this table report the run times, the upper bounds provided by optimally solving the continuous relaxation of the polynomial-size and set-packing formulations, the value of the best feasible solution found, the error bound, and the number of nodes explored. The run-time limit for the CPLEX method is set to 10 hours.

Table 5. Numerical results of solving the polynomial-size model by CPLEX on the first set of problem instances.

Instance	Run time (Seconds)		Upper bound		Best solutionۥs objective value		Error bound %		# of nodes	
	CPLEX method	B&P algorithm	CPLEX method	B&P algorithm	CPLEX method	B&P algorithm	CPLEX method	B&P algorithm	CPLEX method	B&P algorithm
C-50×5×6	32171	3	1878.08	276.76	273.25	276.76	506.34	0.00	42077	1
C-50×5×11	25492	9	1889.61	294.16	289.11	294.16	481.05	0.00	29661	1
C-75×10×6	>36000	22	3585.37	288.99	117.46	288.39	>1000	0.00	748	3
C-75×10×11	>36000	84	3607.29	305.26	129.16	301.41	>1000	0.00	1134	3
C-100×10×6	>36000	1066	4429.21	344.44	172.47	344.00	>1000	0.00	900	3
C-100×10×11	>36000	1734	4444.68	350.19	160.27	350.19	>1000	0.00	649	1
Pe-85×7×6	17237	1921	1314.27	137.44	0.00	53.40	NA	0.00	7705	15
Pe-85×7×11	22129	4085	1347.65	141.84	0.00	68.85	NA	0.00	7639	15
Pe-55×15×6	>36000	672	1248.72	132.30	0.00	107.35	NA	0.00	6250	13
Pe-55×15×11	>36000	1333	1248.72	142.44	0.00	117.39	NA	0.00	5340	19
Pe-12×2×6	17108	1	182.10	84.30	71.08	71.08	0	0.00	2694865	5
Pe-12×2×11	2766.	3	186.16	98.67	96.66	96.66	0	0.00	262497	7
G-21×5×6	12730	3	56577.20	17859.00	17775.06	17859.00	205.95	0.00	252361	1
G-21×5×11	>36000	4	56850.00	18391.90	18290.59	18391.90	202.20	0.00	772750	1
G-22×5×6	11625	8	41905.49	8927.72	8667.44	8927.72	137.31	0.00	237007	1
G-22×5×11	10488	13	42051.33	9097.83	9097.83	9097.83	135.85	0.00	227398	1
G-29×5×6	>36000	246	47167.21	11325.80	10572.10	11264.90	233.16	0.00	96932	3
G-29×5×11	16881	4791	47351.89	11370.20	10861.99	11324.40	238.27	0.00	114681	7
G-1-32×5×6	26710	31	104743.71	23670.60	21291.95	23670.60	329.65	0.00	106410	1
G-1-32×5×11	>36000	724	104781.00	24003.40	22481.34	23994.10	331.82	0.00	76915	5
G-2-32×5×6	>36000	810	107879.00	29917.50	28086.30	29843.30	267.78	0.00	182313	3
G-2-32×5×11	>36000	4843	108271.00	30543.70	28284.30	30543.70	267.92	0.00	183489	1
G-36×5×6	18323	3	2411.49	300.67	260.20	300.67	658.42	0.00	52797	1
G-36×5×11	>36000	129	2411.49	322.62	304.04	319.92	564.79	0.00	35132	17
M-134×8×6	>36000	27	12875.67	72.77	0.00	72.77	NA	0.00	21486	1
M-134×8×11	>36000	81	12938.50	164.44	0.00	164.44	NA	0.00	12933	1
M-27×5×6	22888	1	24423.73	2927.16	2927.16	2927.16	477.32	0.00	189118	1
M-27×5×11	11215	1	24537.94	3543.58	3543.58	3543.58	473.39	0.00	132592	1

As one can realize from Table 5, only two instances Pe-12×2×11 and Pe-12×2×6 can be solved exactly by the CPLEX method. For three other instances G-22×5×11, M-27×5×11, and M-27×5×6, this method is capable to find the exact solutions, but it cannot approve their optimality as the reported error bounds for these instances are large numbers 135.83, 473.39 and 477.32, respectively. Excluding the two instances Pe-12×2×11 and Pe-12×2×6, for the half of the instances, CPLEX reported ۥout of memoryپstatus, before reaching 10-hour run-time limit. The average time consumed by the CPLEX method is 26848.86 seconds (within which only two instances can be solved optimally), while it is 808.86 seconds for the B&P algorithm that can solve all instances exactly. Moreover, the average relative improvement in the upper bound obtained by the set-packing formulation is 81.60%, which clearly explains why the number of B&B nodes drastically decreases for the B&P algorithm.

