Polyhedral analysis and a new algorithm for the length constrained K–drones rural postman problem

James Campbell[1], **Ángel Corberán**[2], **Isaac Plana**[2], **José M. Sanchis**[3]*, **Paula Segura**[3]

[1] University of Missouri-St. Louis (USA)
[2] Universitat de València (Spain)
[3] Universidad Politécnica de Valencia (Spain)

February 16, 2022

Abstract

The Length Constrained K-Drones Rural Postman Problem (LC K-DRPP) is a continuous optimization problem where a set of curved or straight lines of a network have to be traversed, in order to be serviced, by a fleet of homogeneous drones, with total minimum cost. Since the range and endurance of drones is limited, we consider here that the length of each route is constrained to a given limit L. Drones are not restricted to travel on the network, and they can enter and exit a line through any of its points, servicing only a portion of that line. Therefore, shorter solutions are obtained with "aerial" drones than with "ground" vehicles that are restricted to the network.

If a LC K-DRPP instance is digitized by approximating each line with a polygonal chain, and it is assumed that drones can only enter and exit a line through the points of the chain, an instance of the Length Constrained K-vehicles Rural Postman Problem (LC K-RPP) is obtained. This is a discrete arc routing problem, and therefore can be solved with combinatorial optimization techniques. However, when the number of points in each polygonal chain is very large, the LC K-RPP instance can be so large that it is very difficult to solve, even for heuristic algorithms. Therefore, it is necessary to implement a procedure that generates smaller LC K-RPP instances by approximating each line by a few but "significant" points and segments.

In this paper, we present a new formulation for the LC K-RPP with two binary variables for each edge and each drone representing the first and second traversals of the edge, respectively. We make a polyhedral study of the set of solutions of a relaxed formulation and prove that several families of inequalities induce facets of the polyhedron. We design and implement a branch-and-cut algorithm for the LC K-RPP that incorporates the separation of these inequalities. This B&C is the main routine of an iterative algorithm that, by solving a LC K-RPP instance at each step, finds good solutions for the original LC K-DRPP. The computational results show that the proposed method is effective in finding good solutions for LC K-DRPP, and that the branch-and-cut algorithm for the LC K-RPP outperforms the only published exact method for this problem.

Keywords: Drones, Rural Postman Problem, length constraints, facets, branch and cut.

*corresponding author: jmsanchis@mat.upv.es

1 Introduction

Given a graph representing a network, Arc Routing Problems (ARPs) consist of designing tours traversing (servicing) a set of "required" links (arc or edges) and satisfying some conditions, with total minimum cost (see [7, 8, 30]). The Chinese postman problem (CPP), the rural postman problem (RPP), and the capacitated arc routing problem (CARP) are among the best known ARPs. In these "classic" ARPs, the lines (streets, roads, pipelines etc.) that need to be serviced (cleaned, inspected, snow cleared, etc.) are represented by links of a network regardless of the shape of the real lines, because ground vehicles have to traverse a link from one endpoint to the other and cannot travel off the network. In [5] these problems were named Postman ARPs to distinguish them from Drone ARPs, in which the vehicles used are "aerial" drones. As pointed out in [4], the use of drones to perform the service involves important changes in the way of modeling and solving the ARPs. Drones can travel off the network, going directly between any two points, not necessarily endpoints of the links of the graph. This makes the problem a continuous optimization problem with an infinite and non-countable number of feasible solutions, in which it is necessary to take into account the shape of the lines to service. In order to address this problem, each line is digitized as a polygonal chain consisting of many points defined by their coordinates. If we assume that drones can only enter and exit a line through these points, the Drone ARP transforms into a Postman ARP where the segments of the polygonal chain are the required edges (see [4, 5]).

Drones (or UAVs) are increasingly being used for inspection and monitoring of infrastructure and facilities that can be modeled as networks or collections of lines. Two of the main application areas are in energy and transportation. Jordan et al. [22] provides examples of applications of drone inspection in a wide variety of areas including for power lines, railways, sewers, geographical features, bridges, buildings and structures (e.g., wind turbines). In some areas, such as power line inspection, drones are now regularly used (Rauhakallio [35]), as they provide cost effective, faster and safer inspections, and a new industry has developed to provide commercial power line inspection services. There has been limited academic research on power line inspection by drones, such as Liu et al. [27] which models the inspection of straight line segments representing power lines by drones that are launched from ground vehicles at a set of nodes on the road network. Constructive and improvement heuristics are presented to route the drones and the ground vehicles. Another energy-related drone inspection topic receiving attention is wind turbine inspection, especially for offshore wind farms, which are expensive and difficult to access ([16], [37]). Mansouri et al. [29] provides a model for drones inspecting 3D objects (with an example of a wind turbine) based on a spiral pattern of drone flights to trace a sequence of parallel slices of the 3D structure. This could be viewed as an arc routing problem where the 3D slices define circular trajectories around the outside of an object. Another interesting application for drone arc routing is the inspection and monitoring of offshore oil and gas facilities (platforms) and pipelines ([24], [21]). For example, Jones et al. [21] discuss the use of sea drones that can be autonomous vehicles that travel underwater, on the surface, or crawling on the seabed, to assist in the decommissioning of offshore oil and gas installations. (This is an important problem as there are, for example, 475 offshore oil and gas facilities being decommissioned in the UK, along with an associated >45,000 km of pipeline in the North Sea.)

In transportation systems, drone arc routing examples arise in areas such as road traffic monitoring, railroad and transit (light rail) track inspections, roadway surface inspection, monitoring railroad right-of-ways, bridge inspection and managing vegetation. Monitoring

road traffic ([26], [28], [23]), roadways ([31]) and rail lines ([39], [33], [38], [40]) provide important applications for drone arc routing along infrastructure naturally modeled with curved line features. Drone fleets are becoming common among major railroads (e.g., [19]), and even for some regulatory agencies ([6]). Bridges and buildings associated with transportation systems also provide opportunities for drone arc routing to ensure efficient inspections ([31], [33], [38]).

With the growing regulatory approvals of BVLOS (beyond visual line of sight) drone operations, inspection and monitoring of large scale transportation and energy transmission systems that stretch over hundreds of km becomes feasible using long range drones. On a smaller scale, representing structures such as bridges, buildings, oil platforms or wind turbines as 3D networks of linear features creates 3D arc routing problems for drones as well (whether in the air or underwater).

One recent paper that includes drone arc routing as a special case is Puerto and Valverde [34], in which a drone route is designed to visit parts or all of a set of dimensional facilities, including polygonal chains. This is a generalization of the crossing postman problem ([17]), but if the problem includes only polygonal chains where 100% of the length of each must be visited, then it can be viewed as a single drone RPP. Amorosi et al. [1] extends this model to include a ground vehicle that serves as a moving base for the drone with limited range.

The two papers most directly related to this work are [4] and [5]. In [4], the differences between postman ARPs and Drone ARPs are highlighted, and the Drone RPP is introduced. For this problem, a sequential algorithm is proposed that solves a larger RPP instance at each step. It begins with the RPP instance where each line is represented by a single segment. At each iteration, each segment is divided in two by adding an intermediate point taken from the original line. Also in [4] the Length Constrained K-drones Rural Postman Problem (LC K-DRPP) is defined and briefly discussed, and it is studied further in [5]. There, a "directed" formulation for the Length Constrained K-vehicles Rural Postman Problem (LC K-RPP), with two variables x_{ij}^k and x_{ji}^k for each edge (i, j) and each drone k, is proposed. Based on this formulation, a branch-and-cut procedure is presented, but it is tested only in LC K-RPP instances with a single segment representing each line. The computational results show the difficulty of the problem, as only the smallest instances are solved optimally and the final lower bounds are far from the upper bounds. A matheuristic algorithm is also proposed for the solution of the LC K-DRPP, which iteratively finds good solutions for some instances of LC K-RPP with an increasing number of intermediate points, but in a more clever way than the one proposed in [4]. Instead of adding an intermediate point for all the segments, the procedure only does it for the most "promising" ones. Also, the procedure removes some points added in previous iterations that have been proven useless.

The contribution of this work is multiple. We propose a formulation for the LC K-RPP based on the idea, exposed in [10] for the maximum benefit CPP (MBCPP), of using two binary variables for each edge that indicate the first and second traversals of the edge, respectively, for each drone. We carry out a polyhedral study of the set of solutions of a relaxed formulation, and we present some families of valid inequalities and the conditions for them to induce facets of the polyhedron. Based on this formulation, we design and implement a branch-and-cut algorithm, with separation procedures for the new inequalities. We integrate this B&C into a solution procedure for LC K-DRPP with a sequential scheme that iteratively solves instances of LC K-RPP generated by adding and removing some intermediate points in a way that is a refinement of the one presented in [5]. The computational results confirm

the benefits of our approach, clearly improving the results reported in [5].

The rest of the paper is structured as follows. In Section 2 we present the LC K-DRPP and the LC K-RPP, and propose a new formulation for the last problem. The polyhedral study of the set of solutions of a relaxed formulation of the LC K-RPP is done in Section 3, where several families of inequalities are proved to be facet inducing for the defined polyhedron. In Section 4 we present a branch-and-cut algorithm for the LC K-RPP that incorporates new separation procedures for the proposed inequalities. A global algorithm that produces good solutions for the LC K-DRPP is described in Section 5. At each iteration, the algorithm solves a different LC K-RPP instance using the branch-and-cut algorithm described before. The computational results in Section 6 show that the global algorithm is effective in finding good solutions for the LC K-DRPP, and that the branch and cut for the LC K-RPP outperforms the only published exact method for this problem. Finally, some conclusions are presented in Section 7.

2 The Length Constrained K–Drones RPP and K–RPP

The LC K-DRPP, presented in [4] and studied in [5], is defined as follows: "Given a set of lines, each one with an associated service cost, and a point called the depot, assuming that the cost of deadheading between any two points is the Euclidean distance, and given a constant L, find a set of drone routes starting and ending at the depot and with lengths no greater than L such that they jointly traverse all the given lines completely with minimum total cost".

Given that drones can travel off the network and go directly between any two points, not necessarily the endpoints of the required lines, the LC K-DRPP is a continuous optimization problem in which the shape of the lines to service must be taken into account. To deal with this problem, the LC K-DRPP instances are digitized, that is, each line to be served is described as a polygonal chain, with a sequence of points given by their coordinates. If we assume that drones can only enter and exit a line through these points, we have an instance of the LC K-RPP, a postman ARP, defined on an undirected graph $G = (V, E) = (V, E_R \cup E_{NR})$. The set of vertices V contains the points of the polygonal chains plus the depot, denoted by 1. The set of required edges E_R contains the segments of the polygonal chains, while the set of non-required edges E_{NR} contains an edge between each pair of vertices in V, i.e., (V, E_{NR}) is a complete graph. Note that all the vertices in V, except maybe the depot, are incident to, at least, one required edge.

Each $e \in E_R$ (a segment) has a service cost $c^s_e \geq 0$ that is proportional to the service cost of the line, so that the sum of the costs of the segments that approximate a line is equal to the service cost of the line. Each non-required edge $e \in E_{NR}$ has a deadheading cost $c_e \geq 0$ equal to the cost of traveling directly between its endpoints (e.g., the Euclidean distance computed from their coordinates). Note that each $e \in E_R$ has a parallel non-required edge, which we denote as e', and we assume that the cost of traveling while servicing the required segment (i, j) is greater than or equal to the cost of flying directly from i to j ($c^s_e \geq c_{e'}$). E'_{NR} represents the set of non-required edges parallel to a required one, while $E''_{NR} = E_{NR} \setminus E'_{NR}$. The goal of the LC K-RPP is to find K tours (closed walks starting and ending at the depot) with length no greater than L, that jointly traverse (and service) all the required edges, with minimum total cost.

We want to point out that in the LC K-RPP instances the non-required edges form a complete graph, whereas in postman ARPs the graph often corresponds to a sparse network. To make a more general study, in the following sections we do not assume that the edges in E_{NR} form a complete graph, although we keep the assumption that in E there is an edge e' parallel to each $e \in E_R$.

The LC K-RPP can be formulated using a binary variable x_e^k for each edge $e \in E_R$ and for each drone $k \in \{1, \ldots, K\}$, and two binary variables x_e^k and y_e^k for each edge $e \in E_{NR}$ and for each drone $k \in \{1, \ldots, K\}$. Variable x_e^k for each edge $e \in E_R$ takes the value 1 if e is traversed (and serviced) by drone k and 0 otherwise. Variables x_e^k and y_e^k for each edge $e \in E_{NR}$ take the value 1 if e is traversed once or twice, respectively, by drone k and 0 otherwise. In other words, variables x_e^k and y_e^k represent the first and second traversal by drone k of the non-required edge e. The use of these variables is inspired by the work in [10] for the MBCPP.

Note that, for each drone, there are three variables to represent the traversals between the two endpoints i,j of a required edge e. The reason is that we need to distinguish among traversing e while serving it (with a cost c_e^s) and deadheading e' once or twice (with a cost $c_{e'} \leq c_e^s$). Although in all the optimal solutions the three variables will never be non-zero simultaneously (see Theorem 1), we need the three variables to state the objective function of the problem formulation.

Theorem 1 *For each $e \in E_R$ and its parallel edge $e' \in E'_{NR}$, and for each $k \in \{1, \ldots, K\}$, the inequality $x_e^k + y_{e'}^k \leq 1$ is satisfied by any optimal solution of the LC K-RPP (if $c_{e'} > 0$).*

Proof: If a solution satisfies $x_e^k + y_{e'}^k = 2$, necessarily $x_{e'} = 1$ (because $x_{e'} \geq y_{e'}$ holds) and the drone travels three times between i and j, so we can remove two copies and get a better feasible solution. ♦

In this paper we use the following notation. Given two subsets of vertices $S, S' \subseteq V$, $(S : S')$ denotes the edge set with one endpoint in S and the other one in S'. Given a subset $S \subseteq V$, let us denote $\delta(S) = (S : V \setminus S)$ and $E(S) = (S : S)$. For simplicity, when $S = \{i\}$, $i \in V$, we write $\delta(i)$ instead of $\delta(\{i\})$. For any subset $F \subseteq E = E_R \cup E_{NR}$, we denote $F_R = F \cap E_R$ and $F_{NR} = F \cap E_{NR}$, and, for simplicity, we write $\delta_R(S)$ instead of $\delta(S)_R$. Moreover, given a vector x indexed on the edges set E and a subset of edges $F \subset E$, $x(E)$ denotes $\sum_{e \in F} x_e$, while $(x + y)(F)$ and $(x - y)(F)$ represent $x(F) + y(F)$ and $x(F) - y(F)$, respectively.

