Iranian Journal of Operations Research
Vol. 12, No. 1, 2021, pp. 173-183

A branch and cut algorithm for the Undirected Profitable Location Rural Postman Problem

M. Khorramzadeh[1,*], R. Javvi[2]

This paper is concerned with presenting an exact algorithm for the Undirected Profitable Location Rural Postman Problem. This problem combines the profitable rural postman and facility location problems and also has some interesting real-life applications. Fixed costs are associated with end points of each profitable edge and the objective is to choose a subset of profitable edges such that the difference between the profit collected and the cost of opening facilities and traveling cost is maximized. A dominance relation is used to present an integer programming formulation for the problem and a branch and cut algorithm is developed for solving the problem and extensive numerical results on real-world benchmark instances are given to evaluate the quality of presented algorithms.

***Keywords**: Rural Postman Problem, Branch and cut; Location Problem; Arc Routing Problem with Profits; Undirected Graph.*

Manuscript was received on 01/31/2021, revised on 12/07/2021 and accepted for publication on 03/11/2022.

1. Introduction

A typical routing problem is concerned with finding the best route with some properties, satisfying demands of a set of customers. If customers are associated to nodes (arcs), then the corresponding problem is called node (arc) routing problem. It can be shown that an arc routing problem can be converted into node routing problem, replacing each arc with two or three vertices [18, 4,12]. However, the graph of the resulting node routing problem is complete, has larger size and the number of arcs change from linear to quadratic [20]. Moreover, the resulting node routing problem, requires either fixing of variables or the use of edges with infinite cost. This motivates the study of arc routing problems.

In a typical profitable rural postman problem, a profit is assigned to each profitable customer (arc) and a decision has to be taken to determine whether it will be serviced or not. In a facility location problem (FLP) a set of potential locations for the facilities and a set of customers are given. Facilities offer service to customers. The objective is to determine where to locate the facilities and how to satisfy the demand of customers from located facilities [15]. In [3], a new class of problem was introduced, which combines the profitable rural postman problem and facility location problem on a directed graph. The resulting problem is called directed profitable location rural postman problem (DPLRPP) and has many real-world applications [3]. In DPLRPP, fixed costs are associated with end points of each profitable arc and the profit is collected only if both facilities are established on the end vertices of the corresponding profitable arc. The objective is to choose a subset of profitable arcs such that the difference between the profit collected and the cost of opening facilities and traveling cost is maximized. They presented an integer linear programming

* Corresponding Author.
[1] Department of Shiraz, University of Technology, Shiraz, Iran, m.khorrami@sutech.ac.ir.
[2] Department of Shiraz, University of Technology, Shiraz, Iran , r.javvi@sutech.ac.ir.

formulation DPLRPP and solved by using an efficient branch and cut algorithm. They also presented extensive numerical results to examine the efficiency of their presented algorithm.

This paper is concerned with the PLRPP on undirected graphs (UPLRPP).A dominance relation is used to present an integer programming formulation for the problem. The presented model is solved by using an efficient exact algorithm that handles the subtour elimination constraints and parity constraints in a branch and cut framework. The presented algorithm is applied on benchmark instances taken from real-world applications and the numerical results are analyzed.

As far as we know, the UPLRPP has never been studied in the literature so far. However, related problems have been considered. In 1989, Levy and Budin [13], described arc location routing problem (ALRP), where the allocation of customers to depots is an arc oriented problem as well as the resulting routing problem. They also presented an algorithm for solving the ALRP and computational results were given regarding parameter settings and performance of the algorithm. Ghiani and Laporte [9] considered the problem of locating a set of depots in an arc routing context with no side constraints. They presented a branch and cut algorithm for solving the resulting problem. In 2000, Ghiani and Laporte [10] presented a binary linear programming formulation for the undirected rural postman problem and investigated some related polyhedral theory. Moreover, they presented a branch and cut algorithm for the problem and discussed extensive computational results to evaluate the efficiency of their presented algorithm.

