Computers & Operations Research 159 (2023) 106321

Contents lists available at ScienceDirect

Computers and Operations Research

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

Upgrading edges in the Graphical TSP

Mercedes Landete [a], Isaac Plana [b,*], José Luis Sainz-Pardo [a], José María Sanchis [c]

[a] Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández, Spain
[b] Departamento de Matemáticas para la Economía y la Empresa, Universidad de Valencia, Spain
[c] Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Spain

ARTICLE INFO

Dataset link: www.uv.es/plani/instancias.htm

Keywords:
GTSP
Edge upgrading
Branch-and-cut

ABSTRACT

In this paper we present a generalization of the Graphical Traveling Salesman Problem (GTSP). Given a communication graph in which not all direct connections are necessarily possible, the Graphical Traveling Salesman Problem consists of finding the shortest tour that visits each node at least once. In this work, we assume the availability of a budget that allows to upgrade, i.e. reduce their traversal cost, some of the current connections and we propose the problem of designing the minimum cost tour using this budget. We propose and study a formulation for the problem, verifying that the polyhedron associated with the set of feasible solutions of a relaxed version of the problem is a full-dimensional polytope. We present families of valid inequalities that reinforce the model and pre-processing techniques to reduce the number of variables of the formulation. To solve the problem, we propose a branch-and-cut algorithm that uses the introduced valid inequalities, as well as a heuristic to obtain good upper bounds and a tailor-made branching strategy. Comprehensive computational experiments on a new set of benchmark instances are presented to assess the performance of this exact method.

1. Introduction

Many of the paths in transportation networks are maintained in time as long as the external conditions do not change. New scenarios such as the acquisition of vehicles, the modification of the demand or the creation of new communication channels lead to the design of new routes that can coexist or replace existing ones. Upgrading connections is a hot topic in network problems. The efficient management of telecommunications networks implies costs associated with the nodes of the network and also costs associated with the edges of the network. The concern to improve the efficiency of the network by investing in modifying the costs of the nodes and/or the edges is not new. Numerous works have been published that deal with the influence of the modification of node and/or edge costs in the solutions of different network problems. These modifications may be motivated both by an additional budget allocation and by the need to improve the current network.

The upgrading of edges and nodes has been studied in many routing problems. In 1977, in the paper Fulkerson and Harding (1977), the problem of investing a budget for upgrading connections in several shortest path problems was introduced. Since then, interest in this topic has not ceased. Different challenging network upgrading problems were modeled and analyzed in 1995 (Paik and Sahni, 1995). In Krumke et al. (1997), it was assumed that each edge has an associated function that specifies the cost of shortening it, and the problem of finding a spanning tree of smallest total length is approached by means of a polynomial time approximation algorithm, while Hambrusch and Tu (1997) considers edge weight reduction problems in directed acyclic graphs. Drangmeister et al. (1998) shows that, in general the problem of finding an optimal reduction strategy for modifying the network is NP-hard, even for simple classes of graphs and linear cost functions. In Fredericksont and Solis-Oba (1999) the problem of determining the maximum increase in the weight of the minimum spanning trees of a graph caused by the removal of a given number of edges, or by finite increases in the weights of the edges, is studied. The problem of improving spanning trees when the cost of the edges can be reduced by upgrading its endpoints is presented in Krumke et al. (1999). The interest in analyzing the effect of the upgrading of connections in communications networks has been maintained over the last 20 years. The effect of reducing the flow cost on arcs by upgrading the arcs has also been studied in Demgensky et al. (2002), as well as some series of problems that involve finding the best q arcs in a network to upgrade (Campbell et al., 2006). Upgrading actions to minimize the shortest delay paths between demand pairs of terminals in the network are analyzed in Dilkina et al. (2011). Approximation algorithms for budget constrained network upgradeable problems are introduced

* Corresponding author.
E-mail addresses: landete@umh.es (M. Landete), isaac.plana@uv.es (I. Plana), jose.sainz-pardo@umh.es (J.L. Sainz-Pardo), jmsanchis@mat.upv.es (J.M. Sanchis).

https://doi.org/10.1016/j.cor.2023.106321
Received 2 December 2022; Received in revised form 16 June 2023; Accepted 16 June 2023
Available online 22 June 2023

in Saharoy and Sen (2014). More recently, more trees and path upgrading problems have appeared in the literature (Alvarez-Miranda and Sinnl, 2017, Sepasian and Monabbati, 2017 and Zhang et al., 2021).

Routing and location problems are both common in telecommunications networks, and usually the decision made when dealing with a location problem depends on the decision previously made in a routing problem and vice versa. Although this paper belongs to the routing group, it is interesting to note that upgrading problems are also currently present in the location research community. As an example we can mention some recent upgrading location papers: Sepasian (2018) for the 1-center problem, Afrashteh et al. (2020) for the obnoxious p-median location problem, Baldomero-Naranjo et al. (2022) for the maximal covering location problem, Blanco and Marín (2019) for the hub location problem and Espejo and Marín (2020) for the network p-median problem.

In this work, we address the design of routes when the traversal cost of part of the network can be changed by paying a price. We assume that we have a budget to improve/upgrade the current connections and we want to decide how to invest that budget to upgrade the connections in order to visit all the nodes of the network with total minimum cost. We consider that for each edge different investments allow different levels of improvements. The problem is to decide which edges of the network to upgrade without exceeding the budget in order to obtain the shortest tour in the upgraded network. We assume that the communications network does not need to form a complete graph. Thus, we introduce an extension of Graphical Traveling Salesman Problem (GTSP) in which each edge cost can be reduced by investing. The GTSP consists of finding the shortest tour on a graph visiting, at least once, each vertex. The GTSP can be solved on a complete graph obtained by adding edges representing shortest paths and then solving a Traveling Salesman Problem (TSP). This is not possible in our case, since we do not know the final traversal cost of the edges in advance and therefore cannot compute such shortest paths. The GTSP, which may be considered a relaxation of the TSP, was introduced in Fleischmann (1985) and Cornuéjols et al. (1985), and further studied in Naddef and Rinaldi (1991), Naddef and Rinaldi (1992), Naddef and Rinaldi (2007), Oswald et al. (2007), and it still deserves researchers attention (Carr et al., 2023). The research gap covered by the content of this paper is the mathematical optimization of the edge upgrading in the GTSP.

Applications for the problem of upgrading edges in the GTSP are the same as applications of the GTSP, which have been widely discussed in the literature . Improving a connection in a communication network can translate into improving the paving, or paying a toll or even hiring more vehicles for the connection in question. The improvement can be measured in units of cost or in units of time. For instance, a road damaged after an earthquake could be improved by simply removing the debris, by repairing the possible holes and cracks that may have appeared, or by constructing a whole new road. Also, the movement time of maintenance operators on a ski slope can be improved by replacing belts with hangers. The travel time of a truck traveling through a conventional road can be improved by using a toll motorway or, in some cases, by riding the truck on a ferry. Depending on what action we decide to take, a different amount of money will have to be spent and also a different improvement in the traversal time of the road can be achieved.

The remainder of the paper is organized as follows. In Section 2, we present the notation and the new model. In Section 3, we prove necessary and sufficient conditions for the polytope of solutions of a relaxation of the problem to be a full-dimensional polytope and also that part of the new model constraints are facet-inducing of this polytope. We also introduce some families of inequalities that have been adapted from other existing problems: parity inequalities, p-connectivity inequalities, and cover inequalities. In Section 4, we introduce more families of valid inequalities for the model which are based on the traversing costs and upgrading prices, while in Section 5 it is proved that our problem can be expressed as a cost-constrained GTSP, which allows the design of a pre-processing algorithm. Section 6 describes the branch-and-cut algorithm, while results for the computational experiments are summarized in Section 7 and our concluding remarks are presented in Section 8.

2. The problem

In the problem we address in this work, we consider a connected undirected graph $G = (V, E)$ with $|V|$ vertices and $|E|$ edges representing a network. Each edge $e \in E$ has K associated non-negative costs, $c_e^1 \geq c_e^2 \geq \cdots \geq c_e^K \geq 0$, corresponding to its traversal at each one of the K upgrading levels. Furthermore, each edge $e \in E$ has K associated values, $0 \leq \alpha_e^1 \leq \alpha_e^2 \leq \cdots \leq \alpha_e^K$, representing the price that must be paid to upgrade the edge to each level (surely $\alpha_e^1 = 0$, since 1 is the current level, α_e^2 is the price to upgrade the edge e up to level 2, and so on). T is the total budget available to improve the edges. In the case of considering the possibility of investing in the construction of a non-existing connection e, the notation would be maintained and the ∞ value would be assigned to c_e^1. In this case, the price of this investment would probably be significantly higher than the rest of the prices.

In the *Upgrading Graphical Traveling Salesman Problem, U-GTSP*, the aim is to decide to what level each edge should be upgraded by spending a total amount not greater than T, so that the cost of the optimal GTSP tour in the upgraded graph is minimal. We will call *U-GTSP tour* to a GTSP tour in G and a selection of a level upgrade for each edge traversed. It is easy to see that any optimal U-GTSP tour traverses at most twice any edge.

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'. Let us denote $\delta(S) = (S : V\backslash S)$ the edge set with one endpoint in S and the other outside of S, $E(S) = (S : S)$ the edge set with both endpoints in S, and $G(S) = (S, E(S))$ the subgraph of G induced by the vertices in S. For simplicity, when $S = \{i\}$, $i \in V$, we write $\delta(i)$ instead of $\delta(\{i\})$. Given a vector q indexed on the edges set E and a subset of edges $F \subseteq E$, $q(F)$ denotes $\sum_{e\in F} q_e$. For simplicity, we write $x(S : S')$ instead of $x((S : S'))$.

The problem can be formulated by using the following binary variables: For each edge $e \in E$, and for each upgrade level $k = 1, \ldots, K$, let $x_e^k = 1$ if edge e is traversed exactly once after being upgraded to level k, $x_e^k = 0$ otherwise, and let $y_e^k = 1$ if edge e is traversed exactly twice after being upgraded to level k, $y_e^k = 0$ otherwise. In particular, the U-GTSP can be formulated as follows:

$$\text{(U-GTSP}(G)) \qquad \text{Minimize} \sum_{k=1}^{K}\sum_{e\in E}\left(c_e^k x_e^k + 2c_e^k y_e^k\right)$$

s.t.

$$\sum_{k=1}^{K}\left((x^k + 2y^k)(\delta(i))\right) \equiv 0 \pmod 2, \ \forall\ i\in V, \tag{1}$$

$$\sum_{k=1}^{K}\left((x^k + 2y^k)(\delta(S))\right) \geq 2, \ \forall\ S \subset V, \tag{2}$$

$$\sum_{k=1}^{K}\sum_{e\in E}\left(\alpha_e^k x_e^k + \alpha_e^k y_e^k\right) \leq T, \tag{3}$$

$$\sum_{k=1}^{K}\left(x_e^k + y_e^k\right) \leq 1, \ \forall\ e \in E \tag{4}$$

$$x_e^k, y_e^k \in \{0,1\}, \ \forall\ e\in E, \quad \forall\ k = 1, \ldots, K. \tag{5}$$

Constraints (1) force the tour to visit each vertex an even number of times, while conditions (2) ensure the route is connected. These two conditions, together with the binary conditions for the variables (5), ensure that the vector (x_e^k, y_e^k) is a GTSP tour (see, for example, Cornuéjols et al., 1985). Constraint (3) guarantees that the total cost does not exceed the budget T. Inequalities (4) prevent the solutions from traversing an edge with several different upgrade levels. This condition is partly ensured by the fact that any optimal U-GTSP tour traverses any edge at most twice, and this traversal is done with the lowest possible cost level. However, there is an infrequent exception. If an edge e has two equal costs, say $c_e^2 = c_e^3$, the above formulation without inequalities (4) could have an optimal solution in which $x_e^2 = x_e^3 = 1$, which is not a feasible U-GTSP solution. Note that, if in such a solution we replace $x_e^2 = x_e^3 = 1$ by $y_e^3 = 1$, we obtain an equivalent feasible U-GTSP tour, so inequalities (4) are not really necessary. However, we keep them so that there are no solutions of the formulation that are not feasible U-GTSP tours, and because they are helpful for strengthening the formulation.

Remark 1. Let H be a GTSP tour on G. Then, the problem of finding the optimal level of the edges in H without exceeding the budget T is a Multiple-Choice Knapsack Problem (MCKP). The MCKP is a generalization of the well-known knapsack problem, in which the set of items is partitioned into classes and we have to choose exactly one item of each class. In our problem, each edge of H represents a class of items. Then, if we define H^1, H^2 as the sets of edges traversed once or twice by H, respectively, we can formulate an MCKP to obtain the optimal levels of the edges in H as follows:

$$(\mathrm{MCKP}(H)) \qquad \text{Minimize} \sum_{e\in H^1} c_e^k x_e^k + \sum_{e\in H^2} 2c_e^k y_e^k$$

s.t.

$$\sum_{k=1}^{K}\sum_{e\in H^1} \alpha_e^k x_e^k + \sum_{k=1}^{K}\sum_{e\in H^2} \alpha_e^k y_e^k \le T \tag{6}$$

$$\sum_{k=1}^{K} x_e^k = 1, \ \forall e \in H^1 \tag{7}$$

$$\sum_{k=1}^{K} y_e^k = 1, \ \forall e \in H^2 \tag{8}$$

$$x_e^k \in \{0,1\}, \ \forall\, e\in H^1, \ \ \forall\, k = 1,\dots,K \tag{9}$$

$$y_e^k \in \{0,1\}, \ \forall\, e\in H^2, \ \ \forall\, k = 1,\dots,K, \tag{10}$$

where variables x_e^k (y_e^k) take value 1 if edge $e\in H^1$ ($e\in H^2$) is upgraded to level k (that is, if object number k of the class of objects associated with edge e is chosen in the MCKP) and 0 otherwise.

Remark 2. A feasible U-GTSP solution can be obtained by solving a GTSP on G using the cost associated with level 1 (or any other level, for that matter) and then solving a MCKP to find the upgrade levels of the edges that appear in the tour. However, this solution may not be optimal and there is no bound on how far the cost of such a solution may be from the optimal one. Consider, for example, the U-GTSP instance with two upgrade levels depicted in Fig. 1, where the numbers in brackets next to each edge represent the costs c_e^k of levels 1 and 2, respectively, and M is a big number. Let the upgrade costs be $\alpha_e^1 = 0$ and $\alpha_e^2 = 1$ for all the edges and consider a budget $T = 1$. If we solve the GTSP using the costs of the edges for level 1, an optimal GTSP tour is the sequence $(1, 2, 4, 3, 1)$. By solving the associated MCKP to obtain the optimal levels for this tour, we obtain that we can upgrade any edge of the tour we want to level 2, but only one of them, thus resulting in a U-GTSP solution with cost $2M + 3$. However, the optimal U-GTSP solution consists of upgrading edge $(1, 4)$ to level 2 and then traversing the sequence $(4, 3, 4, 1, 4, 2, 4)$, with total cost 10. Nevertheless, this idea of obtaining GTSP tours and finding the upgrade levels solving a MCKP will later be used to develop a primal heuristic that can provide good upper bounds.