4.4 Advantage of differential pricing

To evaluate the effect of price differentiation, for each customer one fixed markup level is considered, whose corresponding markup percentage is set to the average of the markup percentages, i.e., 0.15. Figure 3 displays the *relative improvements* (RIs) obtained by using the proposed PM-LRP, which integrates price, location, and routing decisions; over the case where the markup percentage is fixed to 0.15 before optimizing routing and location decisions for in the first set of our test problems with $|L| = 6$. The average RI is 43.03%, which evidently shows why one should incorporate pricing decisions into determining location decisions when the demands are price sensitive.

It worth noting that if the number of the markup level is set to 1, the problem actually becomes a special PM-LRP with demands that are not price sensitive. In this case, the average run time of solving the instances is 7 seconds, which is considerably smaller than 808.86 seconds, the average run time required for solving the instances with six markup levels. This discloses the complexity degree of the PM-LRP with price-sensitive demands compared to the non-price-sensitive case.

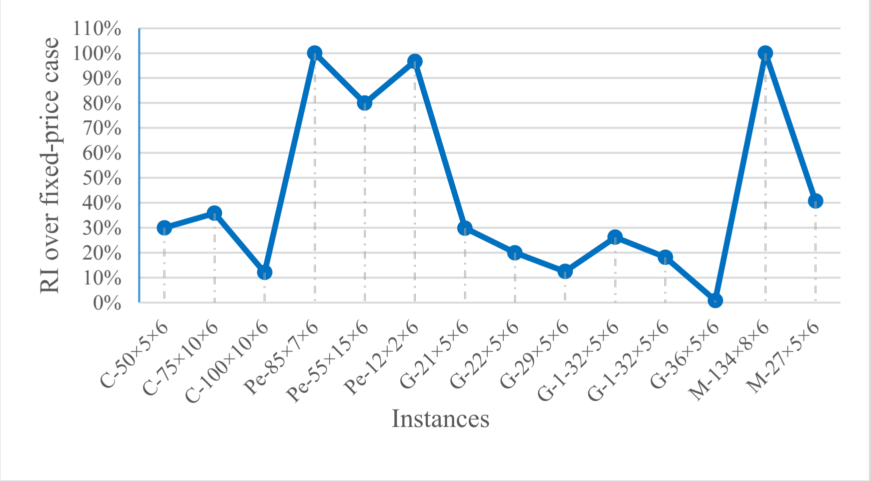

Figure 3. Relative improvement of the proposed PM-LRP with differential pricing over the case with fixed prices.

4.5. Advantage of integrating routing

In the location-routing literature, it is traditional to compare applying the *integrated* model developed for an LRP to the *sequential* approach where the location of facilities, together with other related long-term decisions, is decided first, and the routing aspects are determined next (Salhi & Rand, 1989; Ahmadi-Javid

& Azad, 2010). To apply the sequential approach to our problem; the location, allocation, and pricing decisions are determined first, and the routing of vehicles is decided next. Hence, in the first phase, the locations of DCs, the number of vehicles assigned to each open DC, the set of customers allocated to each open DC (on the simple basis of the distance from DCs), prices offered to the allocated customers are determined so as to maximize the total profit of serving customers. In the second phase, a VRPP is solved for each open DC and its assigned customers, whose demands and profits are known and fixed based on the pricing decisions determined in the first phase.

Figure 4 plots the *relative improvements* (RIs) obtained by using our integrated approach that integrates location, pricing, and routing over the sequential approach described above. The average RI is 51.23%. In an extreme case, the output of using the sequential approach is that establishing any DC is not profitable for two instances Pe-85×7×11 and M-134×8×6 whose RIs are 100%. This analysis clearly shows why one should incorporate routing decisions into determining location decisions when the customer demands are price sensitive.