In 2001 Ghiani and Laporte [10] examined the class of location arc routing problems where a set of required edges have to be served by vehicles starting and ending their tours at some depots which have to be located incurring in a fixed cost. In 2003 Muyldermans [17], presents a variant of the location arc routing problem (LARP) called the p deadmileage problem. In this problem, unlike the previously addressed splitting of the demand is allowed. The objective was to minimize deadmileage (deadheading) and the problem was solved exactly. In 2006 Pia and Filippi [19] probed variants of the capacitated arc routing problem with a structure similar to the LARP. In 2007 Amaya et al. [2] the capacitated arc routing problem with mobile depots and the capacitated arc routing problem with re_ll points. In 2008 Liu et al. [14] surveyed the LARP and presented some future research trends such as variants of the LARP and better algorithms. In 2013, Doulabi and Sei_ [6] studied multi-depot location arc routing problems with vehicle capacity constraints and presented

Two mixed integer programming models for single and multi-depot problems. In 2014 Lopez et al. [16] presented new constructive and improvement methods for LARP and used them within different metaheuristic frameworks. In 2014 Arbib et al. [3], introduced PLRPP on directed graphs. The integer programming formulation presented in [3] is quite different from our presented formulation. Moreover, in [3], the subtour branching strategy is not discussed. In 2019, Fernandez et al. [8], modeled and solved several families of location arc routing problems on an undirected graph that extend the multi-depot rural postman problem to the case where the depots are not fixed. In these problems, the profit is not considered and the aim is to select the facility locations and to construct a set of routes traversing each required edge of the graph, where each route starts and ends at the same facility. They presented a polyhedral study for some of the formulations and solved by branch and cut method.

In section 2, the integer programming formulation of UPLRPP is presented. Section 3 is devoted to the description of presented algorithm. In section 4, numerical results are given to evaluate the efficiency of presented algorithms.

2. Problem description

UPLRPP can be considered as a combination of the facility location and arc routing with profits. UPLRPP can be defined on an undirected graph *G = (V;E)*, where $v = \{0, \dots, n\}$ is the set of vertices, vertex 0 denotes the depot and E is the set of edges. The cost of establishing a facility in vertex $v \in V$ is denoted by f_v. The non-negative cost of traversing arc $e = (u, v) \in E$ is denoted by$c_{e=}c_{(u,v)}$. Let $E_P \subset E$ be the set of profitable edges. A nonnegative profit p_e is associated with each profitable edge $e \in E_P$ which is collected only once if e is traversed and facilities are located at both endpoints of e. The objective is to find the tour that starts from the depot and ends at the depot and maximizes the difference between the total collected profits and the traveling and fixed costs.

In the following, let $\delta(S) = \{\{u, v\} | \{u, v\} \in E, u \in S, v \in V/S\}$ and $\delta(\{v\})$ is denoted by $\delta(v)$. Moreover, let $\gamma(S) = \{\{u, v\} | \{u, v\} \in E, u, v \in S\}$. In order to introduce the integer programming formulation for UPLRPP, for each $e \in E$, let x_e denotes the number of times that e is traversed in the tour and let z_e be a binary variable that takes the value one if the profitable edge e is served. Moreover, for each vertex $\in V$, let w_v be a binary variable that is one if the facility is established at the vertex v.

The integer programming formulation of UPLRPP is as follows.

$$Maximize \sum_{e \epsilon E_P} p_e z_e - \sum_{e \epsilon E} c_e x_e - \sum_{e \epsilon V} f_v w_v \qquad (1)$$

$$s.t. \quad w_0 = 1. \qquad (2)$$

$$x(\delta(0)) \geq 2x_{\acute{e}}. \qquad \acute{e} \epsilon E \setminus \delta(0). \qquad (3)$$

$$x(\delta(0)) \equiv 0 \qquad (mod 2). \qquad \forall v \epsilon V. \qquad (4)$$

$$x(\delta(0)) \geq 2(w_u + w_v - 1). \qquad S \subseteq V, u \in S. v \notin S. \qquad (5)$$

$$z_e \leq x_e. \qquad \forall e \in E_p. \qquad (6)$$

$$z_{uv} \leq w_u. \qquad \{u. v\} \in E_p. \qquad (7)$$

$$z_{uv} \leq w_v. \qquad \{u. v\} \in E_p. \qquad (8)$$

$$z_e \in \{0.1\} \qquad \forall e \in E_p. \qquad (9)$$

$$w_u \in \{0.1\} \qquad \forall e \in V. \qquad (10)$$

$$x_e \in \mathbb{N} \cup \{0\} \qquad \forall e \in E. \qquad (11)$$

Here (1) expresses the objective function as the maximization of the sum of collected profits minus the traveling and fixed costs. The set of inequalities (4) impose degree constraints. The set of constraints (5) guarantee the connectivity of solution. Inequalities (6) indicate that a profitable arc e must be traversed at least once to be served. Inequalities (7) and (8) imply that a profit can be collected only if a facility is located at both end points of the corresponding arc.

