ON THE INTEGRALITY GAP OF THE COMPLETE METRIC STEINER TREE PROBLEM VIA A NOVEL FORMULATION

Ambrogio Maria Bernardelli*
ambrogiomaria.bernardelli01@universitadipavia.it

Eleonora Vercesi†‡
eleonora.vercesi@usi.ch

Stefano Gualandi*
stefano.gualandi@unipv.it

Monaldo Mastrolilli§‡
monaldo@supsi.ch

Luca Maria Gambardella†‡
luca.gambardella@usi.ch

November 1, 2024

ABSTRACT

In this work, we study the metric Steiner Tree problem on graphs focusing on computing lower bounds for the integrality gap of the bi-directed cut (DCUT) formulation and introducing a novel formulation, the Complete Metric (CM) model, specifically designed to address the weakness of the DCUT formulation on metric instances. A key contribution of our work is extending of the Gap problem, previously explored in the context of the Traveling Salesman problems, to the metric Steiner Tree problem. To tackle the Gap problem for Steiner Tree instances, we first establish several structural properties of the CM formulation. We then classify the isomorphism classes of the vertices within the CM polytope, revealing a correspondence between the vertices of the DCUT and CM polytopes. Computationally, we exploit these structural properties to design two complementary heuristics for finding nontrivial small metric Steiner instances with a large integrality gap. We present several vertices for graphs with a number of nodes ≤ 10, which realize the best-known lower bounds on the integrality gap for the CM and the DCUT formulations. We conclude the paper by presenting three new conjectures on the integrality gap of the DCUT and CM formulations for small graphs.

Keywords Metric Steiner Tree Problem · Integrality Gap · Combinatorial Optimization

1 Introduction

Given a graph $G = (V, E)$ with n nodes and cost $c_{ij} \geq 0$ on each edge $\{i, j\} \in E$, and a subset of nodes $T \subset V$ with $|T| \geq 2$, the Steiner Tree Problem (STP) consists of finding the minimum-cost tree that spans T. The nodes in T are called *terminals*, and those in $V \setminus T$ are called *Steiner nodes*. The STP is NP-Hard, and the corresponding decision problem is NP-Complete [Kar10]. The best-known polynomial-time algorithm for the STP guarantees an approximation ratio of 1.39 [Byr+13], and improving this ratio is still an open problem. The only two well-known polynomial-time solvable cases are for $|T| = 2$, which is the shortest path problem, and for $|T| = n$, which is minimum spanning tree. Herein, we restrict to the case $2 < |T| < n$.

The exact solution of STP relies on integer linear programming techniques that use the Bidirected Cut (DCUT) formulation reported below. For a catalog of formulations of STP, we refer to [GM93], while for more recent surveys, to [KM98; Lju21]. The core of the DCUT formulation consists of fixing a root node $r \in T$, replacing each undirected

* Department of Mathematics “Felice Casorati”, University of Pavia, 27100 Pavia, Italy
† Faculty of Informatics, Università della Svizzera italiana, 6900 Lugano, Switzerland
§ Dipartimento Tecnologie innovative, Scuola universitaria professionale della Svizzera italiana, 6900 Lugano, Switzerland
‡ Istituto Dalle Molle di studi sull’intelligenza artificiale (IDSIA USI-SUPSI), 6900 Lugano, Switzerland

edge $\{i, j\}$ with two oriented arcs (i, j) and (j, i), and introducing 0–1 flow variables x_{ij} for each arc. If we denote by A the set of oriented arcs, and by $\delta^-(W) := \{(i, j) \mid i \notin W,\ j \in W\}$ the set of arcs entering one of the nodes in $W \subset V$, the DCUT formulation is as follows:

$$\min \quad \sum_{\{i,j\}\in E} c_{ij}(x_{ij} + x_{ji}) \tag{1a}$$

$$\text{s.t.} \quad x_{ij} + x_{ji} \leq 1, \qquad \{i, j\} \in E, \tag{1b}$$

$$x\left(\delta^-(W)\right) \geq 1, \qquad W \subset V \setminus \{r\},\ W \cap T \neq \emptyset, \tag{1c}$$

$$x_{ij} \in \{0, 1\}, \qquad (i, j) \in A. \tag{1d}$$

Despite their exponential number, the cut constraints (1c) can be separated in polynomial time using any max flow or min-cut algorithm, as detailed in [KM98]. We refer to the DCUT polytope as the set $P_{DCUT} := \{\boldsymbol{x} \in [0, 1]^m \mid$ (1b)–(1c)$\}$, where $m = |A|$. We refer to the LP relaxation defined over P_{DCUT} as RDCUT (Relaxed DCUT). The state-of-the-art (parallel) implementation of an exact STP solver based on the DCUT model is SCIP-Jack [Gam+17]. Other more recent approaches are reviewed in [Lju21].