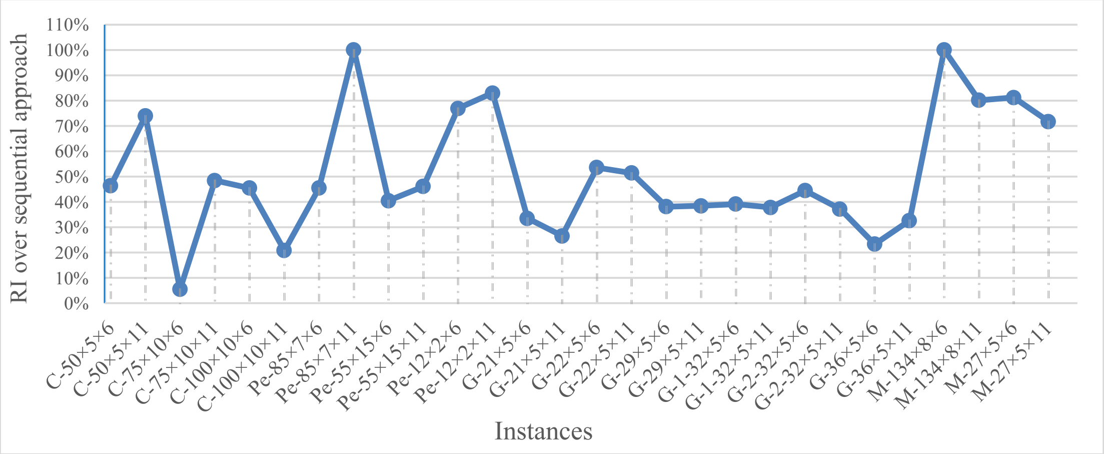

Figure 4. Comparison of the integration approach and sequential approaches to solve the proposed PM-LRP.

5. Concluding remarks

This paper proposes an integrated profit-maximization location-routing problem with price-sensitive demands that optimizes location, routing and pricing decisions simultaneously. The problem determines the location of facilities, the allocation of vehicles and customers to the facilities, the delivered pricing, and the routing decisions, in order to maximize the total profit of serving the customers.

The problem is modeled as a mixed-integer linear program. A branch-and-price algorithm is used as the solution method after reformulating the model to a set-packing master problem and a series of elementary shortest path subproblems, by employing the Dantzig-Wolfe decomposition. A label-setting dynamic programming algorithm is used to solve subproblems.

Computational results are discussed on instances based on Barretoبs data set (2003) for LRPs and the instances of Christofides et al. (1979) for VRPs, but with some modifications to suit our specific problem. The numerical study indicates a significant increase in the total profit collected by applying the proposed integration approach, compared to the traditional approach, which considers a fixed price for the product, and consequently, fixed demands for the customers, and then solves the associated LRP. All instances are solved both in the existence and absence of the valid inequalities. Results report great improvement of the total CPU time by considering the inequalities. Another numerical study is done, to compare the branch-and-price algorithm with the polynomial-size model. Outcomes confirm the excellence of the branch-and-price algorithm for solving the instances. To compare the proposed model to one where routing information is not made use of in the locational decision, another model known as the sequential approach is solved, which proves the importance of incorporating routing decisions into determining location decisions when demands are price sensitive.

Many suggestions for future research can be offered inspired by different types of LRPs. It is sometimes claimed that location-routing has a temporal inconsistency, as location is a long-term investment, while routes can change frequently, especially when the demands vary over time. A way to counter this criticism is considering LRPs with multiple planning periods, also known as dynamic LRPs. Hence, an important future study is to investigate dynamic PM-LRPs. Another way to enhance our model is to incorporate the uncertainty of the customer demands using methods such as robust optimization or stochastic programming. Extending PM-LRPs that consider other relevant decisions such as inventory decisions remains another open research area. This research could also be continued by considering other routing settings, such as considering routing with split delivery.

References

Ahmadi-Javid, A., & Azad, N. (2010). Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E: Logistics and Transportation Review,* 46, 582ت597.

Ahmadi-Javid, A., & Ghandali, R. (2014). An efficient optimization procedure for designing a capacitated distribution network with price-sensitive demand. *Optimization and Engineering*, 15, 801-817.

Ahmadi-Javid, A., & Hoseinpour, P. (2015a). A location-inventory-pricing model in a supply chain distribution network with price-sensitive demands and inventory-capacity constraints. *Transportation Research Part E: Logistics and Transportation Review*, 82, 238-255.

Ahmadi-Javid, A., & Hoseinpour, P. (2015b). Incorporating location, inventory and price decisions into a supply chain distribution network design problem. *Computers & Operations Research*, 56, 110ت119.

Ahmadi-Javid, A., & Seddighi, A.H. (2013). A location-routing problem with disruption risk. *Transportation Research Part E: Logistics and Transportation Review,* 53, 63ت82.

Ahuja, R.K., Magnanti, T.L., & Orlin, J.B. (1993). Network Flows: Theory, Algorithms, and Applications, *Prentice-Hall*.

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

Anderson, S.P., de Palma, A. & Thisse, J.F. (1989). Spatial price policies reconsidered. *Journal of Industrial Economics*, 38, 1-18.