Each feasible solution of UPLRPP can be represented as $(\mathbf{x}, \boldsymbol{z}, \mathbf{w}) \in Z^{|E|} \times \mathbb{B}^{|E_P|} \times \mathbb{B}^{|V|}$ where $\mathbf{x} = (x_e)_{e\in E}, \mathbf{z} = (z_e)_{e\in E_P}, \mathbf{w} = (w_v)_{v\in V}$ and $\mathbb{B}$= {0,1}.

Theorem 1. There exists an optimal solution of the UPLRPP in which any edge is traversed at most twice.

Proof By contradiction, assume that in every optimal solution of the UPLRPP there exists some edge $\bar{e} \in E$ such that$x_{\bar{e}} > 2$. Let $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{w}^*)$ be the optimal solution of UPLRPP with minimum value of $\sum_{e\in E} x_e^*$ be such that $x_{\bar{e}}^* > 2$. Define $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{w}})$ as follows; $\bar{\mathbf{z}} = \mathbf{z}^*, \bar{\mathbf{w}} = \mathbf{w}^*$and $\bar{\mathbf{x}} = (\bar{x}_e)_{e\in E}$, where

$$\bar{x}_e = \begin{cases} x_e^*, & e \neq \bar{e} \\ x_e^* - 2, & e = \bar{e} \end{cases}$$

It can be easily verified P that $(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{w}})$ is also a UPLRPP solution that satisfies $\sum_{e\in E} x_e^* < \sum_{e\in E} \bar{x}_e$. This contradicts the definition of $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{w}^*)$. Therefore, there exists an optimal solution of the UPLRPP in which any edge is traversed at most twice.

Let x_e be a binary variable that is one if e is traversed and y_e be a binary variable that is one if e is traversed twice. Moreover, let z_e, $e \in E_P$ and $w_v, v \in V$, be the same as defined before. Using the Theorem 1, the integer programming formulation of UPLRPP can be presented as follows.

$$Maximize \sum_{e\in E_P} p_e z_e - \sum_{e\in E} c_e x_e - \sum_{e\in E} c_e y_e - \sum_{v\in V} f_v w_v \tag{12}$$

$$s.t. \quad w_0 = 1. \tag{13}$$

$$x\big(\delta(0)\big) \geq 2x_{\acute{e}}. \qquad \acute{e}\epsilon E \setminus \delta(0). \tag{14}$$

$$x\big(\delta(0)\big) + y(\delta(v)) \equiv 0 \qquad (mod2) . \qquad \forall v\epsilon V. \tag{15}$$

$$x\big(\delta(0)\big) + y\big(\delta(v)\big) \geq 2(w_u + w_v - 1). \qquad S \subseteq V. u \in S. v \notin S . \tag{16}$$

$$y_e \leq x_e. \qquad \forall e \in E_p . \tag{17}$$

$$z_e \leq x_e. \qquad \forall e \in E_p . \tag{18}$$

$$z_{uv} \leq w_u . \qquad \{u. v\} \in E_p. \tag{19}$$

$$z_{uv} \leq w_v . \qquad \{u. v\} \in E_p. \tag{20}$$

$$z_e \in \{0.1\} \qquad \forall e \in E_p. \tag{21}$$

$$w_u \in \{0.1\} \qquad \forall e \in V . \tag{22}$$

$$x_e, y_e \in \{0.1\} \qquad \forall e \in E . \tag{23}$$

Constraints (15) can be linearized using the co-circuit inequality [1]. This results in the following integer programming formulation for UPLRPP.

$$Maximize \sum_{e \epsilon E_P} p_e z_e - \sum_{e \epsilon E} c_e x_e - \sum_{e \epsilon E} c_e y_e - \sum_{v \in V} f_v w_v \qquad (24)$$

$$s.t. \quad w_0 = 1. \qquad (25)$$

$$x(\delta(0)) \geq 2x_{\acute{e}}. \qquad \acute{e} \in E \setminus \delta(0). \qquad (26)$$

$$X(\delta(S) \setminus F) + y(F \setminus L) \geq x(F) + y(L) - (|F| + |L|) + 1,$$
$$\forall S \subseteq V. F \subseteq \delta(S). L \subseteq F. |F| + |L| \ \ odd \qquad (27)$$

$$(\delta(S)) + y(\delta(S)) \geq 2(w_u + w_v - 1). \qquad S \subseteq V. u \in S. v \notin S. \qquad (28)$$