Whenever the graph G is complete, and the edge costs are metric, we have a complete metric Steiner Tree problem. The edge costs define a metric if they satisfy the following properties: (i) $c_{ij} = 0$ if and only if $i = j$; (ii) $c_{ij} \geq 0$ (*positivity*), (iii) $c_{ij} = c_{ji}$ (*symmetry*), (iv) $c_{ij} \leq c_{ik} + c_{jk}$ (*triangle inequality*). Metric STP instances are relevant in particular for VLSI circuit design [Pey07], and efficient combinatorial algorithms exist for rectilinear costs [HSV17] and for packing Steiner trees [GMW97]. A review of results for metric (and rectilinear) STP is contained in [HSV17].

In approximation algorithms, we are interested in studying the *integrality gap* of an integer formulation, defined as the supremum among all the ratios between an integer linear model and its natural LP relaxation. If the optimal values of DCUT and RDCUT defined over a graph G with a set of terminals T are denoted by DCUT(G, T) and RDCUT(G, T), respectively, then the integrality gap for the STP is defined as

$$\alpha := \sup_{G=(V,E),\ T\subset V} \frac{\text{DCUT}(G, T)}{\text{RDCUT}(G, T)}. \tag{2}$$

For the STP, the exact value of α is unknown, but it is bounded below by $\frac{6}{5}$ [Vic20], which improved the previous bound of $\frac{36}{31} \cong 1.161$ in [Byr+13]. Until 2024, the best-known upper bound for the integrality gap was two. Recently, [BGT24] demonstrated that this upper bound can be improved to 1.9988. Note that proving that $\alpha < 1.39$ would lead to a better approximation algorithm with respect to the state of the art. The lower bound introduced by [Byr+13] is based on a recursive family of instances, depending on a parameter p and having an integrality gap which tends asymptotically to $\frac{36}{31}$ for $n \to \infty$.

Main Contributions. This paper presents a novel formulation for the complete metric Steiner Tree Problem, which we call the Complete Metric (CM) formulation. This formulation exploits the metric costs to define a polytope, denoted by P_{CM} and defined in (15b)–(15f), having a smaller number of vertices compared to the polytope P_{DCUT} implied by the DCUT formulation. The main motivation for introducing our formulation is to enable a study on the integrality gap of small-size instances of the Steiner Tree problem by adapting the approach designed for the Symmetric TSP and presented in [BE05; BE07]. Without our new CM formulation, it would be nearly impossible to use the method of [BE07] due to the number of vertices of P_{DCUT}, which includes a huge number of feasible vertices for the cut constraints but which will never be optimal for any *metric* cost vector. For instance, Table 1 reports the number of feasible and optimal vertices of P_{DCUT} and P_{CM}, computed by complete enumeration using Polymake, for instances with 4 and 5 nodes and 3 or 4 terminals, showing the potential impact of our approach on the overall number of vertices.

The core intuition of our new CM formulation is that in a complete metric graph, any Steiner node is visited only if its outdegree is at least 2, because if the indegree and outdegree of a Steiner node are both equal to 1, then an optimal solution with a smaller cost that avoids detouring in that node exists. The existence of such a solution is guaranteed by the property that the graph is metric and complete. Indeed, such a solution may not exist in a non-complete graph. Using these relations on the degree of Steiner nodes, we introduce a new family of constraints to the DCUT formulation, reflecting our main intuition.

The main contributions of this paper are as follows:

1. We prove in Theorem 1 that our new polytope P_{CM} contains among the integer vertices only those that could be optimal for metric costs. Given an integer vertex, we describe the cost vector that makes that vertex optimal.