Appleget, J.A., & Wood, R.K. (2000). Explicit-constraint branching for solving mixed-integer programs. In: M. Laguna and J.L. Gonz´alez-Velarde, editors, *Computing Tools for Modeling, Optimization and Simulation,* Kluwer Academic Publishers, Boston, 243-261.

Archetti, C., Bertazzi, L., Laganط, D., & Vocaturo, F. (2017). The Undirected Capacitated General Routing Problem with Profits. *European Journal of Operational Research*, 257(3), 822-833.

Archetti, C., Bianchessi, N., & Speranza, M.G. (2013). Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, 161(4), 547-557.

Archetti, C., Feillet, D., Hertz, A., & Speranza, M.G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60, 831ك842.

Archetti C., Speranza, M.G. & Vigo, D. (2014). Vehicle routing problems with profits. In: Toth, P., & Vigo, D. (Eds.) *Vehicle Routing: Problems, Methods, and Applications* (Vol. 18). SIAM, Philadelphia.

Averbakh, I., & Berman, O. (1994). Routing and location-routing p delivery men problems on a path. *Transportation Science*, 28, 184-197.

Averbakh, I., & Berman, O. (1995). Probabilistic sales-delivery man and sales-delivery facility location problems on a tree. *Transportation Science*, 29, 184-197.

Averbakh, I., & Berman, O. (2002). Minmax p-traveling salesmen location problems on a tree. *Annals of Operations Research*, 110, 55-62.

Averbakh, I., Berman, O., & Simchi-Levi, D. (1994). Probabilistic a priori routing-location problems. *Naval Research Logistics*, 41, 973-989.

Barreto, S. (2003). LRP instances: http://sweet.ua.pt/sbarreto/.

Basu, A., Ingene, C.A., & Mazumdar, T. (2004). The pricing of delivery services. *Journal of Regional Science*, 44(4), 743-772.

Berger, A., Grigoriev, A., Panin, A., & Winokurow, A. (2017). Location, pricing and the problem of Apollonius. *Optimization Letters*, to appear.

Bertsimas, D., & Tsitsiklis, J. N. (1997). *Introduction to linear optimization*. Belmont, MA: Athena Scientific.

Carlton, D.W. (1983). A reexamination of delivered pricing systems. *The Journal of Law and Economics*, 26(1), 51-70.

Chen, X., & Simchi-Levi, D. (2012). Pricing and inventory management. *The Oxford handbook of pricing management.* Oxford University Press.

Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem, in N. Christofides et al. (Eds.), *Combinatorial Optimization*, Wiley, Chichester, 315-338.

Dantzig, G. & Wolfe, P. (1960). Decomposition principle for linear programs, *Operations Research,* 8, 101ت111.

Desrochers, M. (1988). An algorithm for the shortest path problem with resource constraints, Technical report G-88-27, GERAD, Montreal.

Desrochers, M. & Soumis, F. (1989). A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, 23(1), 1ت13.

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

El-Hajj, R., Dang, D.C., & Moukrim, A. (2016). Solving the team orienteering problem with cutting planes. *Computers & Operations Research*, 74, 21-30.

Espinosa, M.P. (1992). Delivered pricing, FOB pricing, and collusion in spatial markets. *The Rand Journal of Economics*, 23(1), 64-85.

Fackler, P.L., & Goodwin, B.K. (2001). Spatial price analysis. *Handbook of Agricultural Economics*, Volume 1, Part B, 971-1024.

Feillet, D., Dejax, P., & Gendreau, M. (2004). An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems, *Networks,* 44(3), 216ت229.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Travelling salesman problem with profits, *Transportation Science*, 39, 188ت205.

Fernعndez, P., Pelegrك, B., Pفez, M.D.G., & Peeters, P.H. (2007). A discrete long-term locationتprice problem under the assumption of discriminatory pricing: Formulations and parametric analysis. *European Journal of Operational Research*, 179(3), 1050-1062.

Gansterer, M., Kغىktepe, M., & Hartl, R.F. (2017). The multi-vehicle profitable pickup and delivery problem. *OR Spectrum*, 39(1), 303-319.

Greenhut, M.L., Hwang, M., & Ohta, H. (1975). Observations on the shape and relevance of the spatial demand function. *Econometrica,* 43, 669 ت682.

Gunawan, A., Lau, H.C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2), 315-332.

Hanjoul, P., Hansen, P., & Peeters, D. (1990). Uncapacitated plant location under alternative spatial price policies. *Management Science*, 36, 41ت57.