$$y_e \leq x_e. \qquad \forall e \in E_p. \qquad (29)$$

$$z_e \leq x_e. \qquad \forall e \in E_p. \qquad (30)$$

$$z_{uv} \leq w_u. \qquad \{u.v\} \in E_p. \qquad (31)$$

$$z_{uv} \leq w_v. \qquad \{u.v\} \in E_p. \qquad (32)$$

$$z_e \in \{0.1\} \qquad \forall e \in E_p. \qquad (33)$$

$$w_u \in \{0.1\} \qquad \forall e \in V. \qquad (34)$$

$$x_e. y_e \in \{0.1\} \qquad \forall e \in E. \qquad (35)$$

Note that every feasible solution of UPLRPP can be represented as

$$(x, y, z, w) \in \mathbb{B}^{|E|} \times \mathbb{B}^{|E|} \times \mathbb{B}^{|\mathbb{E}_P|} \times \mathbb{B}^{|V|}$$

where $\boldsymbol{x} = (x_e)_{e \in E}, \boldsymbol{y} = (y_e)_{e \in E}, \boldsymbol{z} = (z_e)_{e \in E_P}, \boldsymbol{w} = (w_v)_{v \in V}$. In the following, let UPLRPP(G) denotes the convex hull of all UPLRPP solutions. Let $P_1(G)$ be the convex hull of all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{B}^{|E|} \times \mathbb{B}^{|E|}$ that satisfies the following constraints.

$$x(\delta(0)) \geq 2x_{\acute{e}}. \qquad \acute{e} \in E \setminus \delta(0). \qquad (36)$$

$$\mathrm{X}(\delta(S) \setminus F) + y(F \setminus L) \geq x(F) + y(L) - (|F| + |L|) + 1,$$
$$\forall S \subseteq V. F \subseteq \delta(S). L \subseteq F. |F| + |L| \ \ odd \qquad (37)$$

$$(\delta(S)) + y(\delta(S)) \geq 2. \qquad S \subseteq V \setminus \{1\}. \qquad (38)$$

$$y_e \leq x_e. \qquad \forall e \in E. \qquad (39)$$

By a proof that is similar to the proof of Theorem 2 of [5], it can be proved that.

Theorem 2. $\dim(P_1(G)) = 2|E|$ = if and only if every cut set of G has at least three edges.

Let $RPP(G)$ denotes the rural postman problem on G, in which the set of required edges is E_P and the vertex 0 is the depot. In what follows, we assume that $RPP(G)$ has at least one feasible solution. In other words, we assume that there exists a solution $(\mathrm{x}^{\mathrm{RPP}}, \mathrm{y}^{\mathrm{RPP}}) \in \mathbb{B}^{|E|} \times \mathbb{B}^{|E|}$ that starts from the depot, visits all edges of E_P and returns to the depot.

Now, let $\bar{P}(G)$ denotes the convex hull of all $(\mathrm{x},\mathrm{y},z,\mathrm{w})$ that satisfies all constraints of UPLRPP except constraint (25) i.e. the convex hull of all $(\mathrm{x},\mathrm{y},z,\mathrm{w})$satisfying the set of constraints (26), (27), (28), (29) (30), (31), (32), (33), (34) and (35). Clearly $\dim(\mathrm{UPLRPP}) = \bar{P}(G)$-1 . The next theorem is concerned with the dimension of $\dim(\bar{P}(G))$.

Theorem 3. If every cut set of G has at least three edges and $RPP(G)$ has at least one feasible solution, then $\dim\left(\bar{P}(G)\right) = 2|E| + |E_P| + |V|$.

Proof Assume that all solutions in $\bar{P}(G)$ satisfies

$$\sum_{e\in E} c_e^x x_e + \sum_{e\in E} c_e^y y_e + \sum_{e\in E_P} c_e^Z z_e + \sum_{i\in V} c_i^w w_i = c_{rhs} \tag{40}$$

To prove the theorem, it is sufficient to show that $c_e^x = 0$, $e \in E$, $c_e^y = 0$, $e \in E$, $c_e^z = 0$, $e \in E_P$ and $c_i = 0, i \in V$. At first note that, since the zero vector is in $\bar{P}(G)$, we have $c_{rhs} = 0$. Let $(x;\, y)$ be an arbitrary vector in $P_1(G)$. It can be easily verified that $(x, y, 0, 0)$ is in $\bar{P}(G)$. Therefore, we have $\sum_{e\in E} c_e^x x_e + \sum_{e\in E} c_e^y y_e = 0$. Since by Theorem 2, $P_1(G)$ is full dimensional we have $c_e^x = 0$, $e \in E$, $c_e^y = 0$, $e \in E$ and (40) can be written as