The LC K-RPP can be formulated as follows:

$$\text{Minimize} \quad \sum_{k=1}^{K} \left(\sum_{e \in E_R} c_e^s x_e^k + \sum_{e \in E_{NR}} c_e \left(x_e^k + y_e^k \right) \right)$$

s.t.

$$\sum_{e\in\delta_R(i)} x_e^k + \sum_{e\in\delta_{NR}(i)} \left(x_e^k + y_e^k\right) \equiv 0 \pmod 2, \qquad \forall i\in V,\quad \forall k=1,\ldots,K \tag{1}$$

$$\sum_{e\in\delta_R(S)} x_e^k + \sum_{e\in\delta_{NR}(S)} \left(x_e^k + y_e^k\right) \geq 2x_f^k, \qquad \forall S\subseteq V\setminus\{1\}, \forall f\in E(S), \forall k=1,\ldots,K \tag{2}$$

$$\sum_{e\in E_R} c_e^s x_e^k + \sum_{e\in E_{NR}} c_e\left(x_e^k + y_e^k\right) \leq L, \qquad \forall k=1,\ldots,K \tag{3}$$

$$\sum_{k=1}^{K} x_e^k \geq 1, \qquad \forall e\in E_R \tag{4}$$

$$x_e^k \geq y_e^k, \qquad \forall e\in E_{NR},\quad \forall k=1,\ldots,K \tag{5}$$

$$x_e^k \in \{0,1\}, \qquad \forall e\in E_R,\quad \forall k=1,\ldots,K \tag{6}$$

$$x_e^k, y_e^k \in \{0,1\}, \qquad \forall e\in E_{NR},\quad \forall k=1,\ldots,K \tag{7}$$

Constraints (1) force each drone to visit each vertex an even number of times, possibly zero. Conditions (2) ensure each single route is connected and connected to the depot, while constraints (3) guarantee that the length of each route does not exceed L. The traversal of all the required edges is ensured by constraints (4). Constraints (5) guarantee that a second traversal of a non required edge by a drone can only occur when it has been traversed previously by this drone. Constraints (6) and (7) are the binary conditions for the variables.

Note that constraints (1) are not linear, although they could be linearized by introducing additional general integer variables, z_i^k, as follows

$$\sum_{e\in\delta_R(i)} x_e^k + \sum_{e\in\delta_{NR}(i)} \left(x_e^k + y_e^k\right) = 2z_i^k.$$

Instead, as we will see in Section 3.2, we will introduce some linear inequalities on the variables x_e^k, y_e^k (parity and KC inequalities). Note also that the previous formulation allows "not feasible" solutions with isolated subtours of non-required edges, although these solutions are not optimal.

3 Polyhedral study of the K-RPP

In this section we study a polyhedron of solutions associated with the proposed formulation for the LC K-RPP. Since this formulation is based on the one presented in [10] for the MBCPP, some of the inequalities proposed in that article for the MBCPP can be transformed into valid inequalities for the K-RPP (see Theorem 2). However, to prove that these new inequalities, and the inequalities in the K-RPP formulation, are facet inducing, we must to first discuss the polyhedron associated with the K-RPP in the special case when K=1.

As with other routing problems with several vehicles, determining the dimension of the polyhedron defined as the convex hull of the LC K-RPP solutions is a very difficult task, because it depends also on the edge costs c_e and c_e^s, the number of vehicles K, and the length limit L. Even in some cases, the polyhedron could be empty. However, if we remove the constraints (3) that limit the length of each route, the problem becomes the K vehicles Rural

Postman Problem (K-RPP), whose polyhedron can indeed be studied. This is interesting because some of its facets could also be facets of the original LC K-RPP polyhedron, and it is a way of guaranteeing the strength of the constraints in the formulation and of the valid inequalities we can find.

Let a *K-RPP solution* denote any set of K tours on graph G starting and ending at the depot and jointly servicing (traversing) all the required edges. Associated with each K-RPP solution we can consider:

(a) K incidence vectors $(x^k, y^k) \in \mathbb{Z}^{2|E_{NR}|+|E_R|}$, one for each tour, where variables x_e^k take the value 1 if edge e is traversed once, variables y_e^k take the value 1 if edge e is traversed twice, and

(b) K support graphs $(V, E^{(x^k,y^k)})$, one for each tour, where $E^{(x^k,y^k)}$ contains one copy of edge $e \in E$ for each variable $x_e^k = 1$ or $y_e^k = 1$.

Note that the support graphs are even and connected. Conversely, any even and connected subgraph of G corresponds to a tour on G. In fact, an incidence vector or a subgraph may correspond to several different closed walks, but all of them have the same cost and they can be easily computed (with the Hierholzer algorithm [20], for example). Hence, and for the sake of simplicity, we will let "tour on G" denote the closed walk, its incidence vector, and its corresponding support graph.

In what follows, we study the K-RPP polyhedron defined as the convex hull of the vectors $(x^1, y^1, x^2, y^2, \ldots, x^K, y^K) \in \mathbb{Z}^{K(2|E_{NR}|+|E_R|)}$ corresponding to K-RPP solutions on G, that is, (a) each (x^k, y^k), $k = 1, \ldots, K$, is a tour on G starting and ending at the depot, and (b) all the required edges are traversed by, at least, one drone. Let *K-RPP(G)* denote this polyhedron. Note that we do not consider inequalities $x_e^k + y_{e'}^k \leq 1$ from Theorem 1, and we use inequalities (4) instead of $\sum_{k=1}^K x_e^k = 1$. Although the feasible solutions that satisfy $x_e^k + y_{e'}^k = 2$ or $\sum_{k=1}^K x_e^k > 1$ cannot be optimal solutions (if $0 < c_{e'} < c_e^s$), we will use these solutions to make the proofs in Section 3.2.

To make this polyhedral study, we need some results presented in [10] for the MBCPP. Given an undirected connected graph $G = (V, E)$, where $1 \in V$ represents the depot, with two benefits for each edge $e \in E$ associated with the first and the second traversals of e, respectively, the MBCPP consists of finding a tour starting from the depot, traversing some of the edges in E at most twice and returning to the depot, with maximum total benefit. The MBCPP is formulated with two binary variables x_e and y_e for each edge $e \in E$ representing the first and second traversal of e, respectively. It is shown that the convex hull of all the MBCPP tours, i.e., the vectors (x, y) satisfying

$$\sum_{e \in \delta(i)} \big(x_e + y_e\big) \equiv 0 \pmod 2, \qquad \forall i \in V \tag{8}$$

$$\sum_{e \in \delta(S)} \big(x_e + y_e\big) \geq 2x_f, \qquad \forall S \subset V \setminus \{1\}, \quad \forall f \in E(S) \tag{9}$$

$$x_e \geq y_e, \qquad \forall e \in E \tag{10}$$

$$x_e, y_e \in \{0, 1\}, \qquad \forall e \in E, \tag{11}$$

is a full dimensional polytope and several families of valid and facet-inducing inequalities are described.

In this paper we study the K-RPP formulated with only one variable associated with each required edge, while, if we consider the MBCPP on the same graph, we have two variables for each edge, including the required ones. Nevertheless, given a K-RPP solution $(x^1, y^1, x^2, y^2, \ldots, x^K, y^K) \in \mathbb{Z}^{K(2|E_{NR}|+|E_R|)}$, each single route (x^k, y^k) is a closed walk starting and ending at the depot, and it can be completed with variables $y_e = 0$ for each $e \in E_R$ to obtain a MBCPP tour. Hence we have the following theorem:

Theorem 2 *Let $f(x, y) \geq \alpha$ a valid inequality for the MBCPP on graph G. By removing all the variables y_e, $e \in E_R$, the resulting inequality $f(x^k, y^k) \geq \alpha$ is valid for the K-RPP, for each drone $k \in \{1, \ldots, K\}$.*

For example, from inequalities (9) we obtain inequalities (2). Furthermore, from several families of valid inequalities for the MBCPP, namely parity, p-connectivity and K-C inequalities, we will obtain valid inequalities for the K-RPP (see Sections 3.2.1, 3.2.2, and 3.2.3).

Besides some results from the MBCPP, we need some polyhedral results from the 1-RPP for the proofs for the general case $K \geq 2$. These results are detailed in [12] and summarized in the next section.

3.1 The 1-RPP polyhedron

The K-RPP for $K = 1$, or 1-RPP, is the well known RPP ([14], [15], [18], [9], [13], [36]) but with some special features. First, it is defined on a graph that has a non-required edge parallel to each required one. Second, the problem is formulated with three variables associated with the traversal of a required edge e and its parallel non-required one e'. Note that for the 1-RPP, $x_e = 1$ holds for each $e \in E_R$ and, from Theorem 1, we obtain that $y_{e'} = 0$ for each $e' \in E'_{NR}$ in all optimal 1-RPP tours. Hence, these variables could be removed from the formulation. However, since this is not true for $K > 1$, we will keep these variables because they are necessary in the proofs of the polyhedral study of the K-RPP(G) for $K > 1$. Hence, we will accept feasible (but not optimal) solutions with some variables $y_{e'} = 1$.

In the same way, although it is natural in problems with drones to assume that all the vertices (except, maybe the depot) are incident with required edges, we will not consider here this assumption to make a more general study. Therefore, in this section we consider an undirected and connected graph $G = (V, E)$, with a set $E_R \subset E$ of required edges, and where the set V_R formed with the vertices incident with some edge in E_R plus the depot, is not necessarily equal to V. We assume $E = E_R \cup E'_{NR} \cup E''_{NR}$, where E'_{NR} is the set of non-required edges parallel to an edge in E_R.

Let a 1-RPP tour denote to a closed walk on graph G starting and ending at the depot and servicing all the required edges. As before, we will use 1-RPP tour also to denote its incidence vector $(x, y) \in \mathbb{Z}^{2|E_{NR}|+|E_R|}$ and its support graph. The polyhedron *1-RPP(G)* is defined as the convex hull of all the 1-RPP tours in G. Note that the set of constraints of the K-RPP formulation, adapted to the case $K = 1$, is:

$$\sum_{e\in\delta_R(i)} x_e + \sum_{e\in\delta_{NR}(i)} \big(x_e + y_e\big) \equiv 0 \pmod 2, \qquad \forall i \in V \tag{12}$$

$$\sum_{e\in\delta_R(S)} x_e + \sum_{e\in\delta_{NR}(S)} \big(x_e + y_e\big) \geq 2x_f, \qquad \forall S \subseteq V\setminus\{1\}, \forall f \in E(S), \tag{13}$$

$$x_e = 1, \qquad \forall e \in E_R \tag{14}$$

$$x_e \geq y_e, \qquad \forall e \in E_{NR} \tag{15}$$

$$x_e, y_e \in \{0,1\}, \qquad \forall e \in E_{NR}. \tag{16}$$

In [12], it is proved the following theorem about the dimension of polytope 1-RPP(G).

Theorem 3 *If (V, E_{NR}) is a 3-connected graph, then, dim(1-RPP(G))= $2|E_{NR}|$.*

Moreover, in [12], conditions under which some of the constraints in the formulation and other valid inequalities define facets of 1-RPP(G) are studied. Here we summarize these results, where we assume that graph (V, E_{NR}) is 3-connected:

Theorem 4 *The following are facet-inducing inequalities for 1-RPP(G):*

1. *Inequality $y_e \geq 0$, $\forall e \in E_{NR}$,*
2. *inequality $x_e \leq 1$, $\forall e \in E_{NR}$, and*
3. *inequality $x_e \geq y_e$, $\forall e \in E_{NR}$, if graph $(V, E_{NR}\setminus\{e\})$ is 3-edge connected.*

Regarding the connectivity inequalities (13), note that, given that $x_e = 1$ for all $e \in E_R$, if $\delta_R(S) \neq \emptyset$ they are obviously satisfied.

Theorem 5 *Let $S \subseteq V \setminus \{1\}$ such that $E_R(S) = \delta_R(S) = \emptyset$. Let $f \in E(S)$ ($f \in E''_{NR}$). The connectivity inequality (13), which now takes the form*

$$(x + y)(\delta(S)) \geq 2x_f, \tag{17}$$

is facet-inducing for 1-RPP(G) if subgraph $(S, E_{NR}(S))$ is 3-edge connected and either $V\setminus S = \{1\}$ or subgraph $(V\setminus S, E_{NR}(V\setminus S))$ is 3-edge connected.

If, in addition, $\delta_R(S) \neq \emptyset$ holds, then inequalities (13) are dominated by

$$(x + y)(\delta(S)) \geq 2,$$

which are also facet-inducing for 1-RPP(G).

In the remainder of this section, we describe several families of valid inequalities for the 1-RPP proposed in [12]: parity, p-connectivity and K-C inequalities.

Parity inequalities

In [10], the following constraints, which generalize the well known co-circuit inequalities ([2]), were proposed for the MBCPP:

$$(x-y)(\delta(S)\setminus F) \geq (x-y)(F) - |F| + 1, \quad \forall S \subset V,\ \forall F \subseteq \delta(S) \text{ with } |F| \text{ odd}. \tag{18}$$

If $\delta_R(S) \subseteq F$, after removing all the variables y_e, $e \in E_R$, and replacing all the variables x_e, $e \in E_R$, by one, inequalities (18) can be writen as:

$$(x-y)(\delta(S)\setminus F) \geq (x-y)(F_{NR}) - |F_{NR}| + 1, \quad \forall S \subset V,\ \forall F \subseteq \delta(S) \text{ with } |F| \text{ odd}, \tag{19}$$

which are valid for the 1-RPP from Theorem 2. These inequalities cut off (infeasible) solutions in which there is a cut-set with an odd number of edges traversed exactly once (these edges define the set F) and the other edges are traversed twice or none.

Theorem 6 *Parity inequalities (19), for all $S \subset V$, $F \subseteq \delta(S)$ with $|F|$ odd and $\delta_R(S) \subseteq F$, are facet-inducing for 1-RPP(G) if subgraph $(S, E_{NR}(S))$ is 3-edge connected and either $V \setminus S = \{1\}$ or subgraph $(V \setminus S, E_{NR}(V \setminus S))$ is 3-edge connected.*

p-connectivity inequalities

These constraints were introduced in [10] to cut off some fractional solutions in which several edges forming a cycle have variables $x_e = y_e = 0.5$. They are defined as follows.

Let $\{S_0, \ldots, S_p\}$ be a partition of V such that $\delta(S_i) \cap E_R = \emptyset$ for all i (see Figure 1(a), where a triangle represents the depot and a required edge is represented within some sets $E(S_i)$). Assume we divide the set $\{0, 1, \ldots, p\} = \mathcal{R} \cup \mathcal{N}$ (from 'Required' and 'Non-required') in such a way that

- $i \in \mathcal{R}$ if either $1 \in S_i$ or $E_R(S_i) \neq \emptyset$ (note that $1 \leq |\mathcal{R}| \leq p+1$), and
- $i \in \mathcal{N}$ if $1 \notin S_i$ and $E_R(S_i) = \emptyset$ (note that $0 \leq |\mathcal{N}| \leq p$, and $|\mathcal{R}| + |\mathcal{N}| = p+1$),

and select one edge $e_i \in E(S_i)$ for every $i \in \mathcal{N}$. Note that $e_i \in E''_{NR}$. Given that the p-connectivity inequality

$$(x+y)(\delta(S_0)) + 2 \sum_{1 \leq r < t \leq p} x(s_r : S_t) \geq 2 \sum x_{e_i} \tag{20}$$

is valid for the MBCPP ([10]), after replacing all the variables x_e, $e \in E_R$ by one, we obtain the following p-connectivity inequality valid for the 1-RPP:

$$(x+y)(\delta(S_0)) + 2 \sum_{1 \leq r < t \leq p} x(S_r : S_t) \geq 2 \sum_{i \in \mathcal{N}} x_{e_i} + 2\,(|\mathcal{R}| - 1) \tag{21}$$

This inequality with $p = 2$ and $|\mathcal{N}| = 1$ is represented in Figures 1(b) and 1(c), where for each pair (a, b) associated with an edge e, a and b represent the coefficients of x_e and y_e, respectively.