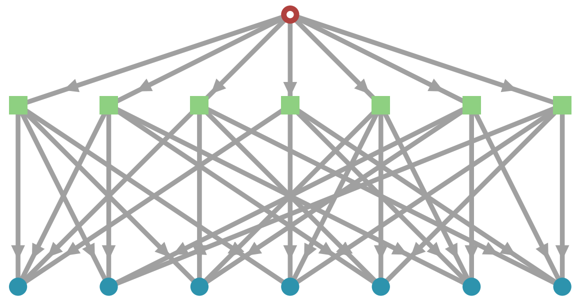


Figure 1: Small STP instance with a large integrality gap: Skutella's graph with $n = 15$, $t = 8$, and $\alpha = \frac{8}{7}$ [KPT11]. The hollow circle represents the root, the circles represent the terminals, and the squares represent the Steiner nodes. Every arc correspond to a variable x_{ij} of value equal to $\frac{1}{4}$.

Table 1: Number of feasible and optimal vertices for P_{DCUT} and P_{CM}. While the DCUT polytope has several (feasible) vertices that cannot be optimal for any metric cost, the CM polytope does not suffer this issue (and it implicitly reduces the number of isomorphic vertices).

		P_{DCUT}		P_{CM}	
n	t	feasible	optimal	feasible	optimal
4	3	256	70	4	4
5	3	28 345	3 655	5	5
5	4	24 297	3 645	44	44

2. We prove in Lemma 3 a connectedness property of the points of P_{CM}, and in Lemma 4 we set an upper bound on the number of edges in integer points.

3. We characterize in Lemma 6–8 isomorphic vertices of polytopes of different dimensions, namely, we link vertices of the polytope corresponding to the STP with n nodes and t terminals with the vertices corresponding to the STP with $n + 1$ nodes and t terminals, and vice versa.

4. Exploiting the previous results, we introduce two new heuristic algorithms for enumerating the vertices of P_{CM}. Using these two algorithms, we compute vertices of P_{CM} and P_{DCUT} with the largest known integrality gap for instances with up to 10 vertices.

To the best of our knowledge, this work is the first attempt to extend the work of [BE07] to the metric Steiner Tree problem. Note that an interesting STP instance with a small value of n (i.e., $n = 15$), but with a large integrality gap (i.e, equal to $\frac{8}{7}$) is the Skutella's graph shown in Figure 1 and reported in [KPT11]. In our computational results, we will present other interesting instances with less than 10 nodes but having a large integrality gap, see Figure 4.

Outline The outline of this paper is as follows. Section 2 reviews the approach introduced in [BB08] for computing the integrality gap of small instances ($n \leq 10$) of the Traveling Salesman Problem and presents how to apply a similar approach to the DCUT formulation of the STP. The critical step is the complete enumeration of the vertices of the RDCUT polytope, which are several thousand already for $n = 5$, as shown in Table 1. Section 3 presents the CM formulation, and we prove interesting properties of the corresponding polytope, which allows us to apply the methodology of [BB08] to look for small instances of STP with a large integrality gap. In Section 4, we observe that the exhaustive enumeration of vertices is intractable for $n \geq 6$, and we present two heuristic procedures for generating vertices of RCM, exploiting graph isomorphism. In Section 5, we present the vertices for graph $n \leq 10$, realizing the best-known lower bounds on the integrality gap for the CM and the DCUT formulation. We conclude the paper with a perspective on future works.

2 Integrality gap of DCUT for fixed n

In this section, we present our strategy to compute the integrality gap of the DCUT formulation for STP, which is based on the approach introduced in [BB08; Ell08].

2.1 Problem definition for computing the integrality gap of small instances

Let $P_{\text{DCUT}}(n,T)$ be the polytope defined by the relaxation of constraints (1b), (1c). Let us consider the complete graph $K_n(c) = (V,E)$, which has n nodes and a cost function c on edges. We define

$$\alpha_{c,T} := \frac{\text{DCUT}(K_n(c),T)}{\text{RDCUT}(K_n(c),T)} \tag{3}$$

$$\alpha_{n,t} := \sup_{c \in \mathcal{C}(V,t)} \alpha_{c,T}, \tag{4}$$

where $\mathcal{C}(V,t) := \{c \text{ metric}, \mid T \subset V,\ |T| = t\}$ Note that $\alpha_{c,T}$ is the integrality gap of a given instance of metric STP, while $\alpha_{n,t}$ is the maximum integrality gap once we fix both the cardinality of T and V. Clearly, the integrality gap (2) is equal to