$$\sum_{e\in E_P} c_e^z z_e + \sum_{i\in V} c_i^w w_i = 0 \tag{41}$$

Next, we show $c_i^w = 0, 0 \le i \le |V|$. Let $1 \le i \le |V|$ be arbitrary and $(\mathrm{x^i}, \mathrm{y^i}, \mathrm{z^i}, \mathrm{w^i})$ be such that $\mathrm{x^i} = \mathrm{x^{RPP}}, \mathrm{y^i} = \mathrm{y^{RPP}}, \mathrm{z^i} = 0$ and $\mathrm{w^i} = (\mathrm{w_t^i})_{\mathrm{t}\in\mathrm{V}}$, where

$$w_t^i = \begin{cases} 1, & t \ne i, \\ 0, & o.w. \end{cases}$$

It can be easily verified that $(\mathrm{x^i}, \mathrm{y^i}, \mathrm{z^i}, \mathrm{w^i})$ is in $\bar{P}(G)$ and therefore satisfies (41). This implies that $c_i^w = 0$. Since i was arbitrary we have $c_i^w = 0, 1 \le i \le |V|$, and (41) can be written as follows.

$$\sum_{e\in E_P} c_e^z z_e = 0 \tag{42}$$

It remains to show that $c_e^z = 0$, $e \in E_P$. Let $\hat{\mathrm{e}}$ be an arbitrary member of E_P and $(\mathrm{x^{\hat{e}}}, \mathrm{y^{\hat{e}}}, \mathrm{z^{\hat{e}}}, \mathrm{w^{\hat{e}}})$ be such that $\mathrm{x^{\hat{e}}} = \mathrm{x^{RPP}}, \mathrm{y^{\hat{e}}} = \mathrm{y^{RPP}}, \mathrm{z^{\hat{e}}} = (\mathrm{z_e^{\hat{e}}})$ and $\mathrm{w^i} = (\mathrm{w_t^i})_{\mathrm{t}\in\mathrm{V}}$, where

$$w_t^i = \begin{cases} 1, & t \ne i, \\ 0, & o.w. \end{cases} \quad z_e^{\hat{e}} = \begin{cases} 1, & e = \hat{e}, \\ 0, & o.w. \end{cases}$$

Clearly, $(\mathrm{x^{\hat{e}}}, \mathrm{y^{\hat{e}}}, \mathrm{z^{\hat{e}}}, \mathrm{w^{\hat{e}}})$ is in $\bar{P}(G)$ and therefore satisfies (42). By substituting this solution into (4), we conclude that $c_e^z = 0$, $e \in E_P$. This completes the proof of the theorem.

Corollary 1. If every cut set of G has at least three edges, $dim(UPLRPP(G)) = 2|E| + |E_P| + |V| - 1$

3. Algorithm description

In this section an efficient branch and cut algorithm is presented for solving UPLRPP. In the presented algorithm the parity inequalities (27) are handled implicitly as follows [1]. Suppose that node h is the current active node and (x^h, y^h, z^h, w^h) is the optimal solution of the LPR of S^h. For each $\in V$, at first we define $F = \{e \in \delta(v) \big| x_e^h \ge 0.5\}$ and $L = \{e \in \delta(v) \big| y_e^h \ge 0.5\}$. If $|F| + |L|$ is even, then we define $u_{e_1} = min\left\{min\{x_e^h \big| e \in F\}\right\}$, $min\{y_e^h \big| e \in L\}$ and$u_{e_2} =$

$max\left\{max\{x_e^h|e\in\delta(v)\setminus F\}\right\}$, $max\{y_e^h|e\in\delta(v)\setminus L\}$. In case $u_{e_1}-0.5\le 0.5-u_{e_2}$ and $e_1\in L$ we let $L=L/\{e_1\}$. On the other hand, in case $u_{e_1}-0.5\le 0.5-u_{e_2}$ and $e_1\in F$ we let $F=F/\{e_1\}$. Otherwise, if $u_{e_1}-0.5>0.5-u_{e_2}$ and $e_2\notin F$ then we let $F=F\cup\{e_2\}$. If $u_{e_1}-0.5>0.5-u_{e_2}$ and $e_2\notin L$ then we let then $L=L\cup\{e_1\}$. If $\sum_{e\in\delta(v)\setminus F}x_e^*+\sum_{e\in F\setminus L}y_e^*+\sum_{e\in F}(1-x_e^*)+\sum_{e\in L}(1-y_e^*)-1<0$ then the following inequality is generated and added to the problem;