Theorem 7 *p-connectivity inequalities (21) are facet-inducing for 1-RPP(G) if subgraphs $(S_i, E_{NR}(S_i))$, $i = 0, \ldots, p$, are 3-edge connected, $|(S_0 : S_i)| \geq 2$, $\forall\ i = 1, \ldots, p$, and the graph induced by $V \setminus S_0$ is connected.*

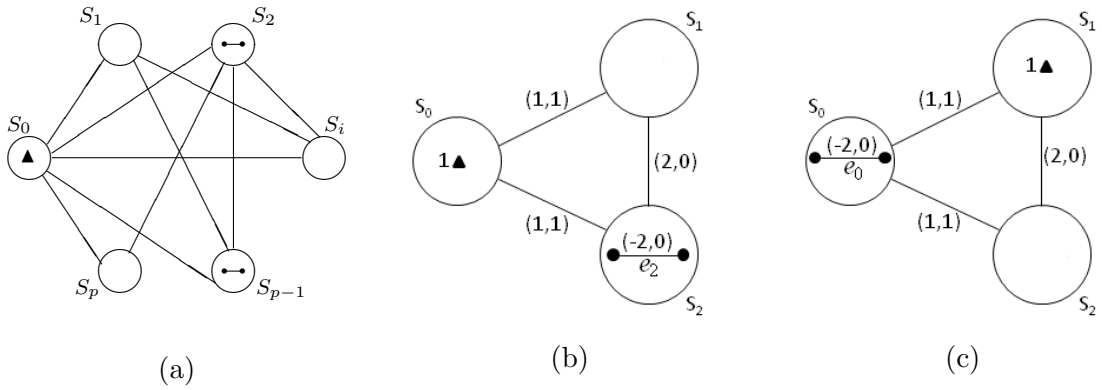

Figure 1: p-connectivity inequalities.

K-C inequalities

K-C inequalities are a well-known family of inequalities for ARPs, which jointly address the parity and connectivity conditions of the solutions.

Let $\{S_0, \ldots, S_K\}$, with $K \geq 3$, be a partition of V such that $\delta(S_i) \cap E_R = \emptyset$ for all $i = 1, 2, \ldots, K-1$. Assume we divide the set $\{1, \ldots, K-1\} = \mathcal{R} \cup \mathcal{N}$ (from 'Required' and 'Non-required') in such a way that

- $i \in \mathcal{R}$ if either $1 \in S_i$ or $E_R(S_i) \neq \emptyset$ (note that $0 \leq |\mathcal{R}| \leq K-1$), and
- $i \in \mathcal{N}$ if $1 \notin S_i$ and $E_R(S_i) = \emptyset$ (note that $0 \leq |\mathcal{N}| \leq K-1$, and $|\mathcal{R}| + |\mathcal{N}| = K-1$),

and select one edge $e_i \in E(S_i)$ for every $i \in \mathcal{N}$. Note that $e_i \in E''_{NR}$. Let $F \subseteq (S_0 : S_K)$ be a set of edges, with $|F| \geq 2$ and even, and $(S_0 : S_K)_R \subseteq F$. The K-C inequalities are as follows:

$$(K-2)(x-y)\Big((S_0 : S_K) \setminus F\Big) - (K-2)(x-y)(F_{NR}) +$$
$$+ \sum_{\substack{0 \leq i < j \leq K \\ (i,j) \neq (0,K)}} \Big((j-i)x(S_i : S_j) + (2-j+i)y(S_i : S_j)\Big) - 2\sum_{i \in \mathcal{N}} x_{e_i} \geq 2|\mathcal{R}| - (K-2)|F_{NR}| \quad (22)$$

The coefficients and structure of the K-C inequalities are shown in Figure 2, where we assume $\mathcal{R} = \{1, \ldots, |\mathcal{R}|\}$ and $\mathcal{N} = \{|\mathcal{R}|+1, \ldots, K-1\}$. Edges in F (required and non-required, if any) are represented by thick lines. For each pair (a, b) associated with an edge e, a and b represent the coefficients of x_e and y_e, respectively. From Theorem 2, K-C inequalities (22) are valid for 1-RPP(G) because they are obtained from the corresponding K-C inequalities for the MBCPP after replacing each x_e by one and by removing the y_e variables for all the required edges e.

Theorem 8 *K-C inequalities (22) are facet-inducing for 1-RPP(G) if subgraphs $(S_i, E_{NR}(S_i))$, $i = 0, \ldots, K$, are 3-edge connected, $|(S_i : S_{i+1})| \geq 2$ for $i = 0, \ldots, K-1$, and $|F_R| \geq 2$.*

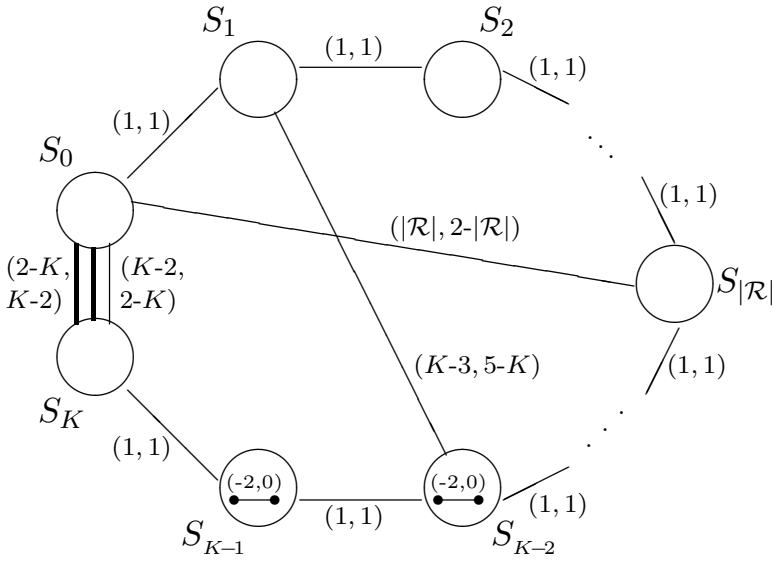

Figure 2: Coefficients of the K-C inequality

3.2 The K-RPP polyhedron ($K \geq 2$)

In this section we study the K-RPP with $K \geq 2$ on graph $G = (V, E) = (V_R, E_R \cup E'_{NR} \cup E''_{NR})$. Recall that K-RPP(G) denotes the polytope defined as the convex hull of the vectors $(x^1, y^1, x^2, y^2, \ldots, x^K, y^K) \in \mathbb{Z}^{(2|E_{NR}|+|E_R|)K}$ corresponding to K-RPP solutions.

Theorem 9 *If (V, E_{NR}) is a 3-connected graph, then K-RPP(G) is a full-dimensional polyhedron, i.e., dim(K-RPP(G))= $K(2|E_{NR}| + |E_R|)$.*

Proof: See the Appendix. ♦

In the following, we will assume that (V, E_{NR}) is a 3-connected graph and thus K-RPP(G) is full-dimensional. Therefore every facet of the polyhedron is induced by a unique inequality (except scalar multiples).

The proofs of the following theorems can be found in the Appendix.

Theorem 10 *Inequality $y_e^k \geq 0$, for each edge $e \in E_{NR}$, and for each drone $k \in \{1, 2, \ldots, K\}$, is facet-inducing of K-RPP(G).*

Theorem 11 *Inequality $x_e^k \leq 1$, for each edge $e \in E_{NR}$, and for each drone $k \in \{1, 2, \ldots, K\}$, is facet-inducing of K-RPP(G).*

Theorem 12 *Inequality (5) $x_e^k \geq y_e^k$, for each edge $e \in E_{NR}$, and for each drone k, is facet-inducing of K-RPP(G) if graph $(V, E_{NR} \setminus \{e\})$ is 3-edge connected.*

Theorem 13 *Inequality $x_e^k \leq 1$, for each edge $e \in E_R$, and for each drone k, is facet-inducing of K-RPP(G).*

Theorem 14 *Inequality $x_e^k \geq 0$, for each edge $e \in E_R$, and for each drone k, is facet-inducing of K-RPP(G).*

Theorem 15 *Inequalities* (4), $\sum_{k=1}^{K} x_e^k \geq 1$, *for each edge* $e \in E_R$, *are facet-inducing of K-RPP(G).*

Theorem 16 *Connectivity inequalities* (2), $x^k\big(\delta_R(S)\big) + (x^k + y^k)\big(\delta_{NR}(S)\big) \geq 2x_f^k$, $\forall S \subseteq V \setminus \{1\}$, $\forall f \in E(S)$, *and* $\forall k = 1, \ldots, K$, *are facet-inducing of K-RPP(G) if subgraph* $(S, E_{NR}(S))$ *is 3-edge connected and either* $|V \setminus S| = 1$ *or subgraph* $(V \setminus S, E_{NR}(V \setminus S))$ *is 3-edge connected.*

3.2.1 Parity inequalities

Since parity inequalities (18) are valid for the MBCPP on G, from Theorem 2 we have that the following parity inequalities

$$x^k(\delta_R(S) \setminus F) + (x^k - y^k)(\delta_{NR}(S) \setminus F) \geq x^k(F_R) + (x^k - y^k)(F_{NR}) - |F| + 1, \tag{23}$$

for each drone k, any $S \subset V$, and $F \subseteq \delta(S)$ with $|F|$ odd, are valid for the K-RPP on G,

Theorem 17 *Parity inequalities* (23) *are facet-inducing of K-RPP(G) if subgraph* $(S, E_{NR}(S))$ *is 3-edge connected and either* $V \setminus S = \{1\}$, *or subgraph* $(V \setminus S, E_{NR}(V \setminus S))$ *is 3-edge connected.*

Proof: See the Appendix. ♦

3.2.2 p-connectivity inequalities

Let $\{S_0, \ldots, S_p\}$ be a partition of V. Assume that $1 \in S_d$, $d \in \{0, \ldots, p\}$ and consider one edge $e_j \in E(S_j)$ for every $j \in \{0, \ldots, p\} \setminus \{d\}$. Since the p-connectivity inequality (20) is valid for the MBCPP on G, from Theorem 2, the following inequality for each drone k

$$x^k(\delta_R(S_0)) + (x^k + y^k)(\delta_{NR}(S_0)) + 2 \sum_{1 \leq r < t \leq p} x^k(S_r : S_t) \geq 2 \sum_{i=0, i \neq d}^{p} x_{e_i}^k, \tag{24}$$

is valid for the K-RPP and will be referred to as *p-connectivity inequality.*

Theorem 18 *p-connectivity inequalities* (24) *are facet-inducing for K-RPP(G) if subgraphs* $(S_i, E_{NR}(S_i))$, $i = 0, \ldots, p$, *are 3-edge connected,* $|(S_0 : S_i)| \geq 2$, $\forall\ i = 1, \ldots, p$, *and the graph induced by* $V \setminus S_0$ *is connected, and all the edges* $e_j \in E(S_j)$ *are required edges.*

Proof: See the Appendix. ♦

3.2.3 K-C inequalities

Let $\{S_0, \ldots, S_K\}$, with $K \geq 3$, be a partition of V. We assume that the depot $1 \in (S_0 \cup S_K)$ and $E(S_i) \neq \emptyset$, for all $i = 1, 2, \ldots, K-1$. We select one edge $e_i \in E(S_i)$ for all i. Let $F \subseteq (S_0 : S_K)$ be a set of edges, with $|F| \geq 2$ and even. For each drone k, the following inequalities

$$(K-2)\,x^k\Big((S_0:S_K)_R \setminus F\Big) - (K-2)\,x^k(F_R) +$$
$$+\,(K-2)\,(x^k-y^k)\Big((S_0:S_K)_{NR} \setminus F\Big) - (K-2)\,(x^k-y^k)(F_{NR}) +$$
$$+ \sum_{\substack{0\le i<j\le K\\(i,j)\neq(0,K)}} (j-i)\,x^k(S_i:S_j)_R + \sum_{\substack{0\le i<j\le K\\(i,j)\neq(0,K)}} \Big((j-i)\,x^k(S_i:S_j)_{NR} + (2-j+i)\,y^k(S_i:S_j)_{NR}\Big) \ge$$
$$\ge 2\sum_{i=1}^{K-1} x^k_{e_i} - (K-2)|F|. \tag{25}$$

will be referred to as *K-C inequalities*. If the depot $1 \in S_d$, $d \in \{1,\dots,K-1\}$, the RHS of the *K-C inequalities* is:

$$\ge 2\sum_{\substack{i=1\\i\neq d}}^{K-1} x^k_{e_i} + 2 - (K-2)|F|. \tag{26}$$

From Theorem 2, K-C inequalities (25) and (26) are valid for the K-RPP because they are obtained from the corresponding K-C inequalities for the MBCPP (see [10]) after removing the y_e variables for all the required edges e.

Theorem 19 *K-C inequalities (25) and (26) are facet-inducing for K-RPP(G) if subgraphs $(S_i, E_{NR}(S_i))$, $i = 0,\dots,K$, are 3-edge connected, $|(S_i : S_{i+1})| \ge 2$ for $i = 0,\dots,K-1$, and $|F_R| \ge 2$ and $e_i \in E_R(S_i)$ for all i.*

Proof: See the Appendix. ♦

3.3 Other valid inequalities for the LC K-RPP

Although we have removed the constraints (3), which limit the length of each route, for the polyhedral study, they have to be taken into account when solving the problem. Based on these constraints, we present some sets of valid inequalities for the LC K-RPP, called *max-length inequalities*.

Let $F \subseteq E_R$ be a subset of required edges. Consider the General Routing Problem, GRP, defined on graph G, with required edges F and required vertex 1 (the depot), if it is not incident with an edge in F. This problem consists of finding a minimum cost tour traversing the edges of F at least once and visiting the depot. Let $\text{GRP}(F)$ be its optimal value (or a lower bound of it). If $\text{GRP}(F) > L$, then we have that the following inequalities are valid for the LC K-RPP:

$$x^k(F) \le |F| - 1, \quad \forall k \in \mathbb{K}, \tag{27}$$

which indicate that a single vehicle cannot service all the arcs in F.

Let S be the set of vertices incident with the edges in F, and suppose that $1 \notin S$. Then, at least two vehicles must enter in S, and we have

$$\sum_{k \in \mathbb{K}} \left(x^k(\delta_R(S)) + (x^k + y^k)(\delta_{NR}(S)) \right) \geq 4 \tag{28}$$

Moreover, to force two *different* vehicles to enter S, instead of a single vehicle entering S twice, we have the following inequalities

$$\sum_{k \neq k'} \left(x^k(\delta_R(S)) + (x^k + y^k)(\delta_{NR}(S)) \right) \geq 2, \qquad \forall k' \in \mathbb{K}. \tag{29}$$

The above inequalities, can be easily generalized to any value $p = \lceil \frac{\text{GRP}(F)}{L} \rceil > 1$ if $\text{GRP}(F) > (p-1)L$.