$$\alpha = \sup_{n,t} \alpha_{n,t}. \tag{5}$$

As mentioned in the introduction, the two cases $t = 2$ and $t = n$ can be solved in polynomial time for every n, since for $t = 2$ the problem reduces to the shortest path (SP) problem, while for $t = n$ it reduces to the minimum spanning tree (MST) problem. In general, the polynomial time algorithms for SP (e.g., [Dij59]) and MST (e.g., [Pri57]) do not use the IP formulation. Hence, the existence of a polynomial time algorithm does not naturally imply an integral formulation when coming to the DCUT formulation. In practice, for $t = n$, the polyhedron is integral [Edm+67], while [GM93] shows the same result for $t = 2$, and hence in these two cases $\alpha = 1$. However, the exact value of α for $2 < t < n$ is unknown.

In this work, we extend the approach presented in [BB08; Ell08] for TSP for computing the exact value of α for the DCUT formulation. Similarly to [BB08], if we divide the costs $c_{ij}, i, j \in V$ of an instance STP(c,T) for the optimal value DCUT$(K_n(c),T)$, we obtain another instance STP(c',T), having an optimal value DCUT$(K_n(c'),T) = 1$ but the same set of optimal solutions. Hence, defining $\mathcal{C}^{DCUT}(V,t) := \{c \text{ metric}, T \subset V,\ |T| = t, \text{DCUT}(K_n(c),T) = 1\}$, we obtain

$$\alpha_{n,t} := \sup_{c \in \mathcal{C}^{DCUT}(V,t)} \frac{1}{\text{RDCUT}(K_n(c),T)},$$

and hence

$$\frac{1}{\alpha_{n,t}} := \inf_{c \in \mathcal{C}^{DCUT}(V,t)} \text{RDCUT}(K_n(c),T). \tag{6}$$

2.2 Graph isomorphism and vertices of the DCUT polytope

We observe that the choice of the terminals is irrelevant for the integrality gap, and the only thing that matters is the *number* of terminals. Consider an instance STP(c,T) where $|T| = t$. This can be re-mapped to STP$(c,\{1,\dots,t\})$ trough a node-colored-edge-weighted graph isomorphism. More formally,

Definition 1 (Graph isomorphism)**.** Let $G = (V,E)$ and $H = (V',E')$ be two undirected graphs, with $|V| = |V'| = n$. The two graphs are *isomorphic*, denoted by $G \cong H$, if there exists a bijection $\sigma : V \to V'$ such that

$$\{i,j\} \in E \iff \{\sigma(i),\sigma(j)\} \in E', \quad \forall \{i,j\} \in E. \tag{7}$$

We also say, with a slight abuse of notation, that $\sigma : G \to H$ is an *isomorphism* between G and H. If the two graphs are edge-weighted, with $w : E \to \mathbb{R}$ and $w' : E' \to \mathbb{R}$ being the weight functions of G and H respectively, the two graphs are isomorphic if it holds

$$w(\{i,j\}) = w'(\{\sigma(i),\sigma(j)\}), \quad \forall \{i,j\} \in E. \tag{8}$$

This definition naturally extends to the case of directed graphs by considering the arc orientation. If the two graphs are node-colored, with $\mathsf{c} : V \to C$ and $\mathsf{c}' : V' \to C'$ being the color mappings of G and H, respectively, it must also hold that

$$\mathsf{c}(i) = \mathsf{c}(j) \iff \mathsf{c}'(\sigma(i)) = \mathsf{c}'(\sigma(j)), \quad \forall i,j \in V. \tag{9}$$

Each instance of the STP naturally leads to an edge-weighted node-colored graph with three colors: one for the root node, one for the terminal minus the root, and one for the nodes in $V \setminus T$, while the edge-weight function is represented by the cost c.

Equation (6) implicitly states that the integrality gap does not depend on the set T but solely on its cardinality. This observation is expressed in the following Lemma, which relies on Definition 1.

Observation 1. Let $K_n(c)$ be a complete metric graph and let $T \subset V$, $|T| = t$, such that DCUT$(K_n(c), T) = 1$. Let σ be an edge-weighted node-colored graph isomorphism. Then

$$\text{DCUT}(\sigma(K_n(c)), \sigma(T)) = \text{DCUT}(K_n(c), T),$$
$$\text{RDCUT}(\sigma(K_n(c)), \sigma(T)) = \text{RDCUT}(K_n(c), T).$$

This follows from a direct application of the definition of graph isomorphism.