$$x(\delta(v)\setminus F)+y(F\setminus L)\ge x(F)+y(L)-|F|-|L|+1\cdot$$

In the presented branch and cut algorithm (given in Algorithm 1), the set of constraints (28) are removed from the formulation of UPLRPP and are treated implicitly as follows. Suppose that node h is the current active node and (x^h,y^h,z^h,w^h) is the optimal solution of the LPR of S^h. Let G^hbe the capacitated undirected graph obtained from G by associating the capacity x_e^h to each $e\in E$. The algorithm proposed by Gusfield [11] is used to compute the minimum-capacity cut between every pair of nodes in G^h. In the Gusfield's algorithm, the algorithm proposed by Edmond and Karp [7] is used for computing the maximum network flow. For each pair of nodes u and v in G^h, if the capacity of the minimum cut that separates u and v is less than $2(w_u^h+w_v^h-1)$, the corresponding violated inequality is inserted to the formulation. If no violated inequality is detected, (x^h,y^h,z^h,w^h) satisfies subtour elimination inequality (28). Since the described exact algorithm for separation of connectivity inequalities is costly, the following heuristic algorithm is also used for separation of the connectivity inequalities (28). At first all connected components of $G^h$are determined. In case the number of components of G^h is grater that one, for each pair of vertices in V such that $w_u^h>0$ and $w_v^h>0$ and w_u^h and w_v^h belong to two different components of G^h, we let S to be the vertices of the component containing u. Then, the corresponding connectivity inequality (28) is checked for violation.

4. Numerical Results

In this section some tables of numerical results are presented to justify the efficiency of the presented branch and cut algorithm for solving UPLRPP. The algorithm is coded in C++ programming language. Moreover, for the implementation, the CPLEX 12.8 MIP Solver with Concert Technology is used. The computational experiments were conducted on a PC Intel Core i7 with 3.50 GHz processor and 8 GB of RAM and Linux Ubuntu operating system.

The results are obtained by testing algorithms on the UPLRPP test instances obtained by following the same ideas as in [3]. The P-RPP benchmark instances of real-world problems, originally proposed by Araoz et al. [1] consisting of 118 arc routing problems is modified for UPLRPP as follows. For each test problem x, $1\le x\le 118$, fixed costs are randomly generated in the intervals $[1,f_{max}]$, where $f_{max}=10$. Next, in each resulting UPLRPP instance, the profit of every profitable edge is the same as that of the original P-RPP instance. The characteristics of the P-RPP instances is presented in Table 1. In this table, $|V|^{min}$ and $|V|^{max}$ ($|E|^{min}$ and $|E|^{max}$) denote the minimum and maximum number of vertices (edges) in the corresponding set of instances, respectively. Moreover, the last column gives the number of instances in each set.

The presented algorithm was run on 118 instances and the average of obtained results are recorded in Table 2. In this table, the first two columns characterize the test instances, where Problem Set denotes the type of test instances, n denotes the number of instances of this type. The next three columns report the results related to the branch and cut algorithm. Here, n denotes the total number of instances, #opt, Time and #Nodes are used to show the number of problems solved to optimality, the average of computing time and the average of number of nodes of the branch and cut algorithm, respectively. #HCO (#ECO) and #Parity are used to denote the number of generated connectivity inequalities by the presented heuristic (exact) separation algorithm and the number of generated parity inequalities, respectively.

Algorithm 1. BC: The branch and cut algorithm

1: **(Initialization)** Add the initial problem S^0 to the list of active nodes L. Let $f_{best} \leftarrow \infty$ and $(x_{best}, y_{best}, \boldsymbol{z}_{best}, w_{best}) \leftarrow NULL$. Set h =0

2: **while L ≠ ∅ do**

3: **(Node selection)** Select an active problem $S^h \in L$ and remove it from L**.**

4: **(Bounding)** Solve the LP relaxation of S^h

5: **(Prune by infeasibility)** If $S^h$is infeasible, h ← h + 1 and go to 2. Otherwise, let $(\boldsymbol{x^h}, \boldsymbol{y^h}, \boldsymbol{z^h}, \boldsymbol{w^h})$ and f^h be the optimal solution and the optimal value of the linear programming relaxation of S^h, respectively.

6: **(Prune by bound)** If $f^h > f_{best}$ Set and h ← h + 1 go to 2**.**

7: **(Heuristic subtour elimination separation)** Determine the components of G^h. If the number of components of G^h is greater than one, for every two nodes u and v, if $w_u^h > 0$, $w_v^h > 0$ and u and v belong to two different component of G^h, then construct add the corresponding subtour elimination constraints to the formulation.