4 Branch-and-cut algorithm for the LC K-RPP

Based on the polyhedral study described in the previous section, we have designed and implemented a branch-and-cut algorithm for the LC K-RPP. This method is based on a cutting-plane algorithm that incorporates separation procedures for the inequalities presented in this article. These procedures are similar to some already proposed for other arc routing problems.

The initial LP relaxation contains inequalities (3), (4), and (5), the bounds on the variables, and a parity inequality (23) with $F = \delta(v)$ for each odd-degree vertex v. Moreover, in order to avoid equivalent solutions produced by the symmetry among the vehicles, the following symmetry-breaking inequalities are also added. Assume the required edges are ordered in descending order according to the distances between them and the depot as $e_1, e_2, \ldots, e_{|E_R|}$. Then, we add:

- $x^1_{e_1} = 1$.
- $x^k_{e_i} <= \sum_{j=1}^{i-1} x^{k-1}_{e_j}, \quad k = 2, ..., |K|, \quad i >= 2$.
- $x^k_{e_i} = 0, \quad k = i+1, ..., |K|, \quad i = 1, ..., |E_R| - 1$.

In addition, to prune the search tree, we use a bound obtained by the heuristic algorithm proposed in [5].

4.1 Separation algorithms

Given a solution of the current LP in an iteration of the cutting-plane algorithm, we describe here the separation algorithms that have been used to identify violated inequalities of the following classes: connectivity, parity, p-connectivity, and KC inequalities.

Given a fractional solution for a vehicle k, $(\bar{x}^k, \bar{y}^k)$, and a parameter $\varepsilon > 0$, we will use two support graphs, $G^k_+(\varepsilon)$ and $G^k_-(\varepsilon)$, which are the graphs induced by the edges $e \in E$ such that $\bar{x}^k(e) + \bar{y}^k(e) > \varepsilon$ and $\bar{x}^k(e) - \bar{y}^k(e) > \varepsilon$, respectively, plus the depot, if necessary.

4.1.1 Connectivity inequalities

For each vehicle k, connectivity inequalities (2) can be exactly separated in polynomial time with the following well-known algorithm. For each edge f such that $\bar{x}^k(f) > 0$, compute the minimum cut in graph G^k_+ separating edge f from the depot. If the weight of this cut is less than $2\bar{x}^k(f)$, then the corresponding inequality (2) is violated.

This algorithm is very time consuming and can produce many violated inequalities that are very similar to each other. Therefore, we avoid studying the edges close to an edge f for which a violated connectivity inequality has already been found. Thus, the number of calculated minimum cuts and inequalities added to the LP are reduced.

We also use two heuristic algorithms. The first one computes the connected components of graph $G^k_+(\varepsilon)$. For each connected component and for the edge f in it with maximum $\bar{x}^k(f)$, the corresponding inequality (2) is checked for violation.

The second is based on reducing the size of the graph on which the minimum cuts are calculated, by shrinking the connected components of the graph induced by edges with $\bar{x}^k(e) \geq 1 - \varepsilon'$ into a single vertex each, where ε' is a parameter. Then, we calculate all the minimum cuts between the vertex corresponding to the component containing the depot and the remaining ones. The corresponding connectivity inequalities are checked for violation.

4.1.2 Parity inequalities

If we replace $x - y$ by x in the parity inequalities (23) for a vehicle k, we obtain

$$x^k(\delta(S) \setminus F) \geq x^k(F) - |F| + 1, \quad \forall S \subset V, \quad \forall F \subset \delta(S) \quad \text{with } |F| \text{ odd},$$

which are the well-known cocircuit inequalities presented in [18]. They can be exactly separated in polynomial time with an algorithm (see [3], for example) based on the computation of odd minimum cuts, which can be computed with the classical Padberg-Rao procedure [32] or with the improved one by Letchford et al. [25].

For the special case where $S = \{v\}$, there is an exact and simple procedure to define the set $F \subseteq \delta(v)$ that corresponds to the most violated parity inequality inequality (see [18]).

We also use a heuristic algorithm based on the computation of the connected components of $G^k_-(\varepsilon)$. For each cut-set corresponding to a connected component, the set F is found by applying the same procedure as for $S = \{v\}$.

4.1.3 p-connectivity inequalities

For each vehicle k, we use a heuristic algorithm similar to the one proposed in [10] for the MBCPP. The algorithm starts by searching for a cut-set $(S, \bar{S})$ corresponding to a tight connectivity inequality (2) among those obtained with the connectivity separation procedures. Let us suppose that $1 \in \bar{S}$ and $S_0 = \bar{S}$. Then we compute the connected components in the subgraph induced in $G(S)$ by the edges $e \in E(S)$ with $\bar{x}^k(e) \geq 1 - \varepsilon$, where ε is a given parameter. For each pair C_i, C_j of such components, we compute

$$s_{ij} = 2\bar{x}^k(V_i, V_j) - 2\min\{\bar{x}^k(e_i), \bar{x}^k(e_j)\},$$

where V_r is the set of vertices in component C_r and e_r is the edge in C_r with the highest value of $\bar{x}^k(e)$. The value s_{ij} represents the savings in the left-hand side of the inequality obtained after shrinking components C_i and C_j. We iteratively shrink the components that maximize s_{ij} while $s_{ij} > 0$. This procedure defines sets $S_1, \ldots, S_p$. The p-connectivity inequality associated with this partition is checked for violation.

4.1.4 KC inequalities

For each vehicle k, we have implemented a heuristic algorithm that is an adaptation of the one proposed in [9] for the General Routing Problem. In that problem, the required edges determine the way in which set V is partitioned in $S_0, \ldots, S_K$. Here, we partition V in the same way but considering as required edges those with $\bar{x}^k(e) \geq 1 - \varepsilon$. Then, the set F is defined by the required edges with $\bar{y}^k(e) \leq \varepsilon'$, where $\varepsilon, \varepsilon'$ are two given parameters.

4.1.5 Max-length inequalities

Two heuristic algorithms are used to separate max-length inequalities.

The first heuristic looks for violated inequalities (27). It tries to cut fractional solutions in which, for a vehicle k, several x_e^k variables, for $e \in E_R$, take values close to 1 and another one takes a value close to 0.5. Let $\{e_1, e_2, \ldots, e_m\}$ be a set of required edges such that $x_{e_1}^k \geq x_{e_2}^k \geq \ldots \geq x_{e_m}^k \geq 0.5$. We define $F = \{e_1, e_2, \ldots, e_f\}$, where f is the maximal number such that $x^k(F) > |F| - 1 + \varepsilon$ (initially we set $\varepsilon = 0.5$), and we call "potential edges" the remaining $\{e_{f+1}, \ldots, e_m\}$. We check if GRP(F) is greater than L and, therefore, the corresponding inequality (27) is violated. Otherwise, for each potential edge $\bar{e}$, we iteratively consider the set $\overline{F} = F \cup \{\bar{e}\}$ and check if GRP$(\overline{F})$ is greater than L. Finally, if no violated inequality has been found for any set $\overline{F}$, we set $\varepsilon = 0$ and repeat the process. For each subset F (or $\overline{F}$) whose corresponding inequality (27) is violated, we look for the cutset of minimum weight between the depot and the edges in F and the corresponding max-length inequalities (28) and (29) are checked for violation.

The second heuristic looks for inequalities (28). We first construct the aggregate solution $x_e = \sum_{k=1}^K x_e^k$ and $y_e = \sum_{k=1}^K y_e^k$. The procedure starts by selecting the vertex $i \neq 1$ farthest from the depot such that the maximum flow from 1 to i is less than $2K$. Then a sequence of vertices is iteratively added in such a way that $\sum_{k=1}^K x^k(\delta_R(S)) + \sum_{k=1}^K (x^k + y^k)(\delta_{NR}(S))$ is minimum for the resulting subset S. For each subset S generated, we compute the minimum number of vehicles needed to service all the edges in $E_R(S)$ by solving the associated GRP,

and the corresponding inequality (28) is checked for violation. If a violated constraint (28) is found, at least one of the inequalities (29) is also violated and it is added. Furthermore, the corresponding inequality (27) is also checked for violation.

Given a set of edges F, the value of $\text{GRP}(F)$ is computed by solving the corresponding GRP with the branch-and-cut algorithm described in [11]. To minimize the number of GRPs solved, two lists containing the already studied sets F of edges are managed.

4.2 The cutting-plane algorithm and branching strategies

The cutting-plane algorithm applies, at each iteration, the separation procedures in this specific order:

1. The first heuristic algorithm for connectivity inequalities with $\varepsilon = 0, 0.25, 0.5$. The value of ε is increased only if the previous one results in no violated inequalities found.
2. If no violated inequalities are found so far, the second connectivity heuristic is executed for all the values $\varepsilon' = 0, 0.1, 0.2, 0.3, 0.4$. For each tight cut-set obtained, the p-connectivity heuristic separation algorithm is invoked with $\varepsilon \in \{0, 0.15, 0.3\}$.
3. The exact parity separation procedure for single vertices is applied.
4. The heuristic procedure for parity inequalities with $\varepsilon = 0, 0.25, 0.5$. The different values of ε are used only if no violated inequalities are obtained with the previous ones.
5. If no violated connectivity nor parity inequalities have been found for a given vehicle k, the separation algorithm for K-C inequalities for this vehicle is applied with parameters $(\varepsilon, \varepsilon') = (0, 0)$. If it fails, values $(0, 0.2), (0.2, 0), (0.2, 0.2)\}$ are tried.
6. Only at the root node and nodes whose depth in the search tree is a multiple of 3, we apply the max-length separation algorithms.
7. Only at the root node and if no violated connectivity, parity nor K-C inequalities have been found, we apply the exact procedure for parity inequalities.
8. Finally, only at the root node and if no violated connectivity, parity nor K-C inequalities have been found, we apply the exact procedure for connectivity inequalities. Again, for each tight cut-set obtained, the p-connectivity heuristic separation algorithm is invoked with $\varepsilon \in \{0, 0.15, 0.3\}$.

5 Global algorithm for the LC K-DRPP

Given a LC K-DRPP instance, if the number of intermediate points to which each line is approximated is large, the size of the LC K-RPP instance is huge and, therefore, very hard to solve. Hence, it is necessary to implement a procedure that iteratively generates LC K-RPP instances by approximating each line by a polygonal chain with few but "significant" points and segments. These smaller LC K-RPP instances are solved with the branch-and-cut procedure described in Section 4.

It begins by applying the heuristic algorithm described in [5] and solving with the branch-and-cut algorithm the LC K-RPP instance, called LC K-RPP(1), in which each original line is approximated by only one (required) edge without intermediate points. We will call V_o the set of endpoints of the original lines. The set of non-required edges defines a complete graph and their costs are the Eulerian distances.

From the two solutions obtained with the heuristic and the branch and cut, we generate a new LC K-RPP instance in the following way. Let W_1 be the set of vertices incident with non-required edges in any of these two solutions. Note that all the vertices in W_1 obtained from the solution of the LC K-RPP(1) instance are vertices of V_o, while some vertices in W_1 obtained from the heuristic solution can be intermediate points. From W_1 we define another set W_2 of intermediate points to generate the next LC K-RPP instance as follows.

For each line incident with an endpoint in W_1, we add to W_2 the intermediate vertex that splits the line in two segments of the same length. Furthermore, all the vertices in $W_1 \setminus V_o$ are added to W_2 to guarantee that the solution from the heuristic is a feasible solution of the new instance.

The instance with vertex set $V_o \cup W_2$, LC K-RPP(2), is generated with the segments joining vertices in $V_o \cup W_2$ as required edges and it is solved with the branch-and-cut algorithm.

Given the solution of instance LC K-RPP(i), $i \geq 2$, with vertex set $V_o \cup W_i$, we generate the instance LC K-RPP(i+1) with vertex set $V_o \cup W_{i+1}$, where W_{i+1} is defined as follows. We initialize W_{i+1} with the vertices of W_i incident with a non-required edge used by the solution. Then, for each required edge of LC K-RPP(i) having at least one endpoint incident with a non-required edge used by the solution, we add to W_{i+1} the intermediate point that splits this edge into two parts of equal length. Note that the vertices of W_i that are not used in the solution are not included in the new instance. This procedure is iteratively applied up to $i = 5$ and while the computing time does not exceed two hours.

Although with this procedure the size of the vertex sets of the instances does not increase much, the number of non-required edges is still huge, since they induce a complete graph. In order to reduce the number of non-required edges, we apply the following preprocessing procedure to each instance LC K-RPP(i), $i \geq 1$.

First, all the non-required edges (i, j) such that there is a vertex k with $c_{ij} \geq 0.99(c_{ik}+c_{kj})$ are removed. Moreover, if c_{max} is the length of the longest non-required edge, we remove those edges (i, j) such that $c_{ij} > c_{max}/3$ and there is a vertex k with $c_{ij} \geq 0.95(c_{ik} + c_{kj})$. However, for $i \geq 2$, if a non-required edge is used in the solution of LC K-RPP(i-1), it is not removed in order to guarantee that the solution of the new instance is not worse than the previous one.

Since we have removed some non-required edges in the previous preprocessing, it is possible that the solution of an instance contains two adjacent non-required edges (i, k), (k, j). If this happens, we replace them with the non-required edge (i, j) if the resulting solution is not disconnected.

6 Computational experiments

Here we will present the computational experiments carried out to assess the performance of our algorithm. First we will describe the LC-DRPP instances used, which were introduced in [5]. Then we will show the results obtained by the proposed procedure. Finally, a comparison of the branch-and-cut for the LC-RPP with the one used in [5] will be presented.

The algorithms have been implemented in C++ and all the tests have been run on an Intel Core i7 at 3.4 GHz with 32 GB RAM. The B&C uses CPLEX 12.6 MIP Solver with a single thread. CPLEX heuristic algorithms were turned off, and CPLEX's own cuts were activated in automatic mode. The optimality gap tolerance was set to zero and best bound strategy was selected.

6.1 Instances

The procedure has been tested on two sets of instances presented in [5], which were based on the ones from [4] for the Drone RPP. The first set consists of 30 *random* instances, while the second one, called *even*, contains 15 instances obtained by modifying the set of required edges of some of the *random* instances to reduce the number of odd-degree vertices. The goal in [4] to generate *even* instances was to obtain instances such that: their solutions were significantly different from those obtained for instances of classical ARPs; they were more difficult for the branch-and-cut algorithms; and the graphs of these instances resemble the shapes that appear in cutting path problems. These instances have between 18 and 92 lines and between 22 and 83 nodes (see Table 1). Each row of Table 1 corresponds to three instances (two *random* and one *even*) sharing the same grid, and shows the average number of vertices and lines.