Hansen, P., Peeters, D., & Thisse, J.F. (1981). Simple plant location under uniform delivered pricing. *European Journal of Operational Research*, 6, 94ت103.

Hansen, P., Thisse, J.F., & Hanjoul, P. (1997). Facility location under zone pricing. *Journal of Regional Science*, 37, 1ت22.

He, Z., Cheng, T.C.E., Dong, J., & Wang, S. (2016). Evolutionary location and pricing strategies for service merchants in competitive O2O markets. *European Journal of Operational Research*, 254(2), 595-609.

Hoover, E. M. (1937). Spatial price discrimination. *The Review of Economic Studies*, 4(3), 182-191.

Hotelling, H. (1929). Stability in competition. *Economic Journal*, 39, 41-57.

Houck Jr, D.J., Picard, J.-C., Queyranne, M. & Vemuganti, R.R. (1980). The Travelling Salesman Problem as a Constrained Shortest Path Problem: *Theory and Computational Experience.* Opsearch 17, 93ت109.

Irnich, S., & Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers,J. Desrosiers, and M. M. Solomon, editors, Column Generation, Springer US, pages 33ت65.

Jamil, M., Batta, R., & Malon, D.M. (1994). The traveling repairperson home base location problem. *Transportation Science*, 28, 150-161.

Lambrecht, A., Seim, K., Vilcassim, N., Cheema, A., Chen, Y., Crawford, G.S., Hosanagar, K., Iyengar, R., Koenigsberg, O., Lee, R., & Miravete, E.J. (2012). Price discrimination in service industries. *Marketing Letters*, 23(2), 423-438.

Laporte, G. (1988). Location-routing problems. In: B. L. Golden & A. A. Assad, (Eds.), *Vehicle routing: Methods and studies*. North-Holland, Amsterdam. 293ت318.

Laporte, G., & Dejax, P.J. (1989). Dynamic location-routing problems. *Journal of the Operational Research Society*, 40, 471-482.

Lederer, P.J., & Thisse, J.F. (1990). Competitive location on networks under delivered pricing. *Operations Research Letters*, 9(3), 147-153.

Lopez, R.B., Ferreira, C., Santos, B.S., & Barretto, S. (2013). A taxonomical analysis, current methods and objectives on location-routing problems. *International Transactions in Operational Research* 20, 795-822.

Miller, C.E., Tucker, A.W., & Zemlin, R.A. (1960). Integer programming formulations and traveling salesman problems. *Journal of Association for Computing Machinery,* 7, 326ت329.

Min H., Jayaraman V., & Srivastava R. (1998). Combined location-routing problems: a synthesis and future research directions. *European Journal of Operational Research,* 108, 1ت15.

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: A review. *European Journal of Operational Research*, 117, 649ت672.

Pelegrک, B., Fernعndez, P., Pعفez, M.D.G., & Hernعndez, S.C. (2012). On the location of new facilities for chain expansion under delivered pricing. *Omega*, 40(2), 149-158.

Pعفez, M.D.G., Hernعndez, P.F., & Pelegrک, B.P. (2004). On price competition in location-price models with spatially separated markets. *TOP*, 12(2), 351-374.

Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research,* 238, 1ت17.

Righini, G., & Salani, M. (2006). Symmetry Helps: Bounded Bi-directional Dynamic Programming for the Elementary Shortest Path Problem with Resource Constraints. *Discrete Optimization,* 3(3), 255ت273.

Righini, G., & Salani, M. (2008). New Dynamic Programming Algorithms for the Resource Constrained Elementary Shortest Path Problem. *Networks,* 51(3), 155ت170.

Ryan, D.M., & Foster. B.A. (1981). An integer programming approach to scheduling. A. Wren, ed. *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*. North Holland, Amsterdam, The Netherlands, 269ت280.

Salhi, S., & Rand, G.K. (1989). The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39, 150-156.

Vansteenwegen, P., Souffriau, W., & Oudheusden, D.V. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209, 1, 1ت10.

Varian, H. R. (1989). Price discrimination. *Handbook of Industrial Organization*, 1, 597-654.

Vidal, T., Maculan, N., Ochi, L. S., & Vaz Penna, P.H. (2015). Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science*, 50(2), 720-734.

Vogel, J. (2011). Spatial price discrimination with heterogeneous firms. *The Journal of Industrial Economics*, 59(4), 661-676.

Wagner, J.L., & Falkson, L.M. (1975). Optimal nodal location of public facilities with price-sensitive demand. *Geographical Analysis*, 7, 69ت83.