8: **(Exact parity separation)** Let $F = \{e \in \delta(v) | x_e^* \geq 0.5\}$ and $L = \{e \in \delta(v) | y_e^* \geq 0.5\}$. If $|F| + |L|$ is even, then let $u_{e_1} = max\{max\{x_e^* | e \in F\}\}$, $max\{y_e^* | e \in L\}$ and $u_{e_2} = max\{max\{x_e^* | e \in \delta(\mathrm{v})/\mathrm{F}\}\}$, $max\{y_e^* | e \in \delta(\mathrm{v})/\mathrm{L}\}$. If $u_{e_1} - 0.5 \leq 0.5 - u_{e_2}$ and $e_1 \in L$ then $= \mathrm{L} \setminus \{e_1\}$. If $u_{e_1} - 0.5 \leq 0.5 - u_{e_2}$ and $e_1 \in F$ then$\mathrm{F} = \mathrm{F} \setminus \{e_1\}$. If $u_{e_1} - 0.5 > 0.5 - u_{e_2}$ and $e_2 \notin F$ then $F = F \cup \{e_2\}$. If $u_{e_1} - 0.5 > 0.5 - u_{e_2}$ and $e_2 \notin L$ then $= L \cup \{e_1\}$. Add the corresponding parity inequality if it is violated.

9: **(Exact subtour elimination separation)** If the heuristic connectivity separation and exact parity separation both fail, then compute the minimum-capacity cut between every pair of nodes in G^h. For every two nodes u and v, if $2(w_u^h + w_v^h - 1)$ is greater that the capacity of the minimum cut separating u and v, add the corresponding subtour elimination constraints to the formulation.

10: **(Prune by optimality)** If $(\boldsymbol{x^h}, \boldsymbol{y^h}, \boldsymbol{z^h}, \boldsymbol{w^h})$ is an integer solution then if $f^h < f_{best}$**,**
Let $(\boldsymbol{x_{best}}, \boldsymbol{y_{best}}, \boldsymbol{z_{best}}, \boldsymbol{w_{best}}) \leftarrow (\boldsymbol{x^h}, \boldsymbol{y^h}, \boldsymbol{z^h}, \boldsymbol{w^h})$ and $f_{best} \leftarrow f^h$. h ← −h + 1 and go to 2

11: **(Branching on fractional variables)** Choose a fractional variable to branch on. Accordingly, generate and add 2 new active nodes to L. h ← h + 1 and go to 2.

12: **end while**

The numerical results show that the presented algorithm was able to solve 115 of 118 test instances. The presented algorithm solves all instances in less than a minute. Indeed, the average computing time ranges from 10 millisecond to 42 seconds. The order of executing the separation procedures has a great impact on the performance of the presented algorithm. For example, we observed that if we apply the exact connectivity separation before the exact parity separation, the computing time of some instances may rise to a few minutes. The set of problem instances with smallest computing time is G08NoRPP with the average of computing time is 10 milliseconds. The set of problem instances with largest average computing time is D27-35NoRPP with the average of computing time is 42165 milliseconds. In average, the number of generated nodes in the branch and cut algorithm increases with the size of the underlying graph. For many small sized instances, the presented branch and cut algorithm solves the problem in the root.

Table 1. Characteristics of the P-RRP instances

Problem Group	$\lvert V \rvert^{min}$	$\lvert V \rvert^{max}$	$\lvert E \rvert^{min}$	$\lvert E \rvert^{max}$	**#instances**
D0-8NoRPP	16	16	31	32	9
D9-17NoRPP	36	36	72	72	9
D18-26NoRPP	64	64	128	128	9
D27-35NoRPP	100	100	200	200	9
G0-8NoRPP	16	16	24	24	9
G9-17NoRPP	36	36	60	60	9
G18-26NoRPP	64	64	112	112	9
G27-35NoRPP	100	100	180	180	9
R0-4NoRPP	20	20	37	60	9
R5-9NoRPP	30	30	70	111	9
R10-14NoRPP	40	40	82	203	9
R15-19NoRPP	50	50	62	203	9
P01-24NoRPP	7	50	10	184	24
ALBAIDAANoRPP	102	102	160	160	1
ALBAIDABNoRPP	90	90	144	144	1