Instance name	Original vertices	Original lines
DroneRPP56	22.3	21.3
DroneRPP66	27.0	23.3
DroneRPP58	34.0	29.6
DroneRPP68	36.6	35.0
DroneRPP77	38.6	41.6
DroneRPP510	41.6	42.0
DroneRPP610	50.0	46.6
DroneRPP79	50.3	53.6
DroneRPP88	56.3	51.0
DroneRPP710	58.3	56.6
DroneRPP89	60.3	56.3
DroneRPP99	66.3	65.3
DroneRPP810	68.6	67.0
DroneRPP910	78.6	74.6
DroneRPP1010	82.0	81.0

Table 1: Characteristics of the instances

In [5], five different values for the length limit L were generated for each instance in such

a way that the number of drones for each one ranges from 2 to 6. Hence, there are 5 LC K-DRPP instances for each one of the 45 Drone RPP instances above, resulting in a total of 225 instances. These instances are available at `http://www.uv.es/corberan/instancias.htm`.

6.2 Computational results of the global algorithm

In this section we present the results obtained with the global algorithm on the 225 LC K-DRPP instances presented before with a time limit of two hours. Table 2 summarizes these results organized by set of instances and number of vehicles. Columns 4, 5, and 6 report the average values obtained for all the instances of each group. The "Imp(%)" column shows the percentage improvement of the deadheading cost of the best solution found by the algorithm with respect to the solution provided by the heuristic proposed in [5], while columns "Time" and "#iter" present the average computing time in seconds and the average number of iterations done by the global algorithm, respectively. Column 7 gives the number of instances in which the global algorithm was able to find a better solution than the one given by the heuristic procedure. The next three columns report the same figures as columns 4 to 6, but only for those instances for which a better solution was found. Last column "#itb" gives the average number of iterations of the global algorithm needed to reach the best solution.

			All instances				Improved instances			
Type	#veh	#inst	Imp(%)	Time	#iter	#impr	Imp(%)	Time	#iter	#itb
R	2	30	0.41	3815.2	4.2	13	0.87	2842.1	4.4	3.6
	3	30	2.12	5253.2	3.2	22	2.65	5182.3	3.4	2.3
	4	30	3.38	6310.1	2.5	19	5.33	5773.5	3.2	2.5
	5	30	1.89	6959.2	1.9	14	3.77	6673.4	2.3	1.5
	6	30	0.66	7130.6	1.8	8	2.47	6819.1	3.1	2.0
E	2	15	1.05	2583.2	4.3	13	1.21	1872.9	4.8	4.1
	3	15	1.44	5922.3	3.1	10	2.16	4003.3	4.1	3.4
	4	15	2.07	6587.9	2.5	9	3.46	5070.5	3.6	2.8
	5	15	0.62	5423.2	2.0	5	1.87	3990.1	3.6	3.4
	6	15	0.38	6364.0	1.9	5	1.13	5363.6	3.6	3.4
Total		225	1.50			118	2.77			

Table 2: Improvement of the global algorithm over heuristic

As we can see in Table 2, the algorithm has improved the results provided by the heuristic in 118 out 225 instances and the average improvement obtained ranges from 0.87% to 5.33%. In the case of the *random* instances, when the number of vehicles is 2, our algorithm improves only 13 out of 30 instances, despite being able to complete the 5 iterations in most of them. This is probably due to the high quality of the solutions provided by the heuristic algorithm for these instances. Note that with 3, 4, and 5 vehicles, the number of improved solutions increases, as well as the average improvement. The results with 6 vehicles present lower improvements, but in this case this may be due to the global algorithm not being able to complete more than 2 iterations on average. Regarding the *even* instances, except those with 2 vehicles, both the number of instances in which the global algorithm improves the heuristic, as well as the percentage of improvement, are smaller. This may be due to the fact that,

as pointed out before, *even* instances are more difficult for branch-and-cut algorithms. The improvements obtained by the global algorithm require a considerable running time, but note that the number of iterations needed to reach the best solution is lower than the total number of iterations.

In order to describe with more detail the behavior of the proposed procedure, we present in Table 3 the results for a specific instance with 2 to 6 vehicles. Column "Heur" reports the deadheading cost of the solution provided by the heuristic. Columns labeled 1 to 5 show the deadheading costs of the solutions obtained in the corresponding iteration of the global algorithm. All these costs correspond to optimal solutions of the branch-and-cut for the LC K-RPP instances except those marked with an "asterisk", which are associated with the best feasible solution known when the time limit was reached. The total computing time in seconds is reported in the last column. In the instance with 2 vehicles, the algorithm is capable of doing 5 iterations in only 5 minutes, but no improvement is obtained. However, in the instances with 3 and 4 vehicles a better solution is found in each of the 5 iterations. When 5 and 6 vehicles are considered, the two hour time limit is reached without the algorithm being able to complete all 5 iterations, although the last feasible solutions found improve those of the heuristic.

		iteration number					
#veh	Heur	1	2	3	4	5	Time
2	1889.37	1889.37	1889.37	1889.37	1889.37	1889.37	294.0
3	2154.41	2060.28	1989.41	1946.42	1924.66	1913.69	526.2
4	2709.32	2634.49	2459.05	2413.75	2388.51	2377.05	6888.3
5	3369.75	3348.92	3348.91	3348.91*			7200
6	4588.93	4626.03	4399.8*				7200

Table 3: Detailed results on *random* instance DroneRPP77_1

6.3 Testing the branch-and-cut algorithm for the LC K-RPP

Although the main goal of this paper is to find good solutions for the LC K-DRPP, to do so we have studied the LC K-RPP, for which we have presented a new formulation, studied its associated polyhedron, and based on this, we have designed and implemented a sophisticated branch-and-cut algorithm. Since the LC K-RPP and its solution have an interest by themselves, in this section we test the performance of the proposed B&C (*BC2* in what follows). In order to do so, we compare it with that presented in [5] (*BC1*). In that paper, the branch and cut was run only on the LC K-RPP instances in which each original line is approximated by only one (required) edge, without intermediate points, denoted LC K-RPP(1). Therefore, we compare both branch-and-cut algorithms only on these instances, with a time limit of two hours. The results are reported in Table 4, where columns 4 and 7 show the percentage average gaps of the lower bounds obtained with both procedures with respect to the cost of the best solution known, while columns labeled "#opt" and "#ub" report, for both algorithms, the number of instances for which an optimal solution or a feasible solution has been found, respectively.

Overall, the performance of *BC2* is very good and clearly superior to that of *BC1*. In particular, it finds the optimal solution in 137 instances and obtains a feasible solution in 211

out of 225, while the *BC1* obtains 74 optimal solutions and 162 feasible ones. Regarding the gaps, in most of the cases the gap obtained by *BC2* is about half the gap by *BC1*. Moreover, in the 53 *random* instances that are optimally solved by both methods, the proposed B&C uses 367.2 seconds on average, while the other one needs 726.0 seconds. The average times corresponding to the 21 *even* instances solved by both methods are 165.1 and 1391.0 seconds, respectively.

			BC1			*BC2*		
Type	#veh	#inst	Gap (%)	#opt	#ub	Gap (%)	#opt	#ub
R	2	30	0.02	28	30	0.01	29	30
	3	30	0.95	16	30	0.23	24	30
	4	30	3.21	3	24	1.48	19	30
	5	30	5.85	3	16	2.98	11	27
	6	30	8.51	3	8	5.09	10	25
E	2	15	0.09	13	15	0.09	13	15
	3	15	1.67	5	15	0.45	11	15
	4	15	4.02	1	13	1.94	9	15
	5	15	6.59	1	6	3.71	6	12
	6	15	8.11	1	5	4.94	5	12
Total		225		74	162		137	211

Table 4: Comparison of branch-and-cut procedures on the LC K-RPP(1) instances

7 Conclusions

In this paper we have presented an iterative procedure to solve the Length Constrained K-Drones Rural Postman Problem. This is a continuous optimization problem that we discretize by approximating each required line by a polygonal chain, thus obtaining an instance of the Length Constrained K-Rural Postman Problem. In order to avoid the size of the instances growing too much, we have included only a reduced number of intermediate points to each polygonal chain. Furthermore, at each iteration these intermediate points are dynamically changed depending on the solution obtained in the previous iteration.

To solve the LC K-RPP instance at each iteration, we have designed and implemented a branch-and-cut algorithm based on a new formulation and its corresponding polyhedral study. This procedure has shown to be very effective in the solution of LC K-RPP instances from the literature and the obtained results improve those previously published. This algorithm has allowed us to obtain very good solutions for the LC K-DRPP.

One line of research we are currently working on is the study of a more general problem that combines the visit of vertices and the traversal of certain lines to deliver goods and inspect areas, for example. Future research includes exploring the dependence of energy consumption on the speed and altitude of drones.

Acknowledgements: The work by Ángel Corberán, Isaac Plana, José M. Sanchis, and

Paula Segura was supported by the Spanish Ministerio de Ciencia, Innovación y Universidades (MICIU) and Fondo Social Europeo (FSE) through project PGC2018-099428-B-I00.

The data instances used for the computational experiments, as well as the information about the best-known solutions, are available in `http://www.uv.es/corberan/instancias.htm`.

References

[1] L. Amorosi, J. Puerto, and C. Valverde, "Coordinating drones with mothership vehicles: The mothership and drone routing problem with graphs", *Computers & Operations Research*, (2021). https://doi.org/10.1016/j.cor.2021.105445.

[2] F. Barahona and M. Grötschel, "On the Cycle Polytope of a Binary Matroid", *Journal of Combinatorial Theory B* 40, 40-62 (1986).

[3] E. Benavent, Á. Corberán and J. M. Sanchis, "Linear Programming based methods for solving arc routing problems", in Arc Routing: theory, solutions and applications (M. Dror, editor), Kluwer Academic Publishers (2000).

[4] J.F. Campbell, Á. Corberán, I. Plana and J.M. Sanchis, "Drone arc routing problems", *Networks* 72, 543-559 (2018).

[5] J.F. Campbell, Á. Corberán, I. Plana, J.M. Sanchis and P. Segura, "Solving the length constrained K-drones rural postman problem", *European Journal of Operational Research* 292, 60-72 (2021).

[6] S. Chapa, "Railroad Commission launches drone fleet for inspections", *Houston Chronicle*, May 12 (2020). https://www.houstonchronicle.com/business/energy/article/Railroad-Commission-launches-drone-fleet-for-15264009.php.

[7] Á. Corberán, R. Eglese, G. Hasle, I. Plana, and J.M. Sanchis, "Arc routing problems: A review of the past, present, and future", *Networks* 77, 88-115 (2021).

[8] Á. Corberán and G. Laporte (editors), *Arc Routing: Problems, Methods, and Applications,* MOS-SIAM Series on Optimization, SIAM, Philadelphia (2014).

[9] Á. Corberán, A.N. Letchford, and J.M. Sanchis, "A Cutting-plane Algorithm for the General Routing Problem", *Mathematical Programming* 90, 291-316 (2001).

[10] Á. Corberán, I. Plana, A.M. Rodríguez-Chía, and J.M. Sanchis, "A branch-and-cut algorithm for the maximum benefit Chinese postman problem", *Mathematical Programming* 141, 21-48 (2013).

[11] Á. Corberán, I. Plana, and J.M. Sanchis, "A branch & cut algorithm for the windy general routing problem and special cases", *Networks* 49, 245-257 (2007).

[12] Á. Corberán, I. Plana, J.M. Sanchis, and P. Segura, "Polyhedral study of a new formulation for the Rural Postman Problem", *Internal Report.* DOI: 10.13140/RG.2.2.36748.85122/1. (2021). Available at `http://www.uv.es/corberan/reports.htm`.

[13] Á. Corberán, A. Romero, and J.M. Sanchis, "The mixed general routing polyhedron", *Mathematical Programming* 96, 103-137 (2003).

[14] Á. Corberán and J.M. Sanchis, "A Polyhedral Approach to the Rural Postman Problem", *European Journal of Operational Research* 79, 95-114 (1994).

[15] Á. Corberán and J.M. Sanchis, "The General Routing Problem polyhedron: Facets from the RPP and GTSP polyhedra", *European Journal of Operational Research* 108, 538-550 (1998).

[16] P. Durdevic, D. Ortiz-Arroyo, S. Li, and Z. Yang, "Vision Aided Navigation of a Quad-Rotor for Autonomous Wind-Farm Inspection", *IFAC-PapersOnLine* 52, Issue 8, 61-66 (2019).

[17] R.S. Garfinkel and I.R. Webb, "On Crossings, the Crossing Postman Problem, and the Rural Postman Problem", *Networks* 34, 173-180 (1999).

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

[19] T. Graetz and S. Lo, "How BNSF is Leading the Way for DRONE Use in Rail: An Interview with Todd Graetz Director TS, Telecomm, Technology Services, BNSF", *Defense Transportation Journal* 74, February 2018, 18-23 (2018).

[20] C. Hierholzer, "Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren", *Mathematische Annalen* 6, 30-32 (1873).

[21] D. Jones, A. Gates, V. Huvenne, A. Phillips, and B. Bett, "Autonomous marine environmental monitoring: Application in decommissioned oil fields", *Science of The Total Environment* 668, 835-853 (2019).

[22] S. Jordan, J. Moore, S. Hovet, J. Box, J. Perry, K. Kirsche, D. Lewis, D., and Z. Tse, "State-of-the-art technologies for UAV inspections", *IET Radar Sonar Navig.* 12, 151-164 (2018).

[23] M. Karaduman, A. Çinar, and H. Eren, "UAV Traffic Patrolling via Road Detection and Tracking in Anonymous Aerial Video Frames", *J Intell Robot Syst* 95, 675-690 (2019).

[24] R. Knight, "UAV Inspection At The Biggest Oil Rig In The World". December 2, 2019. https://www.microdrones.com/en/content/uav-inspection-at-the-biggest-oil-rig-in-the-world/

[25] A.N. Letchford, G. Reinelt, and D.O. Theis, "Odd minimum cut-sets and b-matchings revisited", *SIAM J. Discr. Math.* 22, 1480-1487 (2008).

[26] M. Li, L. Zhen, S. Wang, W. Lu, and X. Qu, "Unmanned aerial vehicle scheduling problem for traffic monitoring", *Computers & Industrial Engineering* 122, 15-23 (2018).

[27] Y. Liu, J. Shi, Z. Liu, J. Huang, and T. Zhou, "Two-Layer Routing for High-Voltage Powerline Inspection by Cooperated Ground Vehicle and Drone", *Energies* 12, 1385 (2019).

[28] H. Luo, P. Zhang, J. Wang, G. Wang, and F. Meng, "Traffic Patrolling Routing Problem with Drones in an Urban Road System", *Sensors* 19, 5164 (2019).

[29] S. Mansouri, C. Kanellakis, E. Fresk, D. Komuiniak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection", *Control Engineering Practice* 74, 118-131 (2018).

[30] M. C. Mourão & L. Pinto, "An updated annotated bibliography on arc routing problems", *Networks* 70, 144–194 (2017).

[31] F. Outay, H.A. Mengash, and M. Adnan, "Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges", *Transportation Research Part A* 141, 116-129 (2020).

[32] M. W. Padberg and M. R. Rao, "Odd minimum cut-sets and b-matchings", *Mathematics for Operations Research* 7, 67-80 (1982).

[33] M. Plotnikov, D. Ni, and D. Price, "The Application of Unmanned Aerial Systems In Surface Transportation - Volume II-A: Development of a Pilot Program to Integrate UAS Technology to Bridge and Rail Inspections", *Massachusetts Department of Transportation* Report 19-010 (2019).