Note that (6) lead to an optimization problem having linear constraints but a quadratic objective function (note that c_e and x_{ij} are both decision variables)) that can be formulated as follows:

$$\begin{aligned}
\min \quad & \sum_{\{i,j\} \in E} c_e(x_{ij} + x_{ji}) && && \text{(10a)}\\
\text{s.t.} \quad & x_{ij} + x_{ji} \le 1, && e = \{i,j\} \in E, && \text{(10b)}\\
& x\left(\delta^-(W)\right) \ge 1, && W \subset V \setminus \{1\},\ W \cap \{1,\dots,t\} \neq \emptyset, && \text{(10c)}\\
& 0 \le x_{ij} \le 1, && \forall i,j \in V,\ i \neq j, && \text{(10d)}\\
& c_{ij} \ge 0, && \forall \{i,j\} \in E, && \text{(10e)}\\
& c_{ij} \le c_{ik} + c_{jk}, && \forall \{i,j\},\{i,k\},\{j,k\} \in E. && \text{(10f)}
\end{aligned}$$

Constraints (10b)–(10d) ensures the feasibility of vector x, while constraints (10e)–(10f) ensure the property of c being metric. Our preliminary computational results show that (10a)–(10f) are intractable even for $n = 5$ and $t = 3, 4$. Therefore, we proceed as done in [BB08; BE07; Ell08], exploiting the vertex representation of $P_{\text{DCUT}}(n, T)$. Since we fix the terminal set to $T = \{1, \dots, t\}$, herein we denote $P_{\text{DCUT}}(n, t) := P_{\text{DCUT}}(n, \{1, \dots, t\})$. Similarly to [BB08; BE07; Ell08], and by recalling that for each cost vector c, there exists an optimal solution attained at a vertex, we can re-write (10a)–(10f) as a *linear program* for each vertex $\overline{x}$ of $P_{\text{DCUT}}(n, t)$.

$$\begin{aligned}
\text{(Gap)} \quad \min \quad & \sum_{\{i,j\} \in E} c_e(\overline{x}_{ij} + \overline{x}_{ji}) && && \text{(11a)}\\
\text{s.t.} \quad & c_{ij} \ge 0 && \forall \{i,j\} \in E, && \text{(11b)}\\
& c_{ij} \le c_{ik} + c_{jk} && \forall \{i,j\},\{i,k\},\{j,k\} \in E, && \text{(11c)}\\
& \text{the optimal solution of } c \text{ is attained at } \overline{x}, && && \text{(11d)}
\end{aligned}$$

and define Gap$(\overline{x})$ as the optimal value of (11).

As in [BB08; Ell08], we observe that constraint (11d) can be formulated using the complementary slackness conditions. Such conditions ensure that $\overline{x}$ belongs to the point minimizing the STP at cost c. Given a vertex $\overline{x}$ of $P_{\text{DCUT}}(n, t)$, we present the Gap problem for the DCUT formulation. This model is the starting point from which the Gap problem for the CM formulation is derived. Denoting by A the set of the orientated arcs, that is, $A := \{(i,j) \mid \text{s.t}\{i,j\} \in E\}$, we have:

$$\begin{aligned}
\min \quad & \sum_{\{i,j\} \in E} c_e(\overline{x}_{ij} + \overline{x}_{ji}) && && \text{(12a)}\\
\text{s.t.} \quad & c_{ij} \le c_{ik} + c_{jk} && \forall \{i,j\},\{i,k\},\{j,k\} \in E, && \text{(12b)}\\
& y_e + \sum_{(i,j) \in \delta^-(W)} z_W + d_{ij} \le c_e, && \forall (i,j) \in A, && \text{(12c)}\\
& y_e = 0, && \forall (i,j) \text{ s.t. } \overline{x}_{ij} + \overline{x}_{ji} < 1,\ e = \{i,j\} \in E, && \text{(12d)}\\
& z_W = 0, && \forall W \subset V \setminus \{r\},\ W \cap T \neq \emptyset \text{ s.t.} \sum_{(i,j) \in \delta^-(W)} \overline{x}_{ij} > 1, && \text{(12e)}\\
& d_{ij} = 0, && \forall (i,j) \in A \text{ s.t } \overline{x}_{ij} = 1, && \text{(12f)}\\
& y_e + \sum_{(i,j) \in \delta^-(W)} z_W + d_{ij} - c_e = 0, && \forall (i,j) \in A \text{ s.t } \overline{x}_{ij} > 0, && \text{(12g)}\\
& y_e, d_{ij}, d_{ji} \le 0, && \forall e = \{i,j\} \in E, && \text{(12h)}\\
& z_W \ge 0, && \forall W \subset V \setminus \{r\},\ W \cap T \neq \emptyset, && \text{(12i)}\\
& c_{ij} \ge 0, && \forall \{i,j\} \in E. && \text{(12j)}
\end{aligned}$$