Table 2. Numerical results for instances with low fixed costs

Problem Set	**n**	**#opt**	**Time**	**#Nodes**	**#HCO**	**#ECO**	**#Parity**
D0-8NoRPP	9	9	68.8	0.0	3.8	17.7	19.0
D9-7NoRPP	9	9	305.7	47.6	9.3	52.6	68.8
D18-26NoRPP	9	8	1872.8	1708.5	14.5	329.1	250.3
D27-35NoRPP	9	9	42165	5649.5	67.6	779.3	537
G0-8NoRPP	9	9	10	0	2.9	7.2	17.6
G9-17NoRPP	9	9	76.7	17.1	7.7	60.9	58.3
G18-26NoRPP	9	9	1059.1	401.1	17.7	294.1	120.3
G27-35NoRPP	9	8	2968.2	520.1	25.9	858.3	224.7
R0-4NoRPP	5	5	47.4	19.8	7.2	10.2	70.6
R5-9NoRPP	5	5	37.7	2.0	7.3	28.0	53.0
R10-14NoRPP	5	5	250.3	227.0	29.7	37.3	177.0
R15-19NoRPP	5	5	1344	707.7	36.3	117.7	251.3
P01-24NoRPP	24	23	91.1	2.1	5.2	161.3	39.6
ALBAIDAANoRPP	1	1	2761	210	32	1952	289
ALBAIDABNoRPP	1	1	2123	166	17	1852	311

For connectivity inequalities, the number of cuts generated by the exact separation algorithm is considerably larger than that of the heuristic one. Moreover, as the size of problem becomes larger, the number of generated violated connectivity inequalities and parity inequalities becomes larger.

5. Conclusion

A combination of the facility location and Undirected Profitable Location Rural Postman Problem was considered. A compact integer programming formulation was given for the problem. Then, an exact algorithms were developed for finding the optimal solution of the problem. The presented algorithm was run on benchmark instances, adapted for the problem, and extensive numerical

Results justified the efficiency of the branch and cut algorithm.

Conflict of Interest:

The author declares that he has no conflict of interest.

References

[1] Araoz, J., Fernandez, E., and Meza, O., Solving the prize-collecting rural postman problem, European Journal of Operational Research, (2009) 196, 886-896.

[2] Amaya A, Langevin A, Trpanier M., The capacitated arc routing problem with refill points., Operations Research Letters, (2007) 35(1):45-53..

[3] C. Arbib, M. Servilio, C. Archetti, M. G. Speranza, The directed profitable location Rural Postman Problem, European Journal of Operational Research, (2014), 236, 811-819.

[4] Baldacci R, Maniezzo V., Exact methods based on node-routing formulations for undirected arc-routing problems, Networks (2006) 47(1):5260.

[5] Corberan A., Plana I., Rodriguez-Chia A. M. and Sanchis, J. M., A branch-and-cut algorithm for the maximum benefit Chinese postman problem, Mathematical Programming, (2013) 141, 2148.

[6] S. H. H. Doulabi, A. Sei_, Lower and upper bounds for location-arc routing problems with vehicle capacity constraints, European Journal of Operational Research, (2013) 224, 189-208.

[7] Edmonds, Jack, Karp, Richard M., Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM, (1972) 19(2), 248-264.

[8] Fernandez, E., Laporte, G., Rodriguez-Pereira, J. (2019). Exact Solution of Several Families of Location-Arc Routing Problems. Transportation Science, (2019) 53(5), 1313-1335.

[9] Ghiani, G., and Laporte, G., Eulerian location problems. Networks, (1999) 34, 291-302.

[10] Ghiani G. and Laporte G., A branch-and-cut algorithm for the Undirected Rural Postman Problem, Mathematical Programming, (2000), 87, 467-481.

[11] Ghiani, G., Laporte, G., Very Simple Methods for All Pairs Network Flow Analysis, SIAM Journal on Computing, (2001) 19(1), 143-155.

[12] Longo H, Aragao MPd, Uchoa E., Solving capacitated arc routing problems using a transformation to the CVRP, Computers and Operations Research, (2006) 33(6), 1823-1837.

[13] Levy L, Bodin L., The arc oriented location routing problem, INFOR, (1989), 27 (1), 74-94.

[14] Liu, T., Jiang, Z., Chen, F., Liu, R., and Liu, S., Combined location-arc routing problems: A survey and suggestions for future research. IEEE International Conference on Service Operations and Logistics and Informatics, (2008) 2, 2336-2341

[15]	Laporte G., Nickel S. and Gama F. S., Location Science, Springer International Publishing, 2016.
[16]	R. B. Lopesa, F. Plastriab, C. Ferreiraa, B. S. Santos, Location-arc routing problem: Heuristic approaches and test instances, Computers and Operations Research, (2014), 43, 309-317.
[17]	Muyldermans L. Routing, districting and location for arc traversal problems [Ph.D. thesis]. Leuven: Katholieke Universiteit Leuven; 2003.