[34] J. Puerto and C. Valverde, "Routing for unmanned aerial vehicles: touring dimensional sets", *European Journal of Operational Research* (2021). https://doi.org/10.1016/j.ejor.2021.06.061.

[35] P. Rauhakallio, "The Past, Present, and Future of Powerline Inspection Automation", *POWER*, Sept 18 (2020). https://www.powermag.com/the-past-present-and-future-of-powerline-inspection-automation/

[36] G. Reinelt and D. Theis, "A note on the Undirected Rural Postman Problem polytope", *Mathematical Programming* 106, 447-452 (2006).

[37] M. Shafiee, Z. Zhou, L. Mei, F. Dinmohammadi, J. Karama, and D. Flynn, "Unmanned Aerial Drones for Inspection of Offshore Wind Turbines: A Mission-Critical Failure Analysis", *Robotics* 10, 26 (2021). https://doi.org/10.3390/robotics10010026.

[38] E. Sherrock and K. Neubecker, "Unmanned Aircraft System Applications in International Railroads", *United States Department of Transportation* Report DOT/FRA/ORD-18/04, Final Report, February 2018 (2018).

[39] United Nations ESCAP (Economic and Social Commission for Asia and the Pacific), Working Group on the Trans-Asian Railway Network, (2019). Inspection and monitoring of railway infrastructure using aerial drones. Note by the secretariat, ESCAP/TARN/WG/2019/4.

[40] J. Wishart, T. Lennertz, and D. Hasson, "Use Cases for Unmanned Aircraft Systems (UAS) in Public Transportation Systems", *Federal Transit Administration* FTA Report No. 0176, John A. Volpe National Transportation Systems Center, December 2020 (2020).

A Appendix: Proofs of Theorems 10 to 19

Theorem 10 *If (V, E_{NR}) is a 3-connected graph, then K-RPP(G) is a full-dimensional polyhedron, i.e., dim(K-RPP(G))= $K(2|E_{NR}| + |E_R|)$.*

Proof: Consider the 1-RPP defined on G and its associated polytope 1-RPP(G). From Theorem 3, and since (V, E_{NR}) is 3-edge connected, we know that dim(1-RPP(G))$= 2|E_{NR}| = m$. Since $0 \notin$ aff(1-RPP(G)), because all its points satisfy equations (14), there exist $m+1$ affinely and linearly independent 1-RPP tours $z_1, z_2, \ldots, z_{m+1}$ on G, each $z_i = (x_i, y_i)$ satisfying $x_i(e) = 1$ for all the edges in E_R. We can assume that one of them, say $z_1 = (x_1, y_1)$ is formed with two copies of each edge in E_{NR} and then replacing one copy of each $e \in E'_{NR}$ by the required edge parallel to e. Hence, it satisfies $x_1(e) = 1$ for all $e \in E_R$, $x_1(e) = 1$, $y_1(e) = 0$ for all $e \in E'_{NR}$, and $x_1(e) = y_1(e) = 1$ for all $e \in E''_{NR}$. We can build $(m+1)K$ K-RPP solutions in the following way. One drone performs any 1-RPP tour z_j above, while the other drones perform z_1. These $(m+1)K$ solutions are depicted as the rows of the three first block rows in the matrix shown in Figure 3, where, for the sake of simplicity, we have supposed we have $K = 3$ drones.

Furthermore, we can build $K|E_R|$ more K-RPP solutions in the following way. Consider the 1-RPP solution z_1 above. For each required edge $e \in E_R$, we define the vector $t_{(e)} = (x_{(e)}, y_{(e)})$ equal to z_1 except for the entries $x_{(e)}(e) = x_{(e)}(e') = 0$, where e' denotes the non-required edge parallel to e. This vector represents a tour on graph G because its support graph is even and connected. For any required edge e, one drone performs $t_{(e)}$ while the other drones perform z_1. These $K|E_R|$ solutions are depicted as the rows of the three last block rows in the matrix shown in Figure 3, where the required edges are $\{e_1, e_2, \ldots, e_{|E_R|}\}$.

If we subtract the first row from all the other rows and then we remove the null rows, we obtain the matrix in Figure 4, where all the non-depicted values are zero, and a big zero in a block means that all the entries of this block are zero. Its $K(2|E_{NR}| + |E_R|)$ rows are linearly independent because the first -1 entry in each pair $-1, -1$ in the rows of the three last block rows is associated with each edge $e \in E_R$, and since $z_i - z_1$ takes value zero in all the required edges, it is the only non-zero entry in the column corresponding to e. Hence, we have $K(2|E_{NR}| + |E_R|) + 1$ affinely independent K-RPP solutions and we are done. ♦

As in Theorem 10 above, in the proofs of the following theorems we will represent the K-RPP solutions that we define only for $K = 3$ drones, although they can be extended to any value of K.

Note 1 The vectors $z_1, z_2, \ldots, z_{m+1}$, from the dimension of 1-RPP(G), and $t_{(e)}$, for each $e \in E_R$, defined in the proof of Theorem 9 will be used also in the proofs of the following theorems. In particular, $z_1 = (x_1, y_1)$ satisfies $x_1(e) = 1$ for all $e \in E_R$, $x_1(e) = 1$, $y_1(e) = 0$ for all $e \in E'_{NR}$, and $x_1(e) = y_1(e) = 1$ for all $e \in E''_{NR}$. For each $e \in E_R$, we define $t_{(e)} = (x_{(e)}, y_{(e)})$ equal to z_1 except for the entries $x_{(e)}(e) = x_{(e)}(e') = 0$, where $e' \in E'_{NR}$ denotes the non-required edge parallel to $e \in E_R$.

Theorem 11 *Inequality $y_e^k \geq 0$, for each edge $e \in E_{NR}$, and for each drone $k \in \{1, 2, \ldots, K\}$, is facet-inducing of K-RPP(G).*

Proof: Let us suppose first that $e \in E'_{NR}$. Consider the 1-RPP defined on G and its associated polytope 1-RPP(G). Given that $y_e \geq 0$ is facet-inducing of 1-RPP(G) (Theorem 4), there exist $m = 2|E_{NR}|$ affinely independent 1-RPP tours $w_1, w_2, \ldots, w_m$ on G, each $w_i = (x_i, y_i)$, satisfying $x_i(a) = 1$ for all $a \in E_R$, and $y_i(e) = 0$. Consider also the tours $z_1, z_2, \ldots, z_{m+1}$ from Note 1 and assume that $z_1 = w_1$, since $z_1 = (\bar{x}_1, \bar{y}_1)$ satisfies also $\bar{y}_1(e) = 0$.

Drone 1	Drone 2	Drone 3
z_1 z_2 ... z_{m+1}	z_1 z_1 ... z_1	z_1 z_1 ... z_1
z_1 z_1 ... z_1	z_1 z_2 ... z_{m+1}	z_1 z_1 ... z_1
z_1 z_1 ... z_1	z_1 z_1 ... z_1	z_1 z_2 ... z_{m+1}
$t_{(e_1)}$ ... $t_{(e_{\|E_R\|})}$	z_1 ... z_1	z_1 ... z_1
z_1 ... z_1	$t_{(e_1)}$ ... $t_{(e_{\|E_R\|})}$	z_1 ... z_1
z_1 ... z_1	z_1 ... z_1	$t_{(e_1)}$ ... $t_{(e_{\|E_R\|})}$

Figure 3: K-RPP solutions to prove dimension

Drone 1	Drone 2	Drone 3
$z_2 - z_1$ ... $z_{m+1} - z_1$	0	0
0	$z_2 - z_1$ ... $z_{m+1} - z_1$	0
0	0	$z_2 - z_1$ ... $z_{m+1} - z_1$
-1 -1 $\ddots$ -1 -1	0	0
0	-1 -1 $\ddots$ -1 -1	0
0	0	-1 -1 $\ddots$ -1 -1

Figure 4: K-RPP solutions to prove dimension

We can build K-RPP solutions in the following way. Drone k performs any 1-RPP tour w_j above, while the other drones perform z_1. These m solutions are depicted as the rows of the first block row in the matrix shown in Figure 5, where we assume that drone k is the first one. Now, a drone different from k performs any tour z_j above, while the other drones do z_1. These $(m+1)(K-1)$ solutions are depicted as the rows of the second and third block rows

in the matrix shown in Figure 5.

Drone 1	Drone 2	Drone 3
$z_1 = w_1$	z_1	z_1
w_2	z_1	z_1
...	...	...
w_m	z_1	z_1
z_1	z_1	z_1
z_1	z_2	z_1
...	...	...
z_1	z_{m+1}	z_1
z_1	z_1	z_1
z_1	z_1	z_2
...	...	...
z_1	z_1	z_{m+1}
$t_{(e_1)}$	z_1	z_1
...	...	...
$t_{(e_{\|E_R\|})}$	z_1	z_1
z_1	$t_{(e_1)}$	z_1
...	...	...
z_1	$t_{(e_{\|E_R\|})}$	z_1
z_1	z_1	$t_{(e_1)}$
...	...	...
z_1	z_1	$t_{(e_{\|E_R\|})}$

Figure 5: K-RPP solutions to prove Theorem 11 (case $e \in E'_{NR}$)

Furthermore, we can build $K|E_R|$ more K-RPP solutions as in the proof of Theorem 9 with the same vectors $t_{(e_j)}$ for each required edge in $E_R = \{e_1, e_2, \ldots, e_{|E_R|}\}$. Note that the corresponding vectors $t_{(e_j)}$ also satisfy $y_{(e_j)}(e) = 0$. These solutions are depicted in the three last block rows in the matrix in Figure 5. As in Theorem 9, if we subtract the first row from all the other rows and then we remove the null rows, we obtain a matrix similar to that in Figure 4 but with $K(2|E_{NR}| + |E_R|) - 1$ rows LI. Hence, we have $K(2|E_{NR}| + |E_R|)$ solutions affinely independent satisfying $y_e^k = 0$ and we are done.

Let us suppose now that $e \in E''_{NR}$. Given that $y_e \geq 0$ is facet-inducing of 1-RPP(G) (Theorem 4), there exist $m = 2|E_{NR}|$ affinely independent 1-RPP tours $w_1, w_2, \ldots, w_m$ on G, each $w_i = (x_i, y_i)$, satisfying $x_i(a) = 1$ for all $a \in E_R$, and $y_i(e) = 0$. Since dim(1-RPP(G))$= 2|E_{NR}| = m$, there exist $m + 1$ 1-RPP tours $z_1, z_2, \ldots, z_{m+1}$, each $z_i = (\bar{x}_i, \bar{y}_i)$ satisfying $\bar{x}_i(e) = 1$ for all $e \in E_R$. We can assume here that one of them, say z_1, satisfies $\bar{x}_1(a) = 1$ for all $a \in E_R$, $\bar{x}_1(a) = 1$, $\bar{y}_1(a) = 0$ for all $a \in E'_{NR}$, $\bar{x}_1(a) = \bar{y}_1(a) = 1$ for all $a \in E''_{NR}$, $a \neq e$, and $\bar{x}_1(e) = \bar{y}_1(e) = 0$. Thereby, we can assume that $z_1 = w_1$. Furthermore, for each required edge $a \in E_R$, the corresponding vector $t_{(a)}$ is equal to z_1 except for the entries $x_{(a)}(a) = x_{(a)}(a') = 0$, and also satisfies $y_{(a)}(e) = 0$. Hence, we can build the K-RPP solutions as in the matrix in Figure 5 and the remainder of the proof is similar to the previous case. ♦

Theorem 12 *Inequality $x_e^k \leq 1$, for each edge $e \in E_{NR}$, and for each drone $k \in \{1, 2, \ldots, K\}$, is facet-inducing of K-RPP(G).*

Proof: The proof is similar to that of Theorem 11 and is omitted here for the sake of brevity. ♦

Theorem 13 *Inequality* (5) $x_e^k \geq y_e^k$, *for each edge* $e \in E_{NR}$, *and for each drone* k, *is facet-inducing of K-RPP(G) if graph* $(V, E_{NR} \setminus \{e\})$ *is 3-edge connected.*

Proof: The proof is omitted for the sake of brevity. ♦

Theorem 14 *Inequality* $x_e^k \leq 1$, *for each edge* $e \in E_R$, *and for each drone* k, *is facet-inducing of K-RPP(G).*

Proof: All the K-RPP solutions shown in Figure 3, except the first one of the fourth row block (the one with drone k performing $t_{(e_1)}$), are affinely independent K-RPP solutions satisfying $x_e^k = 1$. ♦

Theorem 15 *Inequality* $x_e^k \geq 0$, *for each edge* $e \in E_R$, *and for each drone* k, *is facet-inducing of K-RPP(G).*

Proof: The proof is omitted for the sake of brevity. ♦

Theorem 16 *Inequalities* (4), $\sum_{k=1}^{K} x_e^k \geq 1$, *for each edge* $e \in E_R$, *are facet-inducing of K-RPP(G).*

Proof: Consider the 1-RPP tours $z_1, z_2, \ldots, z_{m+1}$ and $t_{(a)}$, $a \in E_R$, from Note 1. Let w_1 the tour on G obtained by replacing in z_1 the traversal of edge e by a second traversal of the corresponding parallel edge e'.

We can build $(m+1)K$ K-RPP solutions in the following way. A drone performs any tour z_j above, while the other drones perform w_1. These $(m+1)K$ solutions are depicted as the rows of the three block rows in the matrix shown in Figure 6a. Now, for any required edge $a \neq e$, a drone performs $t_{(a)}$ while the other drones perform w_1 (see Figure 6a, where the required edges different from e are $\{e_2, \ldots, e_{|E_R|}\}$). All the rows in the matrix depicted in Figure 6a represent K-RPP solutions satisfying $\sum_{k=1}^{K} x_e^k = 1$.