3. The polytope of solutions

The feasible U-GTSP solutions without considering the budget limitation (constraint (3)) define a polytope that is studied in this section.

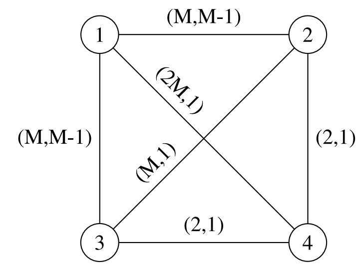

Fig. 1. U-GTSP instance.

We find its dimension, show that some of the constraints of the formulation define facets, and present some other valid inequalities that reinforce the formulation.

For a polyhedral study it is necessary to describe many feasible solutions that are affinely independent. This procedure, however, is particularly complicated in problems in which some constraints limit the length or 'cost' of the routes, because the feasibility of a route depends on the specific limits imposed by these constraints (the parameters α_e^k and T in the case of the U-GTSP). In these cases, even determining the dimension of the polytope defined as the convex hull of the feasible solutions is a very difficult task. However, if we remove the total budget constraint (3) of our problem, we obtain a relaxed problem, that we will call *Relaxed* U-GTSP, RU-GTSP, whose polytope can indeed be studied. This study is interesting because some of the facets found could also be facets of the original polyhedron, and it is a way to ensure the goodness of the inequalities used.

Hence, in this section, let a *RU-GTSP solution* denote any GTSP tour on graph G with a level $k \in \{1, \dots, K\}$ assigned to each traversed edge, whether or not it satisfies the total budget constraint (3). Associated with each RU-GTSP solution we consider an incidence vector

$$\left((x^1, y^1), (x^2, y^2), \dots, (x^K, y^K)\right) \in \mathbb{Z}^{2K|E|},$$

where each vector x^k and y^k is indexed on the set of edges $e \in E$ and variables x_e^k, y_e^k are defined as above. Note that the sum $\sum_{k=1}^{K}(x^k, y^k)$ is a GTSP tour on G.

Let RU-GTSP(G) be the convex hull of all the incidence vectors of RU-GTSP solutions in G. It is a polytope in the $\mathbb{R}^{2K|E|}$ space. To study this polytope, we first need to study the polytope associated with the GTSP using a formulation similar to the one we propose in this paper for the U-GTSP.

Let GTSP(G) be the convex hull of all incidence vectors (x, y) of GTSP tours on G, where, for each edge $e \in E$, x_e takes the value 1 if edge e is traversed exactly once (and zero otherwise) and y_e takes the value 1 if edge e is traversed exactly twice (and zero otherwise), i.e., the vectors $(x, y) \in \mathbb{Z}^{2|E|}$ satisfying:

$$(x + 2y)\left(\delta(i)\right) \equiv 0 \pmod 2, \ \forall\, i\in V, \tag{11}$$

$$(x + 2y)\left(\delta(S)\right) \ge 2, \ \forall\, S \subset V, \tag{12}$$

$$x_e + y_e \le 1, \ \forall\, e \in E \tag{13}$$

$$x_e, y_e \in \{0,1\}, \ \forall\, e\in E. \tag{14}$$

For the sake of simplicity we will call *GTSP tour* both the closed walk and its incidence vector.

Theorem 1. *$\dim(GTSP(G)) = 2|E|$ (GTSP(G) is a full-dimensional polytope) if, and only if, G is a 3-edge connected graph.*

Proof. Let us suppose that G is not a 3-edge connected graph. Then, there is a cut-set $\delta(S)$ that contains at most two edges. If $\delta(S) = \{e\}$ then all the GTSP tours satisfy the equations $x_e = 0$ and $y_e = 1$. Moreover, If $\delta(S) = \{e, f\}$ then all the GTSP tours satisfy $x_e = x_f$. Therefore, the polytope is not full-dimensional.

Let us now suppose that G is a 3-edge connected graph and prove that we have a full-dimensional polytope. Let $ax+by=c$ be an equation satisfied by all the GTSP tours and let us prove that $a=b=c=0$.

Consider the tour with $x_e=0, y_e=1$ for every edge $e\in E$. It is a GTSP tour because it is connected, all the vertices have even degree, and it visits all the vertices of G. Hence, it must satisfy $ax+by=c$ and we obtain

$$b(E)=c. \tag{15}$$

Let $e\in E$ be any edge. Making $y_e=0$ in the previous GTSP tour, and given that G is a 3-edge connected graph, we get another GTSP tour, which also satisfies $ax+by=c$ and we obtain

$$b\big(E\backslash\{e\}\big)=c. \tag{16}$$

By subtracting (15) and (16) we obtain $b_e=0$, for all $e\in E$. Furthermore, after substituting $b_e=0$, for all $e\in E$, in (15), we obtain that $c=0$.

Let C be any cycle in graph G. The tour with all the variables taking the values $x_e=0, y_e=1$ except for the edges $e\in C$, which take the values $x_e=1, y_e=0$, is a GTSP tour which must therefore satisfy $ax+by=c$, and we obtain that

$$a(C)=c\ (=0). \tag{17}$$

Let $e=(i,j)\in E$ be any edge. Given that G is a 3-edge connected graph, there are two edge-disjoint paths $\mathcal{P}_1$ and $\mathcal{P}_2$ connecting vertices i,j that do not traverse edge e. Then, by applying (17) to the three cycles $\mathcal{P}_1\cup\{e\}$, $\mathcal{P}_2\cup\{e\}$, and $\mathcal{P}_1\cup\mathcal{P}_2$, we have

$$a_e+a\big(\mathcal{P}_1\big)=0,\quad a_e+a\big(\mathcal{P}_2\big)=0,\quad a\big(\mathcal{P}_1\big)+a\big(\mathcal{P}_2\big)=0$$

from where we obtain that $a_e=0$, for all $e\in E$. Hence, we have that $a=b=c=0$ and we are done. ♦

Theorem 2. *$\dim(RU\text{-}GTSP(G))=2K|E|$ (RU-GTSP(G) is a full-dimensional polytope) if, and only if, G is a 3-edge connected graph.*

Proof. If G is not a 3-edge connected graph, a reasoning similar to that of the previous theorem concludes that the polytope is not full-dimensional.

On the other hand, if G is a 3-edge connected graph, from Theorem 1 the dimension of GTSP(G) is $2|E|$. Then, there exist $2|E|+1$ affinely independent, and also $2|E|$ linearly independent GTSP tours on graph G.

Let $\bar{A}$ be the incidence matrix of the corresponding $2|E|+1$ affinely independent GTSP tours expressed as rows indexed by the variables x_e, y_e as columns. This matrix $\bar{A}$ has $2|E|+1$ rows and we can assume, w.l.o.g., that,

- after removing the first row we obtain a matrix, say A, with rank $2|E|$ and
- after subtracting the first row of $\bar{A}$ from the rows of A we obtain a matrix, say M, also with rank $2|E|$.

Given any of the above GTSP tours (x,y) and a given upgrade level $k\in\{1,\dots,K\}$, if we define $(x^k,y^k)=(x,y)$ and $(x^{k'},y^{k'})=(\vec{0},\vec{0})$ for any $k'\neq k$, where $\vec{0}$ represents a vector with zeros indexed on the edges $e\in E$, we obtain a RU-GTSP solution. By expressing these solutions as rows, we can build a matrix similar to that shown in Fig. 2a for $K=3$.

This matrix has $2K|E|+1$ rows representing RU-GTSP solutions. If we subtract the first row from all the others and remove the first row we obtain the matrix in Fig. 2b, where r_1 is the first row of matrix $\bar{A}$. The rank of this matrix is $2K|E|$ and, hence, we have $2K|E|+1$ affinely independent RU-GTSP solutions and the dimension of the polyhedron is $2K|E|$. ♦

In what follows, we will assume that G is a 3-edge connected graph and thus both GTSP(G) and RU-GTSP(G) are full-dimensional polytopes. In this case, each facet is induced by a unique inequality (except scalar multiples).

Theorem 3. *Inequality $y_e\geq 0$, for each edge $e\in E$, is facet-inducing of GTSP(G).*

Proof. Let $ax+by\geq c$ be a valid inequality such that

$$\{(x,y)\in \text{GTSP}(G):\quad y_e=0\}\ \subseteq\ \{(x,y)\in\text{GTSP}(G):\quad ax+by=c\}.$$

We have to prove that $ax+by\geq c$ is a scalar multiple of $y_e\geq 0$.

Consider the GTSP tour having all the variables $x_{e'}=0$, $y_{e'}=1$ for every edge $e'\in E\setminus\{e\}$, and $x_e=y_e=0$ (it is connected, it is even, and it visits all the vertices of G). Given that it satisfies $y_e=0$, it also satisfies $ax+by=c$ and

$$b\big(E\backslash\{e\}\big)=c \tag{18}$$

holds. Consider any edge $f\in E\setminus\{e\}$. Since G is 3-edge connected, making $x_f=y_f=0$ in the previous tour, we obtain another GTSP tour satisfying $y_e=0$, which also satisfies $ax+by=c$ and we have

$$b\big(E\backslash\{e,f\}\big)=c. \tag{19}$$

By subtracting the two previous equations, we obtain that $b_f=0$. This is true for all $f\in E\backslash\{e\}$ and, after substituting in (18), we obtain $c=0$.

Let C be any cycle in graph G that does not traverses edge e. The GTSP tour with variables $x_{e'}=1$, $y_{e'}=0$ for the edges in C, $x_e=y_e=0$, and $x_{e'}=0$, $y_{e'}=1$ for all the other edges, satisfies $y_e=0$ and, then,

$$a(C)+b\big(E\backslash(C\cup\{e\})\big)=c=0\implies a(C)=0 \tag{20}$$

holds for any cycle C not traversing e.

Let C be any cycle in graph G that traverses the edge e. The GTSP tour with variables $x_{e'}=1$, $y_{e'}=0$ for the edges in C, and $x_{e'}=0$, $y_{e'}=1$ for all the other edges, satisfies $y_e=0$ and, then,

$$a(C)+b\big(E\backslash C\big)=c=0\implies a(C)=0 \tag{21}$$

holds also for the cycles C in G traversing the edge e.

Let $f=(i,j)\in E$ be an arbitrary edge. Since G is 3-edge connected, there are two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$ joining vertices i and j that do not contain edge f. Then, considering the three cycles $\mathcal{P}_1\cup\{f\}$, $\mathcal{P}_2\cup\{f\}$, and $\mathcal{P}_1\cup\mathcal{P}_2$, for which $a\big(C\big)=0$ holds, we have

$$a_f+a\big(\mathcal{P}_1\big)=0,\quad a_f+a\big(\mathcal{P}_2\big)=0,\quad a\big(\mathcal{P}_1\big)+a\big(\mathcal{P}_2\big)=0,$$

from where we obtain that $a_f=0$, and this is true for every $f\in E$. By substituting all the previous values for a,b and c in $ax+by\geq c$ we obtain $b_e y_e\geq 0$, which is a scalar multiple of $y_e\geq 0$. ♦

Theorem 4. *Inequality $y_e^k\geq 0$, for each edge $e\in E$ and for each level k, is facet-inducing of RU-GTSP(G).*

Proof. W.l.o.g., let us suppose that $k=1$. If G is a 3-edge connected graph, from Theorem 3 the inequality $y_e\geq 0$ induces a facet of GTSP(G) and there are $2|E|-1$ linearly independent GTSP tours on graph G satisfying $y_e=0$.

Let B be the incidence matrix of these $2|E|-1$ GTSP linearly independent tours as rows indexed by the variables x,y as columns. Furthermore, let A be the matrix of the proof of Theorem 2, with rank $2|E|$.

If we assign the level $k=1$ to all the GTSP tours in matrix B and the vector zero to the remaining levels $k>1$ we obtain RU-GTSP solutions on G satisfying $y_e^1=0$. On the other hand, if we assign a given level $k>1$ to all the GTSP tours in matrix A and the vector zero to all the remaining levels, we obtain a RU-GTSP solution on G satisfying also $y_e^1=0$. Hence, we can build a matrix similar to the matrix in Fig. 3, which has been written for the case $K=3$ for the sake of simplicity.

This matrix has $2K|E|-1$ rows representing RU-GTSP solutions that satisfy $y_e^1=0$. The rank of this matrix is $2|E|K-1$ and, therefore, we have $2|E|K-1$ linearly independent RU-GTSP solutions satisfying $y_e^1=0$ and the inequality $y_e^k\geq 0$ is facet-inducing of RU-GTSP(G). ♦

$$
\begin{array}{ccc}
(x^1,y^1) & (x^2,y^2) & (x^3,y^3) \\
\hline
\bar{A} & 0 & 0 \\
0 & A & 0 \\
0 & 0 & A
\end{array}
$$

(a)

$$
\begin{array}{ccc}
(x^1,y^1) & (x^2,y^2) & (x^3,y^3) \\
\hline
M & 0 & 0 \\
\begin{array}{c}-r_1\\ \cdots \\ -r_1\end{array} & A & 0 \\
\begin{array}{c}-r_1\\ \cdots \\ -r_1\end{array} & 0 & A
\end{array}
$$

(b)

Fig. 2. Matrices appearing in the proof of Theorem 2.

$$
\begin{array}{ccc}
(x^1,y^1) & (x^2,y^2) & (x^3,y^3) \\
\hline
B & 0 & 0 \\
0 & A & 0 \\
0 & 0 & A
\end{array}
$$

Fig. 3. Matrix appearing in the proof of Theorem 4.

Theorem 5. *Inequality $x_e \geq 0$, for each edge $e \in E$, is facet-inducing of GTSP(G) if, and only if, graph $G \setminus \{e\}$ is 3-edge connected.*

Proof. If graph $G \setminus \{e\}$ is not 3-edge connected (but G is), there is at least one cut-set with three edges, $\delta(S) = \{e, f, g\}$. In this case, inequality $x_e \geq 0$ is not facet-inducing because it can be obtained as the sum of the following two "parity inequalities" (28) that will be introduced later:

$$x_e + x_g \geq x_f, \quad x_e + x_f \geq x_g$$

Let us suppose that $G \setminus \{e\}$ is a 3-edge connected graph and let $ax + by \geq c$ be a valid inequality such that

$$\{(x,y) \in \mathrm{GTSP}(G): \quad x_e = 0\} \subseteq \{(x,y) \in \mathrm{GTSP}(G): \quad ax + by = c\}.$$

We will show that $ax + by \geq c$ is a scalar multiple of $x_e \geq 0$.