If we subtract the first row from all the other rows we obtain the matrix in Figure 6b. This matrix is shown in more detail in Figure 6c, where the three leftmost entries in each block k correspond to the variables $x_e^k, x_{e'}^k, y_{e'}^k$, and vectors v_i represent the remaining vectors $z_i - z_1$ (and of vectors $z_i - w_1$), $i = 2, \ldots, m+1$. Block (1,1) has full rank. If, in blocks (2,2) and (3,3) we subtract the first row from the remaining rows, we would obtain two blocks where rows 2 to $m+1$ are identical to those in block (1,1). Hence, the rows in the first three block rows are linearly independent. Furthermore, regarding the last three block rows of the matrix, the value -1 corresponding to each required edge $a \neq e$ is the only non-zero value in its column. Hence, any of these rows can be obtained as a linear combination of the other rows. Therefore the matrix in Figure 6c has all its $m+(K-1)(m+1)+K(|E_R|-1) = K(2|E_{NR}|+|E_R|)-1$ rows linearly independent. Hence, we have $K(2|E_{NR}|+|E_R|)$ K-RPP solutions affinely independent satisfying $\sum_{k=1}^{K} x_e^k = 1$ and the proof is complete. ♦

Theorem 17 *Connectivity inequalities* (2), $x^k\big(\delta_R(S)\big) + (x^k + y^k)\big(\delta_{NR}(S)\big) \geq 2x_f^k$, $\forall S \subseteq V \setminus \{1\}$, $\forall f \in E(S)$, *and* $\forall k = 1, \ldots, K$, *are facet-inducing of K-RPP(G) if subgraph* $(S, E_{NR}(S))$ *is 3-edge connected and either* $|V \setminus S| = 1$ *or subgraph* $(V \setminus S, E_{NR}(V \setminus S))$ *is 3-edge connected.*

Drone 1	Drone 2	Drone 3
z_1 z_2 ... z_{m+1}	w_1 w_1 ... w_1	w_1 w_1 ... w_1
w_1 w_1 ... w_1	z_1 z_2 ... z_{m+1}	w_1 w_1 ... w_1
w_1 w_1 ... w_1	w_1 w_1 ... w_1	z_1 z_2 ... z_{m+1}
$t_{(e_2)}$ ... $t_{(e_{\|E_R\|})}$	w_1 ... w_1	w_1 ... w_1
w_1 ... w_1	$t_{(e_2)}$ ... $t_{(e_{\|E_R\|})}$	w_1 ... w_1
w_1 ... w_1	w_1 ... w_1	$t_{(e_2)}$ ... $t_{(e_{\|E_R\|})}$

(a)

Drone 1	Drone 2	Drone 3
$z_2 - z_1$ ... $z_{m+1} - z_1$	0	0
$w_1 - z_1$ $w_1 - z_1$ ... $w_1 - z_1$	$z_1 - w_1$ $z_2 - w_1$ ... $z_{m+1} - w_1$	0
$w_1 - z_1$ $w_1 - z_1$ ... $w_1 - z_1$	0	$z_1 - w_1$ $z_2 - w_1$ ... $z_{m+1} - w_1$
$t_{(e_2)} - z_1$ ... $t_{(e_{\|E_R\|})} - z_1$	0	0
$w_1 - z_1$ ... $w_1 - z_1$	$t_{(e_2)} - w_1$ ... $t_{(e_{\|E_R\|})} - w_1$	0
$w_1 - z_1$ ... $w_1 - z_1$	0	$t_{(e_2)} - w_1$ ... $t_{(e_{\|E_R\|})} - w_1$

(b)

Drone 1	Drone 2	Drone 3
0 α β v_2 ... ... 0 λ μ v_{m+1}	0	0
-1 0 1 -1 0 1 ... -1 0 1	0	1 0 -1 0 1 α (β-1) v_2 ... ... 1 λ (μ-1) v_{m+1}
-1 0 1 -1 0 1 ... -1 0 1	0	0	1 0 -1 0 1 α (β-1) v_2 ... ... 1 λ (μ-1) v_{m+1}
0 * * -1 -1 ... $\ddots$ 0 * * -1 -1	0	0
-1 0 1 ... -1 0 1	0	1 * * -1 -1 ... $\ddots$ 1 * * -1 -1	0
-1 0 1 ... -1 0 1	0	0	1 * * -1 -1 ... $\ddots$ 1 * * -1 -1

(c)

Figure 6: K-RPP solutions to prove Theorem 16

Proof: In some points of this proof we will distinguish the cases $f \in E_{NR}$ and $f \in E_R$. Let $z_1, z_2, \ldots, z_{m+1}$ and $t_{(a)}$, $a \in E_R$, be the 1-RPP tours from Note 1. Consider the graph G^* obtained by deleting from G the required edges in $\delta_R(S) \cup E_R(S)$. Graphs G and G^* have the same set of non-required edges and, given that the hypotheses of Theorem 5 are fulfilled, there are m affinely and linearly independent 1-RPP tours $z_1^*, z_2^*, \ldots, z_m^*$ on G^*, all of them traversing the required edges $a \in E_R(V \setminus S)$ and satisfying $(x + y)(\delta(S)) = 2x_{\bar{f}}$, where, if edge $f \in E_R$, $\bar{f} = f'$ and $\bar{f} = f$ otherwise. Each z_j^* is transformed into a tour on G, w_j, by adding to it an extra entry x_e with value $x_e = 0$ for each removed edge $e \in \delta_R(S) \cup E_R(S)$. In the case $f \in E_R$, we replace a traversal of f' (if any) by the traversal of f. It can be seen that the resulting tours $w_1, w_2, \ldots, w_m$ on G are also affinely independent, satisfy $x(\delta_R(S)) + (x + y)(\delta_{NR}(S)) = 2x_f$, and do not traverse any edge in $\delta_R(S) \cup E_R(S)$ (except f if it is required). We can assume that one of them, say w_1, is equal to z_1 in the entries corresponding to $E(V \setminus S)$, it traverses the cutset $\delta(S)$ twice through a non-required edge, say $\bar{e}$, and $x_1(a) = y_1(a) = 1$ for all $a \in E_{NR}(S)$, $x_1(a) = 0$ for all $a \in E_R(S)$, in the case $f \in E_{NR}$, and $x_1(f) = x_1(f') = 1$, $y_1(f') = 0$, and $x_1(a) = 0$ for all $a \in E_R(S) \setminus \{f\}$, in the case $f \in E_R$.

As in previous theorems, we build the K-RPP solutions depicted in the first three block rows in the matrix shown in Figure 7. We build also the solutions depicted in the last two block rows in the matrix shown in Figure 7, where a drone different from k performs any $t_{(a)}$, while drone k performs w_1 and the remaining drones perform z_1.

For each required edge $a \in E_R(V \setminus S)$, we define the vector $t^*_{(a)} = (x^*_{(a)}, y^*_{(a)})$ equal to w_1 except for the entries $x^*_{(a)}(a) = x^*_{(a)}(a') = 0$. For each required edge $a \in \delta_R(S)$, we define the vector $t^*_{(a)} = (x^*_{(a)}, y^*_{(a)})$ equal to w_1 except for the entries $x^*_{(a)}(\bar{e}) = y^*_{(a)}(\bar{e}) = 0$ and $x^*_{(a)}(a) = x^*_{(a)}(a') = 1$. Note that if $\bar{e} = a'$ then $t^*_{(a)}$ is equal to w_1 except for the entries $y^*_{(a)}(a') = 0$ and $x^*_{(a)}(a) = 1$.

In the case $f \in E_{NR}$, for each required edge $a \in E_R(S)$, we define the vector $t^*_{(a)} = (x^*_{(a)}, y^*_{(a)})$ equal to w_1 except for the entries $x^*_{(a)}(a) = 1$ and $y^*_{(a)}(a') = 0$ (we replace the second traversal of a' by the traversal of a). In the case $f \in E_R$, for each $a \in E_R(S) \setminus \{f\}$, we define the vectors $t^*_{(a)} = (x^*_{(a)}, y^*_{(a)})$ as before. Moreover, we define a new vector $t^*_{(f)}$ equal to w_1 in $E(V \setminus S)$ and zero in all the remaining entries. Note that this vector also satisfies $x(\delta_R(S)) + (x + y)(\delta_{NR}(S)) = 2x_f = 0$. If drone k performs $t^*_{(a)}$ and the remaining drones perform z_1 we obtain the K-RPP solutions depicted as the rows of the fourth block rows in the matrix shown in Figure 7.

If $K \geq 3$ (the case $K = 2$ is argued at the end of this proof), all the rows in the matrix depicted in Figure 7 represent K-RPP solutions satisfying the connectivity inequality (2) as an equality. If we subtract the first row from all the other rows and then remove the zero rows we obtain the matrix in Figure 8. It can be seen that the rows of block B^*, associated with vectors $t^*_{(a)} - w_1$, have the three entries corresponding to each $a \in E_R$ and its corresponding parallel edge a' as follows:

- $x(a) = x(a') = -1$, $y(a') = 0$, for each $a \in E_R(V \setminus S)$,
- $x(a) = x(a') = 1$, $y(a') = 0$, for each $a \in \delta_R(S)$, and
- $x(a) = 1$, $x(a') = 0$, $y(a') = -1$, for each $a \in E_R(S)$ in the case $f \in E_{NR}$, while,
- in the case $f \in E_R$, $x(a) = 1$, $x(a') = 0$, $y(a') = -1$, for each $a \in E_R(S) \setminus \{f\}$, and $x(f) = -1$, $x(f') = -1$, $y(f') = 0$.

Drone 1	Drone 2	Drone 3
w_1	z_1	z_1
w_2	z_1	z_1
...	...	...
w_m	z_1	z_1
w_1	z_1	z_1
w_1	z_2	z_1
...	...	...
w_1	z_{m+1}	z_1
w_1	z_1	z_1
w_1	z_1	z_2
...	...	...
w_1	z_1	z_{m+1}
$t^*_{(a_1)}$	z_1	z_1
...	...	...
$t^*_{(a_{\lvert E_R\rvert})}$	z_1	z_1
w_1	$t_{(a_1)}$	z_1
...	...	...
w_1	$t_{(a_{\lvert E_R\rvert})}$	z_1
w_1	z_1	$t_{(a_1)}$
...	...	...
w_1	z_1	$t_{(a_{\lvert E_R\rvert})}$

Figure 7: K-RPP solutions to prove Theorem 17

In the case $f \in E_{NR}$, the matrix obtained after merging blocks (1,1) and B^* of Figure 8 is detailed in Figure 9, with the columns and rows rearranged. Note that, for $a \in E_R$ the corresponding values -1 or 1 are the only nonzero entries in its corresponding column of the matrix in Figure 8. In the case $f \in E_R$, the entry corresponding to edge f in the block (4,4) of this matrix, is -1 instead of 1. In both cases, this matrix has full rank. Hence, the matrix of Figure 8 has full rank and its $m-1+(K-1)m+K|E_R| = K(2|E_{NR}|+|E_R|)-1$ rows are linearly independent. Hence, we have $K(2|E_{NR}|+|E_R|)$ K-RPP solutions affinely independent satisfying inequality (2) as an equality and we are done.

If $K = 2$, the solutions corresponding to some rows of the last block row of the corresponding matrix, $[w_1, t_{(a_i)}]$, are not actually K-RPP solutions. Note that, if $a_i \in \delta_R(S) \cup E_R(S)$, neither drone 1, performing w_1, nor drone 2, performing $t_{(a_i)}$, traverses this edge. In this case, for each $a_i \in \delta_R(S) \cup E_R(S)$ we consider the solution in which drone 1 performs $t^*_{(a_i)}$ (similar to w_1 but traversing a_i) and drone 2 performs $t_{(a_i)}$, which is a K-RPP solution. It can be seen that the corresponding matrix is also full rank. ♦

Theorem 18 *Parity inequalities (23) are facet-inducing of K-RPP(G) if subgraph $(S, E_{NR}(S))$ is 3-edge connected and either $V \setminus S = \{1\}$, or subgraph $(V \setminus S, E_{NR}(V \setminus S))$ is 3-edge connected.*

Proof: Let $z_1, z_2, \ldots, z_{m+1}$ and $t_{(a)}$, $a \in E_R$, the 1-RPP tours from Note 1. Consider the graph G^* obtained by deleting from G the required edges in $\delta_R(S) \setminus F$ (if any). Graphs G and G^* have the same set of non-required edges and, given that the hypotheses of Theorem 6 are

Drone 1	Drone 2	Drone 3
$w_2 - w_1$ ... $w_m - w_1$	0	0
0	$z_2 - z_1$ ... $z_{m+1} - z_1$	0
0	0	$z_2 - z_1$ ... $z_{m+1} - z_1$
B^*	0	0
0	-1 -1 $\ddots$ -1 -1	0
0	0	-1 -1 $\ddots$ -1 -1

Figure 8: K-RPP solutions to prove Theorem 17

E_{NR}	$E_R(V \setminus S)$	$\delta_R(S)$	$E_R(S)$
$(w_2 - w_1)_{NR}$ ... $(w_m - w_1)_{NR}$	0	0	0
$*$	-1 $\ddots$ -1	0	0
$*$	0	1 $\ddots$ 1	0
$*$	0	0	1 $\ddots$ 1

Figure 9: A submatrix of the matrix in Figure 8

fulfilled, there are m affinely and linearly independent 1-RPP tours $w_1^*, w_2^*, \ldots, w_m^*$ on G^*, all of them traversing the required edges in G^* (all the edges in E_R except the removed ones, in $\delta_R(S) \setminus F$, if any), and satisfying inequalities (19) as equalities. These tours also satisfy $x_e = 1$ for all $e \in F_R$ and, by adding $x(F_R) - |F_R| = 0$ to the right hand side of (19) we obtain that the tours w_j^* satisfy

$$(x - y)(\delta_{NR}(S) \setminus F) = x(F_R) + (x - y)(F_{NR}) - |F| + 1.$$

Each w_j^* is transformed into a tour on G, w_j, by adding to it an extra entry x_e for each

removed edge $\delta_R(S)\backslash F$ with value $x_e = 0$. The resulting tours $w_1, w_2, \ldots, w_m$ on G are also affinely independent, traverse all the required edges except those in $\delta_R(S)\backslash F$, and satisfy

$$x(\delta_R(S)\backslash F) + (x-y)(\delta_{NR}(S)\backslash F) = x(F_R) + (x-y)(F_{NR}) - |F| + 1.$$

Regarding the traversal of the cut-set $\delta(S)$, there are only two types of 1-RPP tours w_i that satisfy the inequality as an equality (see [12]). Tours of type 1 traverse one copy of each edge in F and one more edge in $\delta(S)\backslash F$ (if $\delta(S)\backslash F \neq \emptyset$). Tours of type 2 traverse one copy of each edge in F, except one edge in F_{NR} (if $F_{NR} \neq \emptyset$). We can assume, for example, that $F_{NR} \neq \emptyset$ and that one of the w_i above, say $w_1 = (x_1, y_1)$, traverses once each edge in F except a given edge $e \in F_{NR}$ (in the case $F_{NR} = \emptyset$, and hence, $\delta(S)\backslash F \neq \emptyset$, we would proceed in a similar way).

We build the K-RPP solutions depicted in the first three block rows in the matrix shown in Figure 7, where vectors z_i are the same but vectors w_i are different but denoted equal. We build also the solutions depicted in the last two block rows in the matrix in Figure 7.

For each required edge a we define a vector $t^*_{(a)}$, obtained from w_1, that also satisfies the inequality as an equality (it is of one of the types 1 or 2 above) and, regarding the required edges, only differs from w_1 in the traversals of edge a in the following way (if any edge in E_{NR} turns to be traversed three times, we would delete two traversals of it):

- For each $a \in E_R(S)$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the traversal of a with the traversal of a path joining its two endpoints formed with edges in $E_{NR}(S)$. For each $a \in E_R(V \setminus S)$ we proceed in the same way.
- For each $a \in F_R$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the traversal of a with the traversal of edge e and two paths, formed with edges in $E_{NR}(S)$ and $E_{NR}(V \setminus S)$, joining the endpoints of a and e.
- For each $a \in \delta_R(S) \setminus F$, the vector $t^*_{(a)}$ is obtained from w_1 by adding the traversal of a cycle formed with edges a and e and two paths with edges in $E_{NR}(S)$ and in $E_{NR}(V \setminus S)$.