Consider the GTSP tour having all the variables $x_{e'} = 0$, $y_{e'} = 1$ for every edge $e' \in E$ (it is connected, it is even, and it visits all the vertices of G). Given that it satisfies $x_e = 0$, it also satisfies $ax + by = c$ and $b(E) = c$ holds. Consider any edge $f \in E$ and the GTSP tour having all the variables $x_{e'} = 0$, $y_{e'} = 1$ for every edge $e' \neq f$, and $x_f = y_f = 0$, which satisfies $x_e = 0$ and, hence, it also satisfies $ax + by = c$ and $b(E \setminus \{f\}) = c$ holds. By subtracting from $b(E) = c$ we obtain that $b_f = 0$. This is true for all $f \in E$ and, hence, we obtain also that also $c = 0$.

Let C be any cycle in graph G not traversing the edge e. The tour with all the variables $x_{e'} = 0$, $y_{e'} = 1$ except those $x_{e'} = 1$, $y_{e'} = 0$ for the edges $e' \in C$ is a GTSP tour (is even, connected and visits all the vertices of G) satisfying $x_e = 0$ and, then,

$$a(C) + b(E \setminus C) = c = 0 \implies a(C) = 0 \tag{22}$$

holds for any cycle C in G not traversing edge e.

Consider any edge $f = (i,j) \in E \setminus \{e\}$. Given that $G \setminus \{e\}$ is a 3-edge connected graph, we can consider two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$ connecting i, j and not traversing edge f (nor e, obviously). Then, for the three cycles $\mathcal{P}_1 \cup \{f\}$, $\mathcal{P}_2 \cup \{f\}$, and $\mathcal{P}_1 \cup \mathcal{P}_2$ we have

$$a_f + a(\mathcal{P}_1) = 0, \quad a_f + a(\mathcal{P}_2) = 0, \quad a(\mathcal{P}_1) + a(\mathcal{P}_2) = 0$$

and we obtain that $a_f = 0$, for all $f \neq e$. By substituting in $ax + by \leq c$ we obtain $a_e x_e \geq 0$, which is a scalar multiple of $x_e \geq 0$. ♦

Theorem 6. *Inequality $x_e^k \geq 0$, for each edge $e \in E$ and for each level k, is facet-inducing of RU-GTSP(G) (if graph $G \setminus \{e\}$ is 3-edge connected).*

Proof. The proof is similar to that of Theorem 4 and is omitted here for the sake of brevity. ♦

Note that trivial inequalities $x_e^k \leq 1$ and $y_e^k \leq 1$, for each edge $e \in E$, are dominated by inequalities $\sum_{k=1}^{K} \left(x_e^k + y_e^k\right) \leq 1$, which are valid for the RU-GTSP, and, therefore, they cannot induce a facet of RU-GTSP(G).

The following theorem establishes some conditions for a facet-inducing inequality of GTSP(G) to provide a facet-inducing inequality of RU-GTSP(G):

Theorem 7. *Let $f(x,y) \geq \alpha$, with $\alpha \neq 0$, be a facet-inducing inequality of GTSP(G). Then, inequality $\sum_{k=1}^{K} f(x^k, y^k) \geq \alpha$ is facet-inducing of RU-GTSP(G).*

Proof. If the inequality $f(x,y) \geq \alpha$ induces a facet of GTSP(G) and $\alpha \neq 0$, then the vector zero is not in the affine hull of the points of the polytope that satisfy $f(x,y) = \alpha$, and there exist $2|E|$ linearly independent GTSP tours on graph G satisfying $f(x,y) = \alpha$. Let B be the incidence matrix of these $2|E|$ GTSP tours expressed as rows indexed by the variables x, y as columns.

If we assign a given level k to all the above GTSP tours in matrix B and the vector zero to all the other levels, we obtain a RU-GTSP solution on G satisfying $\sum_{k=1}^{K} f(x^k, y^k) = \alpha$. Hence, we can build a matrix similar to that in Fig. 4.

This matrix has $2K|E|$ rows representing RU-GTSP solutions satisfying $\sum_{k=1}^{K} f(x^k, y^k) = \alpha$. The rank of this matrix is $2K|E|$ and, hence, the $2K|E|$ RU-GTSP solutions are linearly independent and the inequality $\sum_{k=1}^{K} f(x^k, y^k) \geq \alpha$ is facet-inducing of RU-GTSP(G). ♦

$$\begin{array}{c|c|c}
(x^1, y^1) & (x^2, y^2) & (x^3, y^3) \\
\hline
B & 0 & 0 \\
\hline
0 & B & 0 \\
\hline
0 & 0 & B
\end{array}$$

Fig. 4. Matrix appearing in the proof of Theorem 7.

We will use Theorem 7 to show that some families of valid inequalities are facet-inducing of RU-GTSP(G). Note that this theorem does not apply to inequalities $x_e \geq 0$ (or $y_e \geq 0$) because, since the zero vector is obviously in the affine hull of the points that satisfy $x_e = 0$, there are only $2|E|-1$ linearly independent GTSP tours satisfying $x_e \geq 0$. Indeed, $\sum_{k=1}^{K} x_e^k \geq 0$ does not induce a facet of RU-GTSP(G).

Theorem 8. *Inequality $x_e + y_e \leq 1$, for each edge $e \in E$, is facet-inducing of GTSP(G).*

Proof. Let us suppose there is another valid inequality $ax + by \leq c$ such that

$$\{(x, y) \in \text{GTSP}(G) : \quad x_e + y_e = 1\} \subseteq \{(x, y) \in \text{GTSP}(G) : \quad ax + by = c\}.$$

We will prove that inequality $ax + by \leq c$ is a scalar multiple of $x_e + y_e \leq 1$.

Consider the GTSP tour having all the variables $x_{e'} = 0$, $y_{e'} = 1$ for every edge $e' \in E$ (it is connected, it is even, and it visits all the vertices of G). Given that it satisfies $x_e + y_e = 1$, it also satisfies $ax + by = c$ and

$$b(E) = c \tag{23}$$

holds. Consider any edge $f \in E \setminus \{e\}$. Since G is 3-connected, making $y_f = 0$ in the previous tour, we obtain another feasible GTSP tour satisfying $x_e + y_e = 1$, which also satisfies $ax + by = c$ and

$$b\big(E \backslash \{f\}\big) = c \tag{24}$$

holds. By subtracting (23) from (24), we obtain that $b_f = 0$. This is true for all $f \in E \setminus \{e\}$ and, from (23), $b_e = c$.

Let C be any cycle in graph G. The tour with all the variables $x_{e'} = 1$, $y_{e'} = 0$ for the edges in C, and $x_{e'} = 0$, $y_{e'} = 1$ for the edges not in C, is a GTSP tour satisfying $x_e + y_e = 1$ and, then,

$$a(C) + b\big(E \backslash C\big) = c \tag{25}$$

holds. Then, if $e \in C$ we obtain that $a\big(C\big) = c$ while if $e \notin C$ we obtain that $a\big(C\big) + b_e = c$, i.e., $a\big(C\big) = 0$.

Let $f = (i, j) \neq e$ be an arbitrary edge. Since G is 3-edge connected, there are two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$ joining vertices i and j that do not contain edge f. Then, considering the three cycles $\mathcal{P}_1 \cup \{(i, j)\}$, $\mathcal{P}_2 \cup \{(i, j)\}$, and $\mathcal{P}_1 \cup \mathcal{P}_2$, we have

$$a_f + a\big(\mathcal{P}_1\big) = 0, \quad a_f + a\big(\mathcal{P}_2\big) = 0, \quad a\big(\mathcal{P}_1\big) + a\big(\mathcal{P}_2\big) = 0, \quad \text{if } e \notin \mathcal{P}_1 \cup \mathcal{P}_2, \quad \text{and}$$

$$a_f + a\big(\mathcal{P}_1\big) = c, \quad a_f + a\big(\mathcal{P}_2\big) = 0, \quad a\big(\mathcal{P}_1\big) + a\big(\mathcal{P}_2\big) = c, \quad \text{if } e \in \mathcal{P}_1, \text{for example.}$$

In both cases we obtain $a_f = 0$, and this is true for every $f \neq e$. On the other hand, for the edge e, by considering its corresponding two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$, we have

$$a_e + a\big(\mathcal{P}_1\big) = c, \quad a_e + a\big(\mathcal{P}_2\big) = c, \quad a\big(\mathcal{P}_1\big) + a\big(\mathcal{P}_2\big) = 0,$$

and we obtain that $a_e = c$. By substituting in $ax + by \leq c$ we obtain $cx_e + cy_e \leq c$, which is a scalar multiple of $x_e + y_e \leq 1$. ♦

Theorem 9. *Inequality (4), $\sum_{k=1}^{k}(x_e^k + y_e^k) \leq 1$, for each edge $e \in E$, is facet-inducing of RU-GTSP(G).*

Proof. It is an immediate consequence of Theorems 7 and 8. ♦

Theorem 10. *Connectivity inequality (12), $(x + 2y)(\delta(S)) \geq 2$, for each $S \subset V$, is facet-inducing of GTSP(G) if graph G is 3-edge connected and graphs $G(S)$ and $G(V \setminus S)$ are 2-edge connected.*

Proof. Let $ax + by \geq c$ be a valid inequality such that

$$\{(x, y) \in \text{GTSP}(G) : \quad (x + 2y)(\delta(S)) = 2\}$$
$$\subseteq \{(x, y) \in \text{GTSP}(G) : \quad ax + by = c\}.$$

We have to prove that the $ax + by \geq c$ is a scalar multiple of $(x + 2y)(\delta(S)) \geq 2$.

Consider an arbitrarious edge $f \in \delta(S)$ and the GTSP tour having all the variables $x_{e'} = 0$, $y_{e'} = 1$ for every edge e' in $G(S)$ and $G(V \backslash S)$, $x_f = 0$, $y_f = 1$, and $x_{e'} = y_{e'} = 0$ for every edge in $\delta(S) \setminus \{f\}$ (it is connected, it is even, and it visits all the vertices of G). Given that it satisfies $(x + 2y)(\delta(S)) = 2$, it also satisfies $ax + by = c$ and

$$b(E \backslash \delta(S)) + b_f = c \tag{26}$$

holds. Consider any edge $e \in E \setminus \delta(S)$. Since G is 3-edge connected and graphs $G(S)$ and $G(V \setminus S)$ are 2-edge connected, making $y_e = 0$ in the previous tour, we obtain another GTSP tour satisfying $(x+2y)(\delta(S)) = 2$. Therefore, it also satisfies $ax + by = c$ and

$$b(E \backslash (\delta(S) \cup \{e\})) + b_f = c \tag{27}$$

holds. By subtracting the two previous equation we obtain that $b_e = 0$. This is true for all $e \in E \setminus \delta(S)$. Therefore, from (26) we have $b_f = c$, and this holds for every $f \in \delta(S)$.

Let C be any cycle in graph $G(S)$. The first tour described in this proof except for the variables $x_{e'} = 1$, $y_{e'} = 0$ for every edge $e' \in C$ is a GTSP tour satisfying $(x + 2y)(\delta(S)) = 2$ and, by subtracting the corresponding two equations we obtain $a(C) = 0$. The same reasoning is valid for any cycle in graph $G(V \setminus S)$.

Let $f = (i, j) \in E(S)$ be any edge. Given that G is a 3-edge connected graph, we can consider two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$ connecting vertices i, j and not traversing edge f. We consider several cases:

(a) $\mathcal{P}_1$, $\mathcal{P}_2$ are in $G(S)$. Then, by considering the three cycles $\mathcal{P}_1 \cup \{f\}$, $\mathcal{P}_2 \cup \{f\}$, and $\mathcal{P}_1 \cup \mathcal{P}_2$, for which

$$a_f + a\big(\mathcal{P}_1\big) = 0, \quad a_f + a\big(\mathcal{P}_2\big) = 0, \quad a\big(\mathcal{P}_1\big) + a\big(\mathcal{P}_2\big) = 0,$$

holds, we obtain that $a_f = 0$.

(b) $\mathcal{P}_1$ is in $G(S)$ and $\mathcal{P}_2$ traverses $\delta(S)$ exactly twice (if $\mathcal{P}_2$ does not meet this condition, it can be transformed into another path that does, simply by short-circuiting $\mathcal{P}_2$ in the connected subgraph $G(V \backslash S)$). Let (x^1, y^1) be the GTSP tour that traverses the edge f and $\mathcal{P}_2$ once, the path $\mathcal{P}_1$ twice, and the remaining edges in $G(S)$ and $G(V \backslash S)$ twice. Let (x^2, y^2) be the tour obtained from (x^1, y^1) after removing the edge f and one of the two copies of the edges in the path $\mathcal{P}_1$. Both GTSP tours satisfy the connectivity inequality as an equality and, from them, we have $a_f + b(\mathcal{P}_1) = a(\mathcal{P}_1)$. Given that $b(\mathcal{P}_1) = 0$ and $a_f + a(\mathcal{P}_1) = 0$ because $\mathcal{P}_1 \cup f$ is a cycle, we obtain that $a_f = 0$.

(c) Both $\mathcal{P}_1$ and $\mathcal{P}_2$ traverse the cut-set $\delta(S)$ (exactly twice). We can construct two edge-disjoint paths joining i and j, where one of them is in $G(S)$, and so we are in the same situation as in (b). If path $\mathcal{P}_1$ (or $\mathcal{P}_2$) has any edge incident with i or j inside $G(S)$, then $\mathcal{P}_1$ can be short-circuited to obtain a new path in $G(S)$ and we are done. Otherwise, both paths $\mathcal{P}_1$ and $\mathcal{P}_2$ traverse the cut-set $\delta(S)$ directly from i and j. Since graph $G(S)$ is 2-edge connected, there is a path $\mathcal{P}_3$ in $G(S)$ also joining i and j and not containing (i,j). By construction, $\mathcal{P}_3$ and $\mathcal{P}_1$ (or $\mathcal{P}_2$) are edge-disjoint paths, and we are done.

The previous reasoning is also valid for any edge $f=(i,j)\in E(V\backslash S)$. Hence, we have $a_e=b_e=0$, for all $e\in E\setminus\delta(S)$.

Let us suppose that $\delta(S)=\{e_1,e_2,\dots,e_q\}$, with $q\geq 3$ because G is a 3-edge connected graph. Given $e_1,e_2\in\delta(S)$, let (x^1,y^1) be the GTSP tour having all the variables $x_{e'}=0$, $y_{e'}=1$ for every edge in $G(S)$ and $G(V\setminus S)$ and $x_{e_1}=0$, $y_{e_1}=1$, and let (x^2,y^2) be the GTSP tour obtained from (x^1,y^1) after replacing the second traversal of e_1 with the traversals of the edges in a path joining the endpoints of e_1, using e_2 (if any edge appears three times, we would remove two copies of them). By subtracting $ax^1+by^1=c$ from $ax^2+by^2=c$, we obtain $b_{e_1}=a_{e_1}+a_{e_2}$. If we exchange the roles of e_1 and e_2, we obtain that $b_{e_2}=a_{e_1}+a_{e_2}$. Repeating this argument with all the pairs of edges in $\delta(S)$ (recall that $b_{e_i}=c$), we obtain $a_{e_i}=a_{e_j}=\frac{c}{2}$ for all $i\neq j\in\{1,\dots,q\}$ (because $q\geq 3$ holds). By substituting in $ax+by\geq c$, we obtain $(\frac{c}{2}x+cy)(\delta(S))\geq c$, which is a scalar multiple of $(x+2y)(\delta(S))\geq 2$. ♦

Theorem 11. *Connectivity inequality (2), $\sum_{k=1}^{K}(x^k+2y^k)(\delta(S))\geq 2$, for each $S\subset V$, is facet-inducing of RU-GTSP(G) if graph G is 3-edge connected and graphs $G(S)$ and $G(V\setminus S)$ are 2-edge connected.*

Proof. It is an immediate consequence of Theorems 7 and 10. ♦

3.1. Parity inequalities

Constraints (1) for the U-GTSP, and constraints (11) for the GTSP with the formulation we propose, are not linear inequalities. To make the solution even at all the vertices of the graph, and also at all the cut-sets, we use some linear inequalities inspired by the *parity inequalities* proposed in Corberán et al. (2013) for the MBCPP. They are based on the *co-circuit inequalities* presented in Barahona and Grötschel (1986) and are defined as follows. Given a cut-set $\delta(S)$ defined by a vertex set $S\subset V$, and an edge set $F\subseteq\delta(S)$ with $|F|$ odd, we will call

- *parity inequalities* for the GTSP on G to

$$x\big(\delta(S)\setminus F\big)\geq x(F)-|F|+1 \tag{28}$$

- and the corresponding *parity inequalities* for the U-GTSP on G to

$$\sum_{k=1}^{K}x^k\big(\delta(S)\setminus F\big)\geq\sum_{k=1}^{K}x^k(F)-|F|+1. \tag{29}$$

Theorem 12. *Parity inequalities* (28) *and* (29) *are valid for the GTSP and the U-GTSP, respectively.*

Proof. (a) Consider an arbitrarious GTSP tour $(\bar{x},\bar{y})$ on G and let us prove that $\bar{x}\big(\delta(S)\setminus F\big)\geq\bar{x}(F)-|F|+1$ holds. If $(\bar{x},\bar{y})$ satisfies $\bar{x}(F)\leq|F|-1$, then the inequality becomes $\bar{x}\big(\delta(S)\setminus F\big)\geq 0$, which is obviously satisfied. If $(\bar{x},\bar{y})$ satisfies $\bar{x}(F)=|F|$, then $\bar{x}_e=1$ for each $e\in F$, and this implies that $\bar{y}_e=0$ for each $e\in F$. Therefore, $(\bar{x},\bar{y})$ traverses each edge in F exactly once and the inequality reduces to $\bar{x}\big(\delta(S)\setminus F\big)\geq 1$. Since $(\bar{x},\bar{y})$ must traverse the cut-set $\delta(S)$ an even number of times and $|F|$ is odd, it must traverse $\delta(S)\setminus F$ an odd number of times, so $\bar{x}_e=1$ must hold for at least one edge $e\in\delta(S)\setminus F$.

(b) Consider an arbitrarious U-GTSP solution $\big((\bar{x}^1,\bar{y}^1),(\bar{x}^2,\bar{y}^2),\dots,(\bar{x}^K,\bar{y}^K)\big)$. If we add all the levels, $\big(\sum_{k=1}^{K}\bar{x}^k,\sum_{k=1}^{K}\bar{y}^k\big)$ is a GTSP tour on G and, from (a), it satisfies

$$\sum_{k=1}^{K}\bar{x}^k\big(\delta(S)\setminus F\big)\geq\sum_{k=1}^{K}\bar{x}^k(F)-|F|+1.$$

♦

Remark 3. In order to prove that parity inequalities (28) induce facets of GTSP(G) in Theorem 13, we have to build several GTSP tours on G satisfying (28) with equality. Recall that, given a subset $T\subset V$, with $|T|$ even, a T-join is a subset of edges $E'\subset E$ such that, in the subgraph $G'=(V,E')$, the degree of v is odd if and only if $v\in T$, and that a connected graph G has a T-join for each set $T\subset V$ with $|T|$ even. Suppose we choose either to traverse exactly once all the edges in F but one, and zero or two times the other edges in $\delta(S)$, or to traverse exactly once all the edges in F plus one more edge in $\delta(S)$, and zero or two times the remaining edges in $\delta(S)$. Note that if we can add edges in $G(S)$ and $G(V\setminus S)$ to obtain a GTSP tour, this GTSP tour will satisfy (28) with equality. This can be done as follows:

Assume that $G(S)$ and $G(V\setminus S)$ are connected. Let $T\subset S$ be the set of vertices incident with an odd number of the edges in $\delta(S)$ selected to be traversed exactly once. Given that $|T|$ is even, there is a T-join E' in $G(S)$. This same process is done in $G(V\setminus S)$ and we have a T-join E'' in $G(V\setminus S)$. Then, by adding two copies of all the remaining edges in $G(S)$ and $G(V\setminus S)$ not in the T-joins, we obtain a GTSP tour (even, connected and visiting all the vertices in V) that satisfies (28) with equality.

Theorem 13. *Parity inequality (28) is facet-inducing of GTSP(G) if graph G is 3-edge connected and graphs $G(S)$ and $G(V\backslash S)$ are 2-edge connected.*

Proof. Parity inequality (28) can be written as $x(F)-x\big(\delta(S)\setminus F\big)\leq|F|-1$. Let $ax+by\leq c$ be a valid inequality such that

$$\{(x,y)\in\text{GTSP}(G):\quad x(F)-x\big(\delta(S)\setminus F\big)=|F|-1\}$$
$$\subseteq\ \{(x,y)\in\text{GTSP}(G):\quad ax+by=c\}.$$

We will prove that $ax+by\leq c$ is a scalar multiple of $x(F)-x\big(\delta(S)\setminus F\big)\leq|F|-1$.

Let $e\in E(S)\cup E(V\setminus S)$. Since $G(S)$ and $G(V\setminus S)$ are 2-edge connected, they would remain connected after removing edge e and, as in Remark 3, we can build a GTSP tour satisfying $x(F)-x\big(\delta(S)\setminus F\big)=|F|-1$ that does not traverse e. Consider the same tour but adding two traversals of e. By comparing the two tours we obtain that $b_e=0$.

Let $e=(i,j)\in E(S)$ be any edge. Given that G is a 3-edge connected graph and $G(S)$ is a 2-edge connected graph, we can construct two edge-disjoint paths $\mathcal{P}_1$, $\mathcal{P}_2$ connecting vertices i,j and not traversing edge e, such that at least one of them is in $G(S)$.

If both paths $\mathcal{P}_1$ and $\mathcal{P}_2$ are in $G(S)$, we can build a GTSP tour (x^1,y^1) in G satisfying $x(F)-x\big(\delta(S)\setminus F\big)=|F|-1$ such that it traverses edge $e=(i,j)$ exactly once. To do that, the "parity" label of vertices i and j is switched before the T-join is computed and then the edge $e=(i,j)$ is added. We define three more GTSP tours in the following way:

The tour (x^2,y^2) is obtained from (x^1,y^1) by adding to it one traversal of each edge in paths $\mathcal{P}_1$ and $\mathcal{P}_2$ and then removing two traversals of each of these edges traversed three times. The tour (x^3,y^3) is obtained from (x^1,y^1) by removing the traversal of edge e, adding one traversal of each edge in path $\mathcal{P}_1$, and then removing two traversals of each one of these edges traversed three times. The tour (x^4,y^4) is obtained from (x^1,y^1) by removing the traversal of edge e, adding one traversal of each edge in path $\mathcal{P}_2$, and then removing two traversals of each one of these edges traversed three times. All these tours are GTSP tours that satisfy $x(F)-x\big(\delta(S)\setminus F\big)=|F|-1$ and then also satisfy $ax+by=c$.

Let us define $\alpha(\mathcal{P}_i)=\sum_{e\in\mathcal{P}_i^1}a_e+\sum_{e\in\mathcal{P}_i^2}b_e$, where $\mathcal{P}_i^1$ is the set of edges in path $\mathcal{P}_i$ that are traversed only once in (x^1,y^1) and $\mathcal{P}_i^2$ the set of edges in path $\mathcal{P}_i$ that are not traversed at all or are traversed twice in (x_1,y_1). If we subtract the expression $ax^1+by^1=c$ from $ax^2+by^2=c$, we obtain $\alpha(\mathcal{P}_1)+\alpha(\mathcal{P}_2)=0$. In the same way, with the tours 3 and 4 above, we obtain $\alpha(\mathcal{P}_1)=\alpha(\mathcal{P}_2)$ and then $\alpha(\mathcal{P}_1)=\alpha(\mathcal{P}_2)=0$. Finally, with the tours 1 and 3, we obtain $a_e=\alpha(P1)$ and then $a_e=0$.

Let us now suppose that path $\mathcal{P}_2$ is not in $G(V\backslash S)$, i.e. it leaves the graph $G(V\setminus S)$ and traverses the cut-set $\delta(S)$. Given that graph $G(S)$ is connected, we can assume that path $\mathcal{P}_2$ traverses the cut-set $\delta(S)$ exactly once in each direction through two edges, say e_1 and e_2. We consider three cases:

(1) $e_1, e_2 \in F$. As in Remark 3, we can build a GTSP tour (x^1, y^1) satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses $e = (i, j)$ and all the edges in $F \setminus \{e_2\}$ once and does not traverse e_2. It can be seen that three GTSP tours (x^2, y^3), (x^3, y^3) and (x^4, y^4) as defined above also satisfy $x(F) - x(\delta(S) \setminus F) = |F| - 1$. Note that when we add one traversal of each edge in path $\mathcal{P}_2$, we obtain an GTSP tour that traverses each edge in $F \setminus \{e_1\}$ exactly once and edge e_1 twice.

(2) $e_1, e_2 \notin F$. As in Remark 3, we can build a GTSP tour (x^1, y^1) satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses $e = (i, j)$ and all the edges in $F \setminus \{e_1\}$ once and does not traverse e_2. Again, three tours (x^2, y^3), (x^3, y^3) and (x^4, y^4) as defined above satisfy $x(F) - x(\delta(S) \setminus F) = |F| - 1$. Note that when we add one traversal of each edge in path $\mathcal{P}_2$, we obtain an GTSP tour of that traverses each edge in $F \cup \{e_2\}$ exactly once and the edge e_1 twice.

(3) $e_1 \in F$, $e_2 \notin F$. As in Remark 3, we can build a GTSP tour (x^1, y^1) satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses $e = (i, j)$ and all the edges in $F \setminus \{e_1\}$ once and not traversing e_2. Again, the three GTSP tours (x^2, y^3), (x^3, y^3) and (x^4, y^4) as defined above satisfy $x(F) - x(\delta(S) \setminus F) = |F| - 1$. Note that when we add one traversal of each edge in path $\mathcal{P}_2$, we obtain an GTSP tour that traverses each edge in $F \cup \{e_2\}$ exactly once.

In any of the three cases above, following a similar reasoning to that of the case in which path $\mathcal{P}_2$ is in $G(V \setminus S)$, we obtain $a_e = 0$. The same result can be proved for any edge $e \in E(V \setminus S)$. Hence, we have $a_e = b_e = 0$ for all $e \in E(S) \cup E(V \setminus S)$.

Let $e \in \delta(S)$. There is a GTSP tour satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that does not traverse e. Consider the same tour but adding two traversals of e. By comparing the two tours we obtain that $b_e = 0$, and this holds for all $e \in \delta(S)$.

Consider $e_1, e_2 \in F$. There is a GTSP tour satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses exactly once the edges in $F \setminus \{e_1\}$, and another one that traverses those in $F \setminus \{e_2\}$. By comparing the two tours we obtain that $a_{e_1} = a_{e_2}$ (recall that $a_e = b_e = 0$ for all $e \in E(S) \cup E(V \setminus S)$). Since this is true for any pair of edges $e_1, e_2 \in F$, we have that $a_e = \lambda$, for a given parameter λ, for all $e \in F$.

Consider $e \in \delta(S) \setminus F, e_1 \in F$. There is a GTSP tour satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses exactly once the edges in $F \cup \{e\}$, and another GTSP tour satisfying $x(F) - x(\delta(S) \setminus F) = |F| - 1$ that traverses exactly once the edges in $F \setminus \{e_1\}$. By comparing the two tours we obtain that $a_e + a_{e_1} = 0$ and, hence, $a_e = -\lambda$ for all $e \in \delta(S) \setminus F$.

By substituting all the previously computed coefficients a_e, b_e in inequality $ax + by \le c$ we obtain $\lambda x(F) - \lambda x(\delta(S) \setminus F) \le c$. Given that any of the GTSP tours above satisfies the inequality as an equality, we obtain that $c = \lambda|F| - \lambda$ and inequality $ax + by \le c$ is a scalar multiple of $x(F) - x(\delta(S) \setminus F) \le |F| - 1$. ⧫

Theorem 14. *Parity inequality (29) is facet-inducing of RU-GTSP(G) if graph G is 3-edge connected and graphs G(S) and G(V \ S) are 2-edge connected.*

Proof. It is an immediate consequence of Theorems 7 and 13. ⧫

3.2. p-connectivity inequalities

These constraints are based on those introduced in Corberán et al. (2013) to cut off some fractional solutions in which several variables take value 0.5. Consider a U-GTSP instance with the vertex set V divided into three subsets S_0, S_1 and S_2 as shown in Fig. 5(a), and a fractional solution where $\sum_{k=1}^K x_e^k = 0$ and $\sum_{k=1}^K y_e^k = 0.5$ for the three edges plotted. Note that this solution satisfies all the previously described valid inequalities. Specifically, it satisfies as an equality the connectivity inequalities (2) associated with sets S_0, S_1 and S_2. It can be seen that the following inequality is valid for the U-GTSP and cuts the previous fractional solution:

$$\sum_{k=1}^K \left((x^k + 2y^k)(\delta(S_0))\right) + 2\sum_{k=1}^K \left((x^k + y^k)(S_1 : S_2)\right) \ge 4.$$

Note that the fractional solution $(\bar{x}, \bar{y})$ in Fig. 5(a) is cut by the previous inequality $F(x, y) \ge 4$, since $F(\bar{x}, \bar{y}) = 3 \ngeq 4$.