If drone k performs $t^*_{(a)}$ and the remaining drones perform z_1 we obtain the K-RPP solutions depicted as the rows of the fourth block rows in the matrix shown in Figure 7, although note that vectors $t^*_{(a)}$ are different but share the same name.

If $K \geq 3$ (for the case $K = 2$, the argument is similar to that in the proof of Theorem 17), all the rows in the matrix depicted in Figure 7 represent K-RPP solutions satisfying the parity inequality (23) as an equality. If we subtract the first row from all the other rows and then remove the zero rows we obtain the matrix in Figure 8.

The matrix obtained after merging blocks (1,1) and B^* of Figure 8 is detailed in Figure 10, with the columns and rows rearranged. This matrix has full rank. Hence, the matrix of Figure 8 has full rank and we have $K(2|E_{NR}| + |E_R|)$ K-RPP solutions affinely independent satisfying inequality (23) as an equality. ♦

Theorem 19 *p-connectivity inequalities (24) are facet-inducing for K-RPP(G) if subgraphs $(S_i, E_{NR}(S_i))$, $i = 0, \ldots, p$, are 3-edge connected, $|(S_0 : S_i)| \geq 2$, $\forall\ i = 1, \ldots, p$, and the graph induced by $V \setminus S_0$ is connected, and all the edges $e_j \in E(S_j)$ are required edges.*

E_{NR}	$E_R \setminus (\delta_R(S) \setminus F_R)$		$\delta_R(S) \setminus F_R$
$(w_2 - w_1)_{NR}$ ... $(w_m - w_1)_{NR}$	0	0	0
$*$	-1 $\ddots$ -1	0	0
$*$	0	-1 $\ddots$ -1	0
$*$	0	0	1 $\ddots$ 1

Figure 10: A matrix appearing in the proof of Theorem 17

Proof: Let $z_1, z_2, \ldots, z_{m+1}$ and $t_{(a)}$, $a \in E_R$, the 1-RPP tours from Note 1. Consider the graph G^* obtained by deleting from G the required edges in $\delta_R(S_i)$ for all i (if any). Graphs G and G^* have the same set of non-required edges and, given that the hypothesis of Theorem 7 are fulfilled, there are m affinely independent 1-RPP tours $w_1^*, w_2^*, \ldots, w_m^*$ on G^*, all of them traversing the required edges in G^* and satisfying

$$(x + y)(\delta(S_0)) + 2 \sum_{1 \le r < t \le p} x(S_r : S_t) = 2p.$$

Given that all the w_i^* traverse each edge $e_j \in E(S_j)$, they also satisfy

$$(x + y)(\delta(S_0)) + 2 \sum_{1 \le r < t \le p} x(S_r : S_t) = 2 \sum_{i=0, i \ne d}^{p} x_{e_i}$$

Each w_j^* is transformed into a tour on G, w_j, by adding to it an extra entry x_e for each removed edge in $\delta_R(S_i)$ for all i, with value $x_e = 0$. The resulting tours $w_1, w_2, \ldots, w_m$ on G are also affinely independent, traverse all the required edges except those in sets $\delta_R(S_i)$, and satisfy

$$x(\delta_R(S_0)) + (x + y)(\delta_{NR}(S_0)) + 2 \sum_{1 \le r < t \le p} x(S_r : S_t) = 2 \sum_{i=0, i \ne d}^{p} x_{e_i} \quad \big(= 2p\big).$$

Furthermore, we can assume that one of the w_i above, say $w_1 = (x_1, y_1)$, traverses twice a given non-required edge in each set $(S_0 : S_i)$.

We build the K-RPP solutions depicted in the first three and last two block rows in the matrix shown in Figure 7, where vectors z_i, $t_{(a)}$ are the same but vectors w_i are different but denoted equal.

For each required edge a we define a vector $t^*_{(a)}$ obtained from w_1 that also satisfies the inequality as an equality as follows:

- For each $a \in E_R(S_i)$, $a \neq e_i$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the traversal of a with the traversal of a path joining its two endpoints formed with edges in $E_{NR}(S_i)$.
- For each edge $e_i \in E_R(S_i)$, the vector $t^*_{(e_i)}$ is obtained from w_1 by removing all the traversals in $(S_0 : S_i)$ and in $E(S_i)$. Note that this vector satisfies

$$x(\delta_R(S_0)) + (x+y)(\delta_{NR}(S_0)) + 2\sum_{1\le r<t\le p} x(S_r : S_t) = 2\sum_{i=0, i\neq d}^{p} x_{e_i} \quad \big(= 2(p-1)\big).$$

- For each $a \in (S_0 : S_i)_R$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the second traversal of the non-required edge in $(S_0 : S_i)$ traversed twice by w_1 with the traversal of a and a path joining its two endpoints formed with edges in $E_{NR}(S_0) \cup E_{NR}(S_i)$.
- For each $a \in (S_i : S_j)_R$, the vector $t^*_{(a)}$ is obtained from w_1 by adding one copy of a and removing a copy of each non-required edge in $(S_0 : S_i)$ and $(S_0 : S_j)$ traversed twice by w_1 and the traversal of the non-required edges in the paths joining their endpoints.

If drone k performs $t^*_{(a)}$ and the remaining drones perform z_1 we obtain the K-RPP solutions depicted as the rows of the fourth block rows in the matrix shown in Figure 7.

If $K \geq 3$ (for the case $K = 2$, the argument is similar to that in the proof of Theorem 17), all the rows in the matrix depicted in Figure 7 represent K-RPP solutions satisfying the p-connectivity inequality (24) as an equality. If we subtract the first row from all the other rows and then remove the zero rows we obtain a matrix similar to that in Figure 8. The matrix obtained after merging blocks (1,1) and B^* of Figure 8 is detailed in Figure 11, with the columns and rows rearranged. This matrix has full rank and, therefore, also the matrix of Figure 8 has full rank. Hence, we have $K(2|E_{NR}| + |E_R|)$ K-RPP solutions affinely independent satisfying inequality (24) as an equality and it defines a facet of K-RPP(G). ♦

Theorem 20 *K-C inequalities (25) and (26) are facet-inducing for K-RPP(G) if subgraphs $(S_i, E_{NR}(S_i))$, $i = 0, \ldots, K$, are 3-edge connected, $|(S_i : S_{i+1})| \geq 2$ for $i = 0, \ldots, K-1$, and $|F_R| \geq 2$ and $e_i \in E_R(S_i)$ for all i.*

Proof: We do the proof only for inequalities (25) since the proof for inequalities (26) is analogous. Let $z_1, z_2, \ldots, z_{m+1}$ and $t_{(a)}$, $a \in E_R$, be the 1-RPP tours from Note 1. Consider the graph G^* obtained by deleting from G the required edges in $\delta_R(S_i)$ for all $i = 1, \ldots, K-1$ and the required edges in $(S_0 : S_K) \setminus F$. Graphs G and G^* have the same set of non-required edges and, given that the hypothesis of Theorem 8 are fulfilled, there are m affinely independent 1-RPP tours $w^*_1, w^*_2, \ldots, w^*_m$ on G^*, all of them traversing the required edges in G^*, and satisfying inequality (22) with $|\mathcal{R}| = K - 1$ as an equality:

$$(K-2)(x-y)\Big((S_0 : S_K) \setminus F\Big) - (K-2)(x-y)(F_{NR}) +$$
$$+ \sum_{\substack{0\le i<j\le K \\ (i,j)\neq(0,K)}} \Big((j-i)\, x(S_i : S_j) + (2-j+i)\, y(S_i : S_j)\Big) = 2(K-1) - (K-2)|F_{NR}|.$$

E_{NR}	$(S_i : S_j)_R$	$E_R(S_0)$	$E_R(S_1)$	$\cdots$	$E_R(S_p)$
$(w_2 - w_1)_{NR}$ ... $(w_m - w_1)_{NR}$	0	0	0	$\cdots$	0
$*$	1 $\ddots$ 1	0	0	$\cdots$	0
$*$	0	-1 $\ddots$ -1	0	$\cdots$	0
$*$	0	0	-1 $\ddots$ -1 -1 $\cdots$ -1 -1	$\cdots$	0
$*$	0	0	0	$\ddots$	0
$*$	0	0	0	$\cdots$	-1 $\ddots$ -1 -1 $\cdots$ -1 -1

Figure 11: A submatrix of the matrix in Figure 8 for p-connectivity inequalities

Given that all the w_j^* traverse each edge $e_i \in E(S_i)$, they also satisfy

$$(K-2)(x-y)\Big((S_0 : S_K) \setminus F\Big) - (K-2)(x-y)(F_{NR}) +$$
$$+ \sum_{\substack{0\le i<j\le K \\ (i,j)\neq(0,K)}} \Big((j-i)x(S_i : S_j) + (2-j+i)y(S_i : S_j)\Big) = 2\sum_{i=1}^{K-1} x_{e_i} - (K-2)|F_{NR}|.$$

Each w_j^* is transformed into a tour on G, w_j, by adding to it an extra entry x_e for each removed edge in $\delta_R(S_i)$, for all $i = 1, \ldots, K-1$, and in $(S_0 : S_K)_R \setminus F$, with value $x_e = 0$. The resulting tours $w_1, w_2, \ldots, w_m$ on G are also affinely independent, traverse all the required edges except those in sets $\delta_R(S_i)$ and in $(S_0 : S_K)_R \setminus F$, and satisfy

$$(K-2)\,x\Big((S_0 : S_K)_R \setminus F\Big) - (K-2)\,x(F_R) +$$
$$+ (K-2)\,(x-y)\Big((S_0 : S_K)_{NR} \setminus F\Big) - (K-2)\,(x-y)(F_{NR}) +$$
$$+ \sum_{\substack{0\le i<j\le K \\ (i,j)\neq(0,K)}} (j-i)\,x(S_i : S_j)_R + \sum_{\substack{0\le i<j\le K \\ (i,j)\neq(0,K)}} \Big((j-i)\,x(S_i : S_j)_{NR} + (2-j+i)\,y(S_i : S_j)_{NR}\Big) =$$
$$= 2\sum_{i=1}^{K-1} x_{e_i} - (K-2)|F|, \tag{30}$$

because each w_j traverses each edge in F_R and, hence, $x(F_R) = |F_R|$. Furthermore, we can assume that one of the w_j above, say $w_1 = (x_1, y_1)$, traverses once each edge in F, and traverses twice a given non-required edge in each set $(S_j : S_{j+1})$ for $j = 0, \ldots, K-2$.

We build the K-RPP solutions depicted in the first three and last two block rows in the matrix shown in Figure 7, where vectors z_i, $t_{(a)}$ are the same but vectors w_i are different but denoted equal.

For each edge $a \in E_R$ we define a vector $t^*_{(a)}$ obtained from w_1 and also satisfying equation (30) as follows:

- For each $a \in E_R(S_i)$, $a \neq e_i$, $i = 0, 1, \ldots, K$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the traversal of a with the traversal of a path joining its two endpoints formed with edges in $E_{NR}(S_i)$.
- For each $a = e_i \in E_R(S_i)$, $i \neq K-1$, the vector $t^*_{(a)}$ is obtained from w_1 by removing all the traversals in $E(S_i)$, $(S_{i-1} : S_i)$, and $(S_i : S_{i+1})$, and adding two copies of a non-required edge in $(S_{K-1} : S_K)$. For the edge $a = e_{K-1}$, the vector $t^*_{(a)}$ is obtained from w_1 by removing all the traversals in $E(S_{K-1})$ and in $(S_{K-2} : S_{K-1})$.
- For each $a \in (S_j : S_{j+1})_R$, $j = 0, \ldots, K-1$, the vector $t^*_{(a)}$ is obtained from w_1 by replacing the second traversal of the non-required edge traversed twice in $(S_j : S_{j+1})$ with the traversal of a and a path joining its two endpoints formed with edges in $E_{NR}(S_j) \cup E_{NR}(S_{j+1})$.
- For each $a \in (S_i : S_j)_R$, $|i-j| > 1$ the vector $t^*_{(a)}$ is obtained from w_1 by adding one copy of a and removing the second traversal of the non-required edge traversed twice in $(S_i : S_{i+1}), \ldots, (S_{j-1} : S_j)$ and adding the traversal of the non-required edges in some paths in $E(S_i), \ldots, E(S_j)$.
- For each $a \in F_R$, the vector $t^*_{(a)}$ is obtained from w_1 by removing a and all the second traversals of the non-required edges traversed twice in sets $(S_j : S_{j+1})$, $j = 0, \ldots, K-2$, and adding a non-required edge in $(S_{K-1} : S_K)$ and the non-required edges in some paths in $E(S_0), E(S_1) \ldots, E(S_K)$.
- For each $a \in (S_0 : S_K)_R \setminus F$, the vector $t^*_{(a)}$ is obtained from w_1 by adding a, removing all the second traversals of the non-required edges traversed twice in sets $(S_j : S_{j+1})$, $j = 0, \ldots, K-2$, and adding a non-required edge in $(S_{K-1} : S_K)$ and the non-required edges in some paths in $E(S_0), E(S_1) \ldots, E(S_K)$.

If drone k performs $t^*_{(a)}$ and the remaining drones perform z_1 we obtain the K-RPP solutions depicted as the rows of the fourth block rows in the matrix shown in Figure 7.

If $K \geq 3$ (for the case $K = 2$, the argument is similar to that in the proof of Theorem 17), all the rows in the matrix depicted in Figure 7 represent K-RPP solutions satisfying the K-C inequality as an equality. If we subtract the first row from all the other rows and then remove the zero rows we obtain a matrix similar to that in Figure 8. The matrix obtained after merging blocks (1,1) and B^* of Figure 8 is detailed in Figure 12, with the columns and rows rearranged.

For the sake of simplicity, the elements in the block corresponding to edges in $E_R(S_0 \cup S_K)$ are ± 1, representing 1 for the edges in F_R, and -1 for the edges in $E_R(S_0)$, $E_R(S_K)$, and $(S_0 : S_K)_R \setminus F$. This matrix has full rank. Hence, we have $K(2|E_{NR}| + |E_R|)$ K-RPP solutions affinely independent satisfying the K-C inequality (25) as an equality. ♦

E_{NR}	$(S_i{:}S_j)_R$	$E_R(S_0 \cup S_K)$	$E_R(S_1)$	$\cdots$	$E_R(S_{K-1})$
$(w_2 - w_1)_{NR}$ ... $(w_m - w_1)_{NR}$	0	0	0	$\cdots$	0
$*$	$\begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix}$	0	0	$\cdots$	0
$*$	0	$\begin{matrix} \pm 1 & & \\ & \ddots & \\ & & \pm 1 \end{matrix}$	0	$\cdots$	0
$*$	0	0	$\begin{matrix} -1 & & & \\ & \ddots & & \\ & & -1 & \\ -1 & \cdots & -1 & -1 \end{matrix}$	$\cdots$	0
$*$	0	0	0	$\ddots$	0
$*$	0	0	0	$\cdots$	$\begin{matrix} -1 & & & \\ & \ddots & & \\ & & -1 & \\ -1 & \cdots & -1 & -1 \end{matrix}$

Figure 12: A submatrix of the matrix in Figure 8 for K-C inequalities