In general, given a partition $\{S_0, \dots, S_p\}$ of vertex set V (see Fig. 5(b)), we will call

- *p-connectivity inequality* for the GTSP on G to

$$(x + 2y)(\delta(S_0)) + 2\sum_{1 \le r < t \le p} (x + y)(S_r : S_t) \ge 2p, \tag{30}$$

- and the corresponding *p-connectivity inequality* for the U-GTSP on G to

$$\sum_{k=1}^K \left((x^k + 2y^k)(\delta(S_0))\right) + 2\sum_{1 \le r < t \le p} \sum_{k=1}^K \left((x^k + y^k)(S_r : S_t)\right) \ge 2p. \tag{31}$$

These p-connectivity inequalities generalize the double degree constraints presented in Carr et al. (2023).

Theorem 15. *p-connectivity inequalities (30) and (31) are valid for the GTSP and the U-GTSP, respectively.*

Proof. (a) Consider an arbitrarious GTSP tour $(\bar{x}, \bar{y})$ and an upgrade level for each edge it traverses. It can be seen that if $\sum_{k=1}^K (\bar{x}_e^k + \bar{y}_e^k) = 1$ for some edge $e \in (S_r : S_t)$ with $1 \le r < t \le p$, we can define a new partition with p–1 subsets where S_r and S_t have been merged into $S'_r = S_r \cup S_t$ and, if its associated (p–1)-connectivity inequality is satisfied by $(\bar{x}, \bar{y})$, then the original p-connectivity inequality is also satisfied by $(\bar{x}, \bar{y})$. Hence, we can assume that $\sum_{k=1}^K \left((\bar{x}^k + \bar{y}^k)(S_r : S_t)\right) = 0$ for any $1 \le r < t \le p$. In this case, set S_0 has to be directly connected to each one of the sets S_i, $i = 1, \dots, p$. Therefore, $\sum_{k=1}^K \left((\bar{x}^k + 2\bar{y}^k)(S_0 : S_i)\right) \ge 2$ for each $i = 1, \dots, p$ and the inequality holds.

(b) For the U-GTSP the proof is similar to that in Theorem 12. ⧫

It is possible that $|S_i| = 1$ for each $i = 1, \dots, p$. In that case, if we call $S = \cup_{i=1}^p S_i$, the p-connectivity inequality (31) can be written as:

$$\sum_{k=1}^K \left((x^k + 2y^k)(\delta(S))\right) \ge 2\left(|S| - \sum_{k=1}^K \left((x^k + y^k)(E(S))\right)\right). \tag{32}$$

Theorem 16. *p-connectivity inequalities (31) are facet-inducing for U-GTSP(G) if graph G is 3-edge connected, subgraphs $G(S_i)$, $i = 1, \dots, p$, are 3-edge connected, $|(S_0 : S_i)| \ge 2$, $\forall\, i = 1, \dots, p$, and the graph induced by $V \setminus S_0$ is connected.*

Proof. From Theorem 7 it suffices to show that the p-connectivity inequalities (30) for the GTSP on G are facet-inducing of GTSP(G). Let $ax + by \ge c$ be a valid inequality such that

$$\{(x, y) \in \text{GTSP}(G) : \quad (x + 2y)(\delta(S_0)) + 2\sum_{1 \le r < t \le p} (x + y)(S_r : S_t) = 2p\} \subseteq$$
$$\subseteq \{(x, y) \in \text{GTSP}(G) : \quad ax + by = c\}.$$

We have to prove that inequality $ax + by \ge c$ is a scalar multiple of (30). In the GTSP tours used in this proof we will not describe how the edges in each set $E(S_i)$ are traversed. It can be seen that all these tours can be completed by traversing once the edges in certain T-joins and then adding two traversals of the remaining edges in $E(S_i)$, as described in Remark 3 for the parity inequalities.

For each $i = 0, 1, \dots, p$, similar arguments to those used in the proof of Theorem 13, lead to prove that $b_e = 0$, for each $e \in E(S_i)$. Furthermore, using the 3-edge connectivity of graph $G(S_i)$ we obtain that also $a_e = 0$. Hence, we have $a_e = b_e = 0$ for all $e \in E(S_i)$, $i = 0, 1, \dots, p$.

Let S_i and S_j, $i, j \ne 0$ be two sets such that there is an edge $e \in (S_i : S_j)$. Since all the sets $(S_0 : S_k)$ are non-empty, and subgraphs $G(S_k)$ are 3-edge connected, we can build a GTSP tour that traverses twice

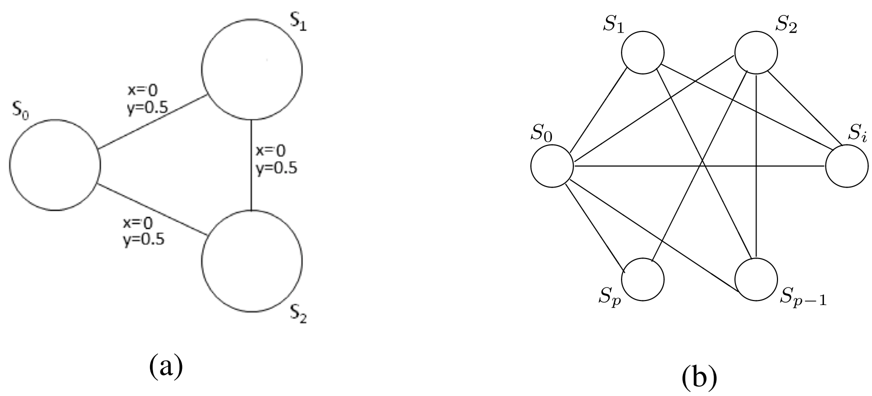

Fig. 5. p-connectivity inequalities.

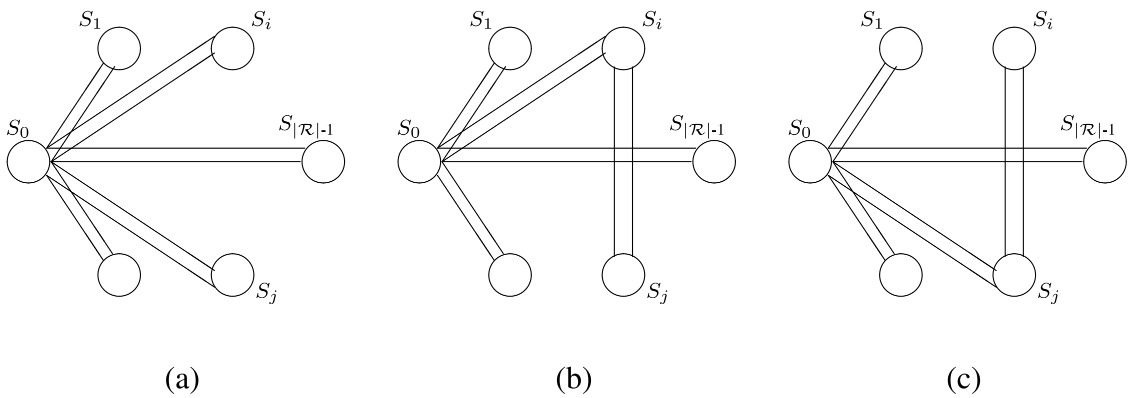

Fig. 6. GTSP tours satisfying (30) with equality.

an edge $f \in (S_0 : S_k)$ and visits all the sets S_k (see Fig. 6(a)). This tour satisfies inequality (30) as an equality. We define two more GTSP tours satisfying (30) with equality such as those depicted in Fig. 6(b) and 6(c). By comparing (a) and (b), we obtain $b_{0j} = b_{ij}$, and by comparing (a) and (c) we obtain $b_{0i} = b_{ij}$, where b_{kl} represents the coefficient of the y variable corresponding to any edge in $(S_k : S_l)$. Given that the graph induced by $V \setminus S_0$ is connected, we can iterate this argument to conclude that $b_e = 2\lambda$ for every edge $e \in (S_i : S_j)$ (including $(S_0 : S_i)$).

For each $i \in \{1, 2, \dots, p\}$, let e_1, e_2 be two edges in $(S_0 : S_i)$ (recall that $|(S_0 : S_i)| \geq 2$ holds). We have already proved that $b_{e_1} = b_{e_2} = 2\lambda$. We can build four GTSP tours satisfying inequality (30) as an equality as follows. One tour traverses e_1 once and does not traverses e_2. Another tour traverses e_2 once and does not traverses e_1. By comparing these tours we obtain $a_{e_1} = a_{e_2}$. The third tour traverses both e_1 and e_2 once, and the fourth one traverses e_1 twice and does not traverses e_2. By comparing them, we obtain $a_{e_1} + a_{e_2} = b_{e_1} = 2\lambda$. Therefore, $a_{e_1} = a_{e_2} = \lambda$, and we have $a_e = \lambda$ and $b_e = 2\lambda$ for each edge $e \in (S_0 : S_i)$, $i = 1, \dots, p$.

As above, let S_i and S_j, $i, j \neq 0$ be two sets such that there is an edge $e = (u, v) \in (S_i : S_j)$. There is a GTSP tour that traverses once edge e, an edge $e_i \in (S_0 : S_i)$, and an edge $e_j \in (S_0 : S_j)$ and satisfies inequality (30) as an equality. If we remove in this tour the traversal of e and add the traversal of the edges in a path joining u and v formed with edges e_i, e_j plus some edges in $G(S_0)$, $G(S_i)$ and $G(S_j)$ (if any of these last edges is traversed three times, two copies would be removed), we obtain another GTSP tour satisfying (30) as an equality. By comparing both tours we obtain $a_{e_i} + a_{e_j} + a_e = b_{e_i} + b_{e_j} = 4\lambda$, which implies $a_e = 2\lambda$ (and recall that $b_e = 2\lambda$). Hence, $a_e = b_e = 2\lambda$, for each edge $e \in (S_i : S_j)$, $i \neq j$.

By substituting all the previously computed coefficients a_e, b_e in inequality $ax + by \geq c$ we obtain

$$(\lambda x + 2\lambda y)(\delta(S_0)) + \sum_{1 \leq r < t \leq p} (2\lambda x + 2\lambda y)(S_r : S_t) \geq c.$$

Given that the GTSP tour in Fig. 6(a), for example, satisfies this inequality with equality, we obtain $2\lambda p = c$ and, hence, inequality $ax + by \geq c$ reduces to

$$(\lambda x + 2\lambda y)(\delta(S_0)) + \sum_{1 \leq r < t \leq p} (2\lambda x + 2\lambda y)(S_r : S_t) \geq 2\lambda p,$$

which is a scalar multiple of (30). ♦

3.3. Cover inequalities

The following inequalities are based on the knapsack cover inequalities (Glover, 1973). Given a subset of edges $F \subseteq E$ and, for each edge $e \in F$, a level k_e such that $\sum_{e \in F} \alpha_e^{k_e} > T$, it is trivial to see that the following *cover inequalities* are valid for the U-GTSP:

$$\sum_{e \in F} \sum_{k=k_e}^{K} (x_e^k + y_e^k) \leq |F| - 1. \tag{33}$$

4. Valid inequalities based on costs

Consider two edges e_1, e_2 and two upgrade levels k_1, k_2. If there are two other upgrade levels for these edges such that the sum of their costs is smaller, and the sum of their $\alpha-$ values is also smaller, no optimal solution will traverse the two edges e_1, e_2 with the upgrade levels k_1 and k_2. This is better expressed in the following theorems.

Theorem 17. *Consider two pairs of variables $x_{e_1}^{k_1}, x_{e_2}^{k_2}$ and $y_{e_1}^{k_1}, y_{e_2}^{k_2}$ corresponding to two edges $e_1, e_2 \in E$ and two upgrade levels $k_1, k_2 \in \{1, \dots, K\}$, and suppose there are two other upgrade levels $k_3, k_4 \in \{1, \dots, K\}$ such that*

$$\alpha_{e_1}^{k_1} + \alpha_{e_2}^{k_2} \geq \alpha_{e_1}^{k_3} + \alpha_{e_2}^{k_4} \quad \textit{(upgrade levels } k_3, k_4 \textit{ are cheaper)}, \tag{34}$$

$$c_{e_1}^{k_1} + c_{e_2}^{k_2} > c_{e_1}^{k_3} + c_{e_2}^{k_4} \quad \textit{(upgrade levels } k_3, k_4 \textit{ have lower traversal cost)}. \tag{35}$$

Then, the variables are incompatible, i.e., any optimal U-GTSP tour satisfies the inequalities:

$$x_{e_1}^{k_1} + x_{e_2}^{k_2} \leq 1, \tag{36}$$

$$y_{e_1}^{k_1} + y_{e_2}^{k_2} \leq 1. \tag{37}$$

Proof. Let us consider a U-GTSP tour that does not satisfy (36), i.e., it satisfies

$$x_{e_1}^{k_1} + x_{e_2}^{k_2} = 2.$$

This means that edge e_1 is upgraded to level k_1 and traversed once ($x_{e_1}^{k_1} = 1, y_{e_1}^{k_1} = 0$) and edge e_2 is upgraded to level k_2 and traversed once ($x_{e_2}^{k_2} = 1, y_{e_2}^{k_2} = 0$). We can construct another U-GTSP tour traversing the same edges but changing the upgrade level of edge e_1 to k_3 and that of edge e_2 to k_4. This new solution maintains the same structure as the previous one (it traverses the same edges the same number of times) and satisfies (3) as (34) implies that the new α-cost is lower. Since the cost of this new solution is lower than that of the original one due to (35), the original solution is not optimal. Let us consider now a solution of the U-GTSP that does not satisfy (37), i.e., it satisfies

$$y_{e_1}^{k_1} + y_{e_2}^{k_2} = 2.$$

This means that edge e_1 is upgraded to level k_1 and traversed twice ($x_{e_1}^{k_1} = 0, y_{e_1}^{k_1} = 1$) and edge e_2 is recovered to level k_2 and traversed twice also. As before, we can construct another feasible U-GTSP tour by changing the upgrade level of edge e_1 to k_3 and that of edge e_2 to k_4. This solution, that satisfies (37), has a lower cost due to (35), so the original solution is not optimal. ◆

Theorem 18. *Consider two variables $y_{e_1}^{k_1}, x_{e_2}^{k_2}$ corresponding to two edges $e_1, e_2 \in E$ and two upgrade levels $k_1, k_2 \in \{1, \dots, K\}$, and suppose there are two upgrade levels $k_3, k_4 \in \{1, \dots, K\}$ such that*

$$\alpha_{e_1}^{k_1} + \alpha_{e_2}^{k_2} \geq \alpha_{e_1}^{k_3} + \alpha_{e_2}^{k_4} \quad \textit{(cheaper)}, \tag{38}$$

$$2c_{e_1}^{k_1} + c_{e_2}^{k_2} > 2c_{e_1}^{k_3} + c_{e_2}^{k_4} \quad \textit{(less cost)}. \tag{39}$$

Then, the two variables are incompatible, i.e., any optimal U-GTSP tour satisfies the inequalities:

$$y_{e_1}^{k_1} + x_{e_2}^{k_2} \leq 1. \tag{40}$$

Proof. As in the proof of Theorem 17, if we have a U-GTSP solution that does not satisfy (40), we can transform it into another one that does with lesser cost by changing the upgrade levels of edges e_1 and e_2 from k_1 and k_2 to k_3 and k_4, respectively. ◆

We will refer to inequalities (36), (37), and (40) as xx cost-based, yy cost-based, and xy cost-based inequalities, respectively. The xx and yy cost-based inequalities can be easily lifted as the following corollary shows. It is not difficult to see that xy cost-based inequalities can also be lifted in a similar way.

Corollary 1. *Let $e_1, e_2 \in E$ be two edges. Let $S, T \subset \{1, \dots, K\}$ two sets of upgrade levels such that, for each pair $k_1 \in S, k_2 \in T$, there is another pair $k_3, k_4 \in \{1, \dots, K\}$ satisfying* (34) and (35). *Then, any optimal solution of the U-GTSP satisfies the inequalities:*

$$\sum_{k \in S} x_{e_1}^{k} + \sum_{k \in T} x_{e_2}^{k} \leq 1, \tag{41}$$

$$\sum_{k \in S} y_{e_1}^{k} + \sum_{k \in T} y_{e_2}^{k} \leq 1. \tag{42}$$

Note that xx and yy cost-based inequalities, which involve two edges, can be extended to sets of three edges as the following theorem shows:

Theorem 19. *Consider three edges $e_1, e_2, e_3 \in E$ and three upgrade levels k_1, k_2, k_3. Suppose that there are other three upgrade levels $k_4, k_5, k_6 \in \{1, \dots, K\}$ such that*

$$\alpha_{e_1}^{k_1} + \alpha_{e_2}^{k_2} + \alpha_{e_3}^{k_3} \geq \alpha_{e_1}^{k_4} + \alpha_{e_2}^{k_5} + \alpha_{e_3}^{k_6} \quad (\textit{upgrade levels } k_4, k_5, k_6 \textit{are cheaper}) \textit{ and} \tag{43}$$

$$c_{e_1}^{k_1} + c_{e_2}^{k_2} + c_{e_3}^{k_3} > c_{e_1}^{k_4} + c_{e_2}^{k_5} + c_{e_3}^{k_6} \quad (\textit{upgrade levels } k_4, k_5, k_6 \textit{ have less cost}). \tag{44}$$

Then, any optimal U-GTSP solution satisfies the inequalities:

$$x_{e_1}^{k_1} + x_{e_2}^{k_2} + x_{e_3}^{k_3} \leq 2, \tag{45}$$

$$y_{e_1}^{k_1} + y_{e_2}^{k_2} + y_{e_3}^{k_3} \leq 2. \tag{46}$$

5. Preprocessing

In this section we will study if, for a given instance, some variables x_e^k and y_e^k can be dropped from the model or fixed to zero in order to reduce its size.

Let $G = (V, E)$ be a connected undirected graph with set of vertices V and set of edges E, and two costs c_e, α_e associated with each edge. The cost-constrained GTSP (CC-GTSP) consists of finding a minimum cost tour (computed in terms of costs c_e) visiting all the vertices at least once with an α-cost less than or equal to a certain amount T. For every edge $e \in E$, let variable x_e be equal to 1 if edge e is traversed exactly once and 0 otherwise, and variable y_e equal to 1 if e is traversed exactly twice and 0 otherwise. The CC-GTSP in G can be formulated as follows:

$$\text{(CC-GTSP(G))} \qquad \text{Minimize} \quad \sum_{e \in E} \left(c_e x_e + 2c_e y_e\right)$$

s.t.

$$(x + 2y)\left(\delta(i)\right) \equiv 0 \pmod 2,\ \forall\, i \in V, \tag{47}$$

$$(x + 2y)\big(\delta(S)\big) \geq 2,\ \forall\ S \subset V \tag{48}$$

$$\sum_{e \in E} \left(\alpha_e x_e + \alpha_e y_e\right) \leq T, \tag{49}$$

$$x_e + y_e \leq 1,\ \forall\ e \in E \tag{50}$$

$$x_e, y_e \in \{0, 1\},\ \forall e \in E \tag{51}$$

The CC-GTSP is a graphical relaxation of the cost-constrained TSP (Sokkappa, 1990) or the Resource-Constrained TSP (Pekny and Miller, 1990). The CC-GTSP in a graph G is a particular case of the U-GTSP in G with $K = 1$ upgrade levels. Interestingly, the U-GTSP with K levels in a graph G can be solved as a CC-GTSP in a multigraph $\hat{G}$ that has K parallel copies of each edge of G, as shown in the following theorem.

Theorem 20. *The optimal solution of any U-GTSP can be obtained by solving a CC-GTSP on a multigraph.*

Proof. Let us consider the U-GTSP defined on a graph $G = (V, E)$, where each edge $e \in E$ has K possible levels, and for each level an associated cost c_e^k and price α_e^k. Then let $\hat{G} = (V, \hat{E})$ be the augmented graph of G such that for each edge $e \in E$ we have K edges $e^k \in \hat{E}$ ($|\hat{E}| = |E| \times K$). For each edge $e^k \in \hat{E}$ we define the costs $c_{e^k} = c_e^k$ and $\alpha_{e^k} = \alpha_e^k$. The CC-GTSP in $\hat{G}$ would be expressed as follows:

$$\text{(CC-GTSP}(\hat{G})) \qquad \text{Minimize} \quad \sum_{e^k \in \hat{E}} \left(c_{e^k} x_{e^k} + 2c_{e^k} y_{e^k}\right)$$

s.t.

$$(x + 2y)\left(\delta(i)\right) \equiv 0 \pmod 2,\ \forall\, i \in V, \tag{52}$$

$$(x + 2y)\big(\delta(S)\big) \geq 2,\ \forall\ S \subset V \tag{53}$$

$$\sum_{e^k \in \hat{E}} \left(\alpha_{e^k} x_{e^k} + \alpha_{e^k} y_{e^k}\right) \leq T, \tag{54}$$

$$x_{e^k} + y_{e^k} \leq 1,\ \forall\ e^k \in \hat{E} \tag{55}$$

$$x_{e^k}, y_{e^k} \in \{0,1\},\ \forall e^k \in \hat{E} \tag{56}$$

Note that, if we identify variables x_e^k of the U-GTSP formulation with variables x_{e^k} of the CC-GTSP one, both formulations are the same except for inequalities (4) and (55), since constraints (55) do not have a summation over the values of k. It is obvious that any feasible solution of the U-GTSP is feasible for the CC-GTSP, since inequality (4) dominates (55).

Let us suppose there is a feasible solution S of $CC-GTSP(\hat{G})$ with an edge f such that $\left(x_{f^k}+y_{f^k}\right) \le 1$ for every $k \in \{1,\dots,K\}$ but $\sum_{k=1}^{K}\left(x_{f^k}+y_{f^k}\right) \ge 1$. Let $\hat{k}$ be the maximum value in $1,\dots,K$ such that $x_{f^{\hat{k}}}+y_{f^{\hat{k}}}=1$. Let $\hat{S}$ be the CC-GTSP solution we get from S such that $\hat{x}_{e^k}=x_{e^k}$, $\hat{y}_{e^k}=y_{e^k}$ for all $e \in E \setminus f$, $\hat{x}_{f^k}=0$ and $\hat{y}_{f^k}=0$ except:

- $\hat{y}_{f^{\hat{k}}}=1$, if $\sum_{k=1}^{K}\left(x_{f^k}+2y_{f^k}\right) \equiv 0 \pmod 2$,
- $\hat{x}_{f^{\hat{k}}}=1$, if $\sum_{k=1}^{K}\left(x_{f^k}+2y_{f^k}\right) \equiv 1 \pmod 2$.

$\hat{S}$ satisfies $\sum_{k=1}^{K}\left(\hat{x}_{f^k}+\hat{y}_{f^k}\right) \le 1$ and is also feasible for both CC-GTSP($\hat{G}$) and U-GTSP(G) since:

- $(\hat{x}+2\hat{y})\left(\delta(i)\right) \equiv 0 \pmod 2$ for all $i \in V$ given that $\sum_{k=1}^{K}\left(x_{f^k}+2y_{f^k}\right) \equiv \hat{x}_{f^{\hat{k}}}+2\hat{y}_{f^{\hat{k}}} \pmod 2$,
- $\sum_{k=1}^{K}\left(\hat{x}_{e^k}+\hat{y}_{e^k}\right) \le 1$,
- $(\hat{x}+2\hat{y})\left(\delta(W)\right) \ge 2$ for all $W \subset V$ given that $\sum_{k=1}^{K}\left(x_{f^k}+2y_{f^k}\right) \equiv \hat{x}_{f^{\hat{k}}}+2\hat{y}_{f^{\hat{k}}} \pmod 2$ and the solution remains connected,
- $\sum_{k=1}^{K}\sum_{e\in E}\left(\alpha_e^k\hat{x}_{e^k}+\alpha_e^k\hat{y}_{e^k}\right) \le T$.

Moreover, clearly $\hat{S}$ also has a lower objective function value, so solution S is not optimal. Therefore, the optimal solution of the CC-GTSP is an optimal solution of the U-GTSP. ♦

Theorem 21. *Let $e=(u,v) \in E$ be an edge of graph G and k an upgrade level. If there is a path $\mathcal{P}=\{\hat{e}_1,\dots,\hat{e}_{|\mathcal{P}|}\}$ in $\hat{G}$ from u to v and a set of upgrade levels $\{k_1,k_2,\dots,k_{|\mathcal{P}|}\}$ for the edges in $\mathcal{P}$ such that $\sum_{i=1}^{|\mathcal{P}|}c_{\hat{e}_i}^{k_i} \le c_e^k$ and $\sum_{i=1}^{|\mathcal{P}|}\alpha_{\hat{e}_i}^{k_i} \le \alpha_e^k$, then there is an optimal solution of the U-GTSP in G satisfying $x_e^k=0$ and $y_e^k=0$.*

Proof. If a U-GTSP solution uses edge $e=(u,v)$ with level k, we can replace it with the edges of the path $\{e_1,\dots,e_{|\mathcal{P}|}\}$ with levels $k_1,\dots,k_{|\mathcal{P}|}$. The new solution will be feasible (it has less α-cost) and better (it has less c-cost). ♦

Theorem 21 allows to set the variables x_e^k and y_e^k to zero in the formulation of the U-GTSP, if a path from u to v with both cheaper costs is found in the corresponding CC-GTSP. One way to try to find cheaper paths from u to v consists of finding a constrained shortest path on $\bar{G}$ from u to v. In general, the term *Constrained Shortest Path Problem (C-SPP)* refers to the problem of finding a shortest path but establishing an upper limit on the sum of another edge cost (Handler et al., 1980). Although the Shortest Path Problem can be solved optimally in polynomial time (Deo and Pang, 1984), unfortunately the C-SPP is NP-hard (Hartmanis, 1982). However, in Lozano and Medaglia (2013) an exact method for the C-SPP capable of handling large-scale networks in a reasonable amount of time is proposed. This method is based on a Dynamic Programming algorithm called *Pulse Algorithm* .

In Algorithm 1 we have adapted the *Pulse Algorithm* in order to get the negligible levels of an edge $e=(u,v)$ according to Theorem 21. Given an edge $e=(u,v)$ and a level k, Algorithm 1 returns true if there exists some path from u to v in $\bar{G}$ with both c-cost and α-cost lower than c_e^k and α_e^k, respectively. In this case, x_e^k and y_e^k can be fixed to zero.

As we are not searching for the shortest path but any path that can replace e, the algorithm is not programmed in a recursive way, which allows to finalize at the moment a shorter path is found. Initially, both costs of any vertex are $+\infty$ since no vertex has been visited. $\mathcal{A}$ is a list of items (each item is a 3-tuple formed by a node that has already been visited, a c-cost and an α-cost) that have yet to be explored. A node v could be in several items of $\mathcal{A}$ but with different costs, meaning that there are different paths to get to node v from u. The algorithm starts by introducing in $\mathcal{A}$ the source node u with zero costs, $c_1=0$ and $\alpha_1=0$.

Algorithm 1:

Input : $e=(u,v)$, k
Output: *true* if there is a path from u to v in $\bar{G}$ with total costs respectively lower than c_e^k and α_e^k
false in else case
r = false
foreach $i \in V$ **do**
  $d_c(i)=+\infty$
  $d_\alpha(i)=+\infty$
$\mathcal{A}=\{(v_1=u, c_1=0, \alpha_1=0)\}$
while $\mathcal{A} \ne \emptyset$ ***and not*** r **do**
  $(u', c', \alpha')=\mathcal{A}_{|\mathcal{A}|}$
  $\mathcal{A}=\mathcal{A} \setminus \mathcal{A}_{|\mathcal{A}|}$
  foreach $\bar{e}=(u',v') : v',\ \bar{e} \ne e$ **do**
    foreach $k'=1,..,K$ **do**
      if $c'+c_{\bar{e}^{k'}} \le c_e^k$ ***and*** $\alpha'+\alpha_{\bar{e}^{k'}} \le \alpha_e^k$ **then**
        if $v'=v$ **then**
          r = true
        else if $c'+c_{\bar{e}^{k'}} \le d_c(v')$ ***or*** $\alpha'+\alpha_{\bar{e}^{k'}} \le d_\alpha(v')$ **then**
          $\mathcal{A}=\mathcal{A} \cup (v', \alpha'+\alpha_{\bar{e}^{k'}}, c'+c_{\bar{e}^{k'}})$
          if $c'+c_{\bar{e}^{k'}} < d_c(v')$ ***and*** $\alpha'+\alpha_{\bar{e}^{k'}} < d_\alpha(v')$ **then**
            $d_c(v')=c'+c_{\bar{e}^{k'}}$
            $d_\alpha(v')=\alpha'+\alpha_{\bar{e}^{k'}}$
return r

In the main loop, the last item of $\mathcal{A}$ (containing the last visited node u') is explored and removed from the list. Basically, we get to each node v' in the neighborhood of u' by exploring all the edges parallel to $\bar{e}=(u',v')$. Those partial paths that are feasible, that is, with both updated costs $c'+c_{\bar{e}^{k'}} \le c_e^k$ and $\alpha'+\alpha_{\bar{e}^{k'}} \le \alpha_e^k$, are inserted in $\mathcal{A}$ as an item $(v', c'+c_{\bar{e}^{k'}}, \alpha'+\alpha_{\bar{e}^{k'}})$. If at any point we reach node v, the algorithm ends and returns *true*. The algorithm can also terminate because $\mathcal{A}$ is empty, which means that there is no path with costs respectively equal to or lower than c_e^k and α_e^k, so *false* will by returned.

In order to save computational time, the *Pulse Algorithm* also avoids exploring the partial paths that are dominated by others already explored. For this purpose, a list of non-dominated costs associated with each node is saved. Depending on the instances, this list can be very large, thus reduction strategies are proposed, but the optimal solution would also be reached in case of exploring the dominated paths too. In our case, we have reduced the list of dominated costs to one item per node. If a node is reached and both costs are better than the previous ones, then its dominating costs d_c and d_α will be updated. This means that a node will not be explored unless one of its costs is better than its saved equivalent.

6. A branch-and-cut algorithm

In this section, we present a branch-and-cut algorithm for the U-GTSP. First we will introduce the linear relaxation solved at the root node. Then we will describe the separation algorithms that are part of the cutting-plane procedure applied at each node of the tree. We will also propose a primal heuristic to obtain good upper bounds from the fractional solutions of the linear relaxations and a branching strategy.

6.1. Initial linear relaxation

In the linear relaxation solved at the root node of the branch-and-cut tree, we include constraints (4), connectivity constraints (2) only for sets S with one single vertex (to ensure that all the vertices are visited) and the budget constraint (3). We also consider the p-connectivity inequalities with $p = 2$ and $|k_1| = |k_3| = 1$, i.e., inequalities (32) with $|S| = 2$, since there are only $|E|$ of them and they help increase the lower bound. Parity inequalities (1) are relaxed, as well as the integrality condition from (5).

Let us call $(\bar{x}, \bar{y})$ the solution of the linear relaxation of the U-GTSP solved at one of the nodes of the branch-and-cut algorithm. Then, we will define two aggregated values, $x_e^{agg} = \sum_{k=1}^{K} \bar{x}_e^k$ and $y_e^{agg} = \sum_{k=1}^{K} \bar{y}_e^k$.

6.2. Separation algorithms

In this section we describe the separation algorithms that have been applied to identify the following families of inequalities that are violated by the LP solution at any iteration of the cutting-plane algorithm: connectivity and parity inequalities (12) and (29), p-connectivity inequalities (2), cover inequalities (33), xx cost-based inequalities (36), yy cost-based inequalities (37), and xy cost-based inequalities (40).

6.2.1. Connectivity inequalities

To separate violated connectivity inequalities, we use a heuristic algorithm that is based on computing the connected components of the graph induced by the edges e such that $x_e^{agg} \geq \epsilon$, where ϵ is a given parameter. For each connected component, we check the corresponding connectivity inequality for violation. We try $\epsilon = 0, 0.25, 0.5, 0.75$, but a given value is tried only when the previous one did not succeed in finding a violated inequality. If the solution of the linear relaxation is integer, this procedure applied with $\epsilon = 0$ guarantees that, if the solution is not connected, it will find at least one violated connectivity inequality.

6.2.2. Parity inequalities

In order to find cutsets that may correspond to a violated parity inequality (29), we use the idea that an edge e for which the fractional part of x_e^{agg} is close to 0.5 should not appear in the cutset, since they would either decrease the value of the right-hand side of the inequality (if the edge is assigned to set F) or increase the value of the left-hand side (if the edge is not assigned to F).

Given an edge e, we denote with $\{x_e^{agg}\}$ the fractional part of x_e^{agg}. Then, we sort all the edges according to the value $\left|\{x_e^{agg}\} - 0.5\right|$ in increasing order. In that way, the first edges of this ordering will be those that we do not want to appear in the cutset. We choose the first edge $e = (i,j)$ of the list and initialize $S = \{i,j\}$. Then, we choose $F \subseteq \delta(S)$ according to the procedure described in Campbell et al. (2021) and based on the one proposed in Ghiani and Laporte (2000) and check if the associated parity inequality is violated. If it is violated, we add the cut, mark all the edges in $E(S)$ as used, and start the process again by choosing from the list another initial edge for S that is not marked as used yet. If the parity inequality is not violated, then we add one more vertex to S by choosing among the vertices adjacent to those in S the one that improves the violation of the inequality the most, i.e., it maximizes the difference between the increase of the right-hand side minus the increase of the left-hand side of the inequality. We continue adding vertices to S until the inequality is violated or no more vertices can be added.

6.2.3. p-connectivity inequalities

In p-connectivity inequalities (31), the set of vertices V is partitioned into $p+1$ subsets $S_0, S_1, \ldots, S_p$. We have designed a heuristic separation algorithm to find violated p-connectivity inequalities with $|S_i| = 1$ for $i = 1, \ldots, p$, i.e. inequalities (32).

For each edge $e = (i,j) \in E$ such that $0 < x_e^{agg} \leq 0.9$, we initialize $S = \{i,j\}$. We construct a candidate list of vertices C containing all the vertices in $V \setminus S$ that are adjacent to a vertex in S. Now, for each vertex $v \in C$, we evaluate the p-connectivity inequality associated with the vertex set resulting from incorporating v to S. If we find a vertex such that the corresponding p-connectivity inequality is violated, we add it to the LP and start the separation algorithm again with a different edge. Otherwise, we choose the candidate vertex for which the resulting inequality is closest to being violated and add it to S. Then, we update the list of candidate vertices C and continue evaluating them and increasing the size of S until a violated p-connectivity inequality is found or $C = \emptyset$.

Table 1
Results obtained with the different configurations of the cutting-plane procedure.

Conf.	# opt	LB	UB	Gap	Nodes opt	Nodes no opt	Time
Base	7	40125.62	47214.33	8.52	1983.7	10607.7	184.1
C1	7	40114.32	47439.44	9.96	1063.1	6979.9	303.8
C2	7	40115.23	47701.89	10.25	977.3	1397.1	344.8
C3	7	**40131.48**	**46773.22**	**6.92**	1267.3	9210.6	**153.8**
C4	**8**	40128.22	47549.89	9.47	1505.1	7860.2	207.0

6.2.4. Cover inequalities

To find violated cover inequalities (33), we first initialize F as the set of all the edges such that $\sum_{k=1}^{K}(x_e^k + y_e^k) \geq 0.9$. For each edge $e \in F$, we set k_e as the first level of edge $e \in F$ such that $\sum_{k=1}^{K}(x_e^k + y_e^k) > 0$, and for each edge $e \notin F$, $k_e = 1$. If the cover inequality associated with this set F is not violated, that is $\sum_{e\in F}\sum_{k=k_e}^{K}(x_e^k + y_e^k) \leq |F| - 1$, we stop and no new inequality is introduced in the LP. Note that the inequality can be violated but not valid yet, since $\sum_{e\in F}\alpha_e^{k_e} > T$ must be satisfied. If the cover inequality is violated, let us define t_e as

$$t_e = \begin{cases} \dfrac{\alpha_e^{\bar{k}_e} - \alpha_e^{k_e}}{x_e^{k_e} + y_e^{k_e}} & \text{if } e \in F \\[2ex] \dfrac{\alpha_e^{\bar{k}_e}}{1 - \sum_{k=1}^{K}(x_e^k + y_e^k)} & \text{if } e \notin F \end{cases},$$

where $\bar{k}_e$ is the first upgrade level of edge e greater than k_e for which $x_e^k + y_e^k > 0$ (if there is no such level, $t_e = 0$). The value of t_e is the ratio between the increase of the α-cost and the loss of violation in the cover inequality when we increase the chosen level for edge e to $\bar{k}_e$ (which also implies adding e to F if $e \notin F$).

Now we choose the edge $e \in E$ that maximizes t_e while ensuring that the cover inequality is still violated when changing level k_e to $\bar{k}_e$ and, if $e \notin E$, adding e to F. Then, we set $k_e = \bar{k}_e$, add e to F if it was not in F yet, and update t_e. We continue choosing new edges while the inequality is violated. When no more edges with $t_e > 0$ can be chosen, we check if $\sum_{e\in F}\alpha_e^{k_e} > T$. If so, the associated cover inequality is valid and violated, so we added it to the LP.

6.2.5. Cost-based inequalities

Violated cost-based inequalities (36), (40), and (37) are separated by means of an exhaustive search for all pairs of edges e_1, e_2 such that $x_{e_1}^{agg} + x_{e_2}^{agg} > 1$, $y_{e_1}^{agg} + y_{e_2}^{agg} > 1$, and $y_{e_1}^{agg} + x_{e_2}^{agg} > 1$, respectively, since the time needed for such search is small.

6.3. Primal heuristic

To obtain good upper bounds for the branch-and-cut algorithm, we have designed a matheuristic algorithm that is applied to some of the fractional solutions obtained when solving the linear relaxations along the search tree. Given the optimal solution of the linear relaxation solved at the current node of the search tree, if vectors x^{agg} and y^{agg} are integer, we check if the route defined by these values satisfies connectivity and parity. If it does, then we have a feasible solution of the GTSP defined on G, and we can obtain a feasible solution of the U-GTSP by solving a Multi-Choice Knapsack Problem as defined in Remark 1.

6.4. Branching strategy

We have tried and compared two different branching strategies. The first one consists of applying the strong branching strategy (Applegate et al., 1995) implementation of Cplex to branch on the variables of the formulation. In the second strategy, we proceeded as follows.

Given a fractional solution, we select the edge $e \in E$ for which $x_e^{agg} - \lfloor x_e^{agg} \rfloor$ or $y_e^{agg} - \lfloor y_e^{agg} \rfloor$ is maximal, and we branch by adding the constraints $\sum_{k=1}^{K} x_e^k = 0$ and $\sum_{k=1}^{K} x_e^k = 1$ (or with the y_e^k variables). If no such fractional values are found for any edge e, we let Cplex choose a variable for branching using the strong branching strategy.

The idea behind this strategy is that, in that way, we would reach solutions where x_e^{agg} and y_e^{agg} are integer sooner, and then we would be able to apply the primal heuristic earlier in the tree and more often to obtain upper bounds. However, in our initial computational tests the results obtained with this second strategy performed considerably worse than just using strong branching. We suspect that the reason was that the strong branching strategy is very powerful compared with a simple rule based on just selecting the most fractional value. Therefore, we decided to try adding the following new variables to the formulation

$$z_e^1 = \sum_{k=1}^{K} x_e^k, \qquad z_e^2 = \sum_{k=1}^{K} y_e^k,$$

and use the strong branching strategy giving higher priority to these new variables. In this way, we achieve the same effect as when branching with the previous constraints without losing the benefits of strong branching.

7. Computational results

We have generated U-GTSP instances from some undirected graphs taken from the web page www.uv.es/plani/instancias.htm Although these are instances defined for different arc routing problems, we have considered only the underlying undirected graphs. From each undirected graph $G = (V, E)$, with a cost c_e for each edge $e \in E$, we have generated a U-GTSP instance. We have considered $K = 4$ upgrade levels and, for each $e \in E$, we have generated the values for the costs c_e^k and the prices α_e^k, $k = 1, 2, 3, 4$ as follows:

$$c_e^4 = c_e(\text{the original cost}), \quad c_e^3 = 2c_e, \quad c_e^2 = 3c_e, \quad \text{and} \quad c_e^1 = 4c_e,$$

$$\alpha_e^4 = 3c_e r_e, \quad \alpha_e^3 = 2c_e r_e, \quad \alpha_e^2 = c_e r_e, \quad \text{and} \quad \alpha_e^1 = 0,$$

where r_e is a random real number between 1 and 3. We select a different number r_e for each $e \in E$ in order to avoid a direct relationship between the cost of traversing an edge that has been upgraded to a level k and the price of upgrading the edge to this level k. The total available budget T is calculated as the average value of α_e^3 for all the edges $e \in E$, multiplied by the number of vertices in the graph, $|V|$. Note that this value for T is an estimation of the budget for a GTSP tour (which has approximately $|V|$ edges) in the case that all the edges in the tour have been upgraded to level 3.

The sets of undirected graphs taken from www.uv.es/plani/instancias.htm are:

- Graphs "Albaida" ($|V| = 116$ and $|E| = 174$) and "Madrigueras" ($|V| = 196$ and $|E| = 316$) graphs, from the corresponding Rural Postman Problem instances.
- 6 GTSP instances, with $150 \le |V| \le 225$ and $296 \le |E| \le 433$, obtained from 7 planar Euclidean TSP instances from TSPLIB.
- 12 graphs from the Rural Postman Problem instances called UR5* with $298 \le |V| \le 499$ and $597 \le |E| \le 1526$, one from the UR7* instances ($|V| = 749$, $|E| = 2314$) and one from the UR1* ones ($|V| = 1000$, $|E| = 3083$).

We also tried 24 smaller instances obtained from the well-known Christofides Rural Postman Problem ones, but they were all optimally solved in less than 1 s, so we are not reporting the results here.

All the algorithms have been coded in C++ using Cplex 12.1 Concert Technology with Cplex cuts turned off. The computational experiments have been executed on a i7-9700F CPU at 3 GHz with a time limit of one hour.

In order to assess the contribution of the different families of cuts, we have run the 22 instances with different configurations of the cutting-plane procedure. The *base* configuration uses the separation algorithms for connectivity, parity, and p-connectivity inequalities, since some preliminary experiments had shown that they were all fundamental for the success of the branch-and-cut algorithm and removing any of them produces much worse results. The following configurations were also tried:

- C1: Base configuration and xx and yy cost-based inequalities.
- C2: Configuration C1 and cover inequalities.
- C3: Base Configuration with xx and yy cost-based inequalities introduced as local cuts.
- C4: Configuration C3 with xy cost-based inequalities introduced as local cuts.

The reason for using cost-based inequalities as local cuts in configurations C3 and C4 is that the experiments with configuration C1 showed that none of these cuts were found at the root node, but a huge number of such violated inequalities was obtained once we started to branch. We thought that the appearance of these violated cuts was due to the fact that when branching some variables are fixed to 1. But if this was true, these cuts found would probably be useless in the other branch of the tree in which this same variable is fixed to 0, so introducing them in the linear relaxations as local cuts, i.e., cuts that are only used at this node and the ones that stem from it, could reduce the size of the linear problems solved and, thus, the time needed to solve them.

The results obtained with these configurations are summarized in Table 1. The first column shows the configuration name. Column '# opt' gives the total number of instances that were solved optimally. The next two columns, labeled 'LB' and 'UB', report the average values of the lower bound at the end of the execution, and the final upper bound, respectively. The 'Gap' column gives the average percentage gap between the final lower and upper bounds for those instances that could not be solved optimally and for which an upper bound was obtained with all the configurations (this amounts to a total of nine instances). Column 'Nodes opt' and 'Nodes no opt' show the average number of branch-and-cut nodes explored for the instances that were solved optimally and for those that were not, respectively, while column 'Time' presents the average computing time, in seconds, for the seven instances that were solved optimally by all the configurations.

As can be seen in Table 1 , introducing cost-based inequalities as global cuts in configuration C1 resulted in worse results both in terms of average gap and time. Interestingly, the number of nodes explored reduced drastically in this configuration both for the solved and the unsolved instances. This seemed to imply that these inequalities were useful to reduce the number of nodes needed to find the optimal solution, but that the time needed to solve each linear relaxation was greater due to the huge number of cost-based inequalities added. The addition of cover inequalities in configuration C2 caused the gap and solution time to be even worse. However, changing cost-based inequalities from global to local in C3 produced the best results in terms of gap and computing time. Unfortunately, the further addition of xy cost-based inequalities did produce worse gaps and times, although one more optimal solution could be found. Since configuration C3 was the one producing the best results, we decided to use it for the following computational experiments.

In Table 2 we present the detailed results for all the 22 instances obtained with the branch-and-cut algorithm using the strong branching strategy implemented in Cplex and no heuristic procedure (called

Table 2
Effect of the primal heuristic and the new branching strategy.

Instance	Optimal?		LB		UB		Time		Gap	
	B&C1	B&C2	B&C1	B&C2	B&C1	B&C2	B&C1	B&C2	B&C1	B&C2
Albaida	Yes	Yes	14160	14160	14160	14160	80	20	0.00	0.00
Madrigueras	Yes	Yes	22185	22185	22185	22185	436	81	0.00	0.00
rat195 g	Yes	Yes	3962	3962	3962	3962	32	4	0.00	0.00
kroA150 g	Yes	Yes	46589	46589	46589	46589	314	54	0.00	0.00
pr152 g	Yes	Yes	150183	150183	150183	150183	48	49	0.00	0.00
ts225 g	No	No	222466	221370	222589	222589	3600	3600	0.06	0.55
kroB200 g	Yes	Yes	51137	51137	51137	51137	34	43	0.00	0.00
kroA200 g	Yes	Yes	49674	49674	49674	49674	131	40	0.00	0.00
UR532	No	Yes	18366	18367	18368	18367	3600	829	0.01	0.00
UR535	No	No	28675	28870	30340	29505	3600	3600	5.80	2.20
UR537	No	No	31370	31671	34702	32656	3600	3600	10.62	3.11
UR542	No	No	14886	14911	15087	15011	3600	3600	1.34	0.67
UR545	No	No	24012	24040	26970	31668	3600	3600	12.32	31.73
UR547	No	No	26358	26416	29984	29282	3600	3600	13.75	10.85
UR552	No	No	16325	16392	18348	16487	3600	3600	12.39	0.58
UR555	No	No	23174	23070	24571	–	3600	3600	6.03	–
UR557	No	No	23248	23319	–	–	3600	3600	–	–
UR562	No	No	17690	17719	18523	17936	3600	3600	4.71	1.22
UR565	No	No	21191	21134	24173	24440	3600	3600	14.07	15.64
UR567	No	No	20914	20886	–	–	3600	3600	–	–
UR767	No	No	26560	26434	–	–	3600	3600	–	–
UR167	No	No	29761	29755	–	–	3600	3600	–	–

'B&C1') versus the results obtained using the branching strategy described in Section 6.4 together with the primal heuristic presented in Section 6.3 (called 'B&C2'). The first column corresponds to the name of the instance. The columns labeled 'Optimal?' report if the optimal solution has been found or not for each instance. Columns 'LB' and 'UB' show the final lower and upper bounds, while columns 'Time' and 'Gap' present the computation time, in seconds, and the percentage gap between the lower and upper bounds. When no upper bound has been found within the time limit, the symbol '-' is reported in the 'UB' and 'Gap' columns.

From the results in Table 2, it can be seen that one more instance has been solved by the algorithm B&C2. This version of the algorithm including the new branching strategy and the primal heuristic obtains a better lower bound in 9 instances, while B&C1 outperforms it in 6. The upper bound obtained with B&C2 is better on 7 of the 14 instances that were not previously solved. Strangely, B&C1 obtains a better upper bound in two of these instances, while there is an instance in which B&C2 is not able to obtain a feasible solution but B&C1 does. Regarding the gap, B&C2 outperforms B&C1 in 7 instances, while the opposite happens in 3 instances. The average gap for those instance for which an upper bound is obtained with both procedures is 0.5% better with B&C2. Finally, the average computing time for the instances solved by both algorithms is 154 s with B&C1 and 42 with B&C2. So we can conclude that version of the algorithm including the primal heuristic and the new branching strategy clearly performs better than the previous one. We also have tried running both algorithms increasing the time limit to two hours. While algorithm B&C1 could not solve any further instance optimally, B&C2 was able to find two new optimal solutions, namely U542 and U552.

To study the effect of the number of upgrade levels on the difficulty of the instances, we have applied algorithm B&C2 to the same instances using only the first two and the first three levels. When solving the instance with two levels, the total budget has been reduced to 25% of the original one, while in the case of three levels it is 50% of the original budget. The results are summarized in Table 3. The first column gives the number of levels used. Column 'Optimal' reports the number of instances (out of 22) optimally solved. 'No UB' presents the number of instances for which no feasible solution was found within the time limit, and 'Gap' shows the average gap between the final upper and lower bounds (for those instances for which a feasible solution has been found). Column 'Time' reports the average computing time, in seconds, for those instances that have been solved to optimality. It can be seen that the average gap, as well as the number of instances for which no feasible solution is found, diminish when the number of levels is smaller. The number of optimal solutions found increases when we move from four to three levels, although it stays the same when the number of levels diminishes to 2.

Table 3
Effect of the number of levels on the difficulty of the instances.

Levels	Optimal	No UB	Gap	Time
2	11	0	0.81	471.21
3	11	2	3.12	552.42
4	8	5	3.91	140.44

8. Conclusions

In this work, we have proposed a framework for selecting upgrade levels for the edges in a non-complete graph with the goal of obtaining routes in the graph traversing all the nodes with the minimum possible total costs and within a budget threshold.

We have developed a novel formulation. For this formulation, we have analyzed the polyhedron of solutions and proposed several valid inequalities. For the branch-and-cut algorithm we have proposed, we have designed a math-heuristic for improving the upper bounds, which exploits the structure of a related multi-choice knapsack problem. Moreover, we have proved that any U-GTSP can be expressed as a cost-constrained GTSP and we take advantage of this property for stating effective preprocessing rules.

We have conducted a numerical study to test the proposed branch and cut and to get insights about the designed framework that has revealed the good performance of most of the valid inequalities, as well as the great effect of the primal heuristic and the new branching strategy.

The study of the upgrading of edges in other routing problems as well as the development of clever math-heuristic procedures for obtaining quality solutions for the U-GTSP within reasonable times will be the topic of forthcoming works.

CRediT authorship contribution statement

Mercedes Landete: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Isaac Plana:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **José Luis Sainz-Pardo:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **José María Sanchis:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing.

Data availability

The data instances used in the computational experiments and the best solutions obtained can be found at www.uv.es/plani/instancias.htm.

Acknowledgments

This work was supported by Generalitat Valenciana, Spain through project CIGE/2021/161 and by grant PID2021-122344NB-I00 funded by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe".

References

Afrashteh, E., Alizadeh, B., Baroughi, F., 2020. Optimal approaches for upgrading selective obnoxious p-median location problems on tree networks. Ann. Oper. Res. 289, 153–172.
Alvarez-Miranda, E., Sinnl, M., 2017. Lagrangian and branch-and-cut approaches for upgrading spanning tree problems. Comput. Oper. Res. 83, 13–27.
Applegate, F., Bixby, R., Chvátal, V., Cook, W., 1995. Finding Cuts in the TSP (a Preliminary Report). DIMACS Technical Report 95-05.
Baldomero-Naranjo, M., Kalcsics, J., Marín, A., Rodríguez-Chía, A.M., 2022. Upgrading edges in the Maximal Covering Location Problem. European J. Oper. Res. 303 (1), 14–36.
Barahona, F., Grötschel, M., 1986. On the cycle polytope of a binary matroid. J. Combin. Theory B 40, 40–62.
Blanco, V., Marín, A., 2019. Upgrading nodes in tree-shaped hub location. Comput. Oper. Res. 102, 75–90.
Campbell, J.F., Corberán, Á., Plana, I., Sanchis, J.M., Segura, P., 2021. Solving the length constrained K-drones rural postman problem. European J. Oper. Res. 292 (1), 60–72.
Campbell, A.M., Lowe, T.J., Zhang, L., 2006. Upgrading arcs to minimize the maximum travel time in a network. Networks 47 (2), 72–80.
Carr, R., Ravi, R., Simonetti, N., 2023. A new integer programming formulation of the graphical traveling salesman problem. Math. Program. 197, 877–902.
Corberán, Á., Plana, I., Rodríguez-Chía, A.M., Sanchis, J.M., 2013. A branch-and-cut algorithm for the maximum benefit Chinese postman problem. Math. Program. 141, 21–48.
Cornuéjols, G., Fonlupt, J., Naddef, D., 1985. The Traveling Salesman Problem on a graph and some related integer polyhedra. Math. Program. 33, 1–27.
Demgensky, I., Noltemeier, H., Wirth, H.-C., 2002. On the flow cost lowering problem. European J. Oper. Res. 137 (29), 265–271.
Deo, N., Pang, C.Y., 1984. Shortest-path algorithms: Taxonomy and annotation. Networks 14 (2), 275–323.
Dilkina, B., Lai, K.J., Gomes, C.P., 2011. Upgrading shortest paths in networks. In: Achterberg, T., Beck, J.C. (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. CPAIOR 2011, In: Lecture Notes in Computer Science, vol. 6697, Springer.
Drangmeister, K.U., Krurnke, S.O., Marathe, M.V., Noltemeier, H., Ravi, S.S., 1998. Modifying edges of a network to obtain short subgraphs. Theoret. Comput. Sci. 203, 91–1211.
Espejo, I., Marín, A., 2020. Upgraded network p-median problem. In: X International Workshop on Locational Analysis and Related Problems, Vol. 51.
Fleischmann, B., 1985. A cutting-plane procedure for the Traveling Salesman Problem on a road network. European J. Oper. Res. 21, 307–317.
Fredericksont, G.N., Solis-Oba, R., 1999. Increasing the weight of minimum spanning trees. J. Algorithms 33, 244–266.
Fulkerson, D.R., Harding, G.C., 1977. Maximizing the minimum source–sink path subject to a budget constraint. Math. Program. 13 (1), 116–118.
Ghiani, G., Laporte, G., 2000. A branch–and–cut algorithm for the Undirected Rural Postman Problem. Math. Program. 87, 467–481.
Glover, F., 1973. Unit-Coefficient Inequalities for Zero–One Programming. Management Science Report 73-7, University of Colorado.
Hambrusch, S.E., Tu, H.-Y., 1997. Edge weight reduction problems in directed acyclic graphs. J. Algorithms 24 (1), 66–93.
Handler, G.Y., Zang, I., I, 1980. A dual algorithm for the constrained shortest path problem. Networks 10 (4), 293–309.
Hartmanis, J., 1982. Computers and intractability: a guide to the theory of np-completeness. Siam Rev. 24.
Krumke, S.O., Noltemeier, H., Marathe, M.V., Ravi, S.S., Drangmeister, K.U., 1997. Modifying networks to obtain low cost trees. In: d'Amore, F., Franciosa, P.G., Marchetti-Spaccamela, A. (Eds.), Graph-Theoretic Concepts in Computer Science. WG 1996, In: Lecture Notes in Computer Science, vol 1197, Springer, Berlin, Heidelberg.
Krumke, S.O., Noltemeier, H., Wirth, H.-C., Marathe, M.V., Ravi, R., Ravi, S., Sundaram, R., 1999. Improving spanning trees by upgrading nodes. Theor. Comput. Sci. 221 (1), 139–155.
Lozano, L., Medaglia, A.L., 2013. On an exact method for the constrained shortest path problem. Comput. Oper. Res. 40 (1), 378–384.
Naddef, D., Rinaldi, G., 1991. The symmetric traveling salesman polytope and its graphical relaxation: Composition of valid inequalities. Math. Program. 51, 359–400.
Naddef, D., Rinaldi, G., 1992. The graphical relaxation: A new framework for the symmetric traveling salesman polytope. Math. Program. 58, 53–88.
Naddef, D., Rinaldi, G., 2007. The symmetric traveling salesman polytope: New facets from the graphical relaxation. Math. Oper. Res. 32 (1), 233–256.
Oswald, M., Reinelt, G., Theis, D.O., 2007. On the graphical relaxation of the symmetric traveling salesman polytope. Math. Program B 110, 175–193.
Paik, D., Sahni, S., 1995. Network upgrading problems. Networks 26, 45–58.
Pekny, J.F., Miller, D.L., 1990. An exact parallel algorithm for the resource constrained traveling salesman problem with application to scheduling with an aggregate deadline. In: Proceedings of the 1990 ACM Annual Conference on Cooperation. pp. 208–214.
Saharoy, D., Sen, S., 2014. Approximation algorithms for budget constrained network upgradeable problems. arXiv:1412.3721.
Sepasian, A.R., 2018. Upgrading the 1-center problem with edge length variables on a tree. Discrete Optim. 29, 1–17.
Sepasian, A.R., Monabbati, E., 2017. Upgrading min–max spanning tree problem under various cost functions. Theoret. Comput. Sci. 704, 87–91.
Sokkappa, P.R., 1990. The Cost-Constrained Traveling Salesman Problem (Ph.D. thesis). Lawrence Livermore National Lab, CA (USA).
Zhang, Q., Guan, X., Pardalos, P.M., 2021. Maximum shortest path interdiction problem by upgrading edges on trees under weighted norm. J. Global Optim. 79 (4), 959–987.