3 A novel formulation for the complete metric case

Scip-Jack, the state-of-art solver for the (Graphic) STP, relies on a modified version of the DCUT formulation introduced in [Gam+17]. The formulation used by Scip-Jack (SJ) is as follows:

$$
\begin{aligned}
\text{(SJ)} \quad \min \quad & \sum_{\{i,j\}\in E} c_e(x_{ij}+x_{ji}) && && \text{(13a)}\\
\text{s.t.} \quad & x\left(\delta^-(W)\right) \geq 1 && W \subset V\setminus\{r\},\ W\cap T \neq \emptyset, && \text{(13b)}\\
& x\left(\delta^-(r)\right) = 0 && && \text{(13c)}\\
& x\left(\delta^-(v)\right) = 1 && v \in T\setminus\{r\}, && \text{(13d)}\\
& x\left(\delta^-(v)\right) \leq 1 && v \in S, && \text{(13e)}\\
& x\left(\delta^-(v)\right) \leq x\left(\delta^+(v)\right) && \forall v \in S, && \text{(13f)}\\
& x\left(\delta^-(v)\right) \geq x_a && \forall a \in \delta^+(v), v\in S, && \text{(13g)}\\
& x_a \in \{0,1\}, && \forall a \in A, && \text{(13h)}
\end{aligned}
$$

where $\delta^+(W) := \{(i,j) \mid i \in W,\ j \notin W\}$.

Constraints (13c)–(13e) describe the inflow of every node: the first equation ensures that no inflow is present in the root, the second equation ensures that the inflow of terminal nodes is exactly equal to 1, since every terminal must be reached, and the third equation ensures that the inflow of non-terminal nodes is smaller or equal than 1 since a non-terminal node may or may not be part of an optimal solution. Note that both terminal and non-terminal nodes have an inflow of at most 1 so that at most one path exists from the root to any node. Constraint (13f) ensures that non-terminal nodes cannot be leaves of the solution. Constraint (13g) ensures that no flow is generated from non-terminal nodes. Notice that this formulation is not specific to the metric we want to attack, as illustrated by the example below.

Example 1. Let $G = (V,E)$ be a complete metric graph with $V = \{0,1,2,3,4\}$ and let $T = \{0,1\}$. Define x as the following

$$
x_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \{(0,1),(2,3),(3,4),(4,2)\},\\ 0, & \text{else.} \end{cases} \tag{14}
$$

We have that x is feasible for the SJ formulation with $r = 0$, but it is never optimal for any metric cost since by setting $x_{2,3} = x_{3,4} = x_{4,2} = 0$ we obtain a feasible solution with a strictly smaller cost. Note that, in particular, x is not connected.

To prevent this issue, we introduce a stronger formulation tailored to the Complete Metric (CM) case, which is presented below.

$$
\begin{aligned}
\text{(CM)} \quad \min \quad & \sum_{\{i,j\}\in E} c_e(x_{ij}+x_{ji}) && && \text{(15a)}\\
\text{s.t.} \quad & x\left(\delta^-(W)\right) \geq 1 && W \subset V\setminus\{r\},\ W\cap T \neq \emptyset, && \text{(15b)}\\
& x\left(\delta^-(r)\right) = 0 && && \text{(15c)}\\
& x\left(\delta^-(v)\right) \leq 1 && v \in V\setminus r, && \text{(15d)}\\
& 2x\left(\delta^-(v)\right) \leq x\left(\delta^+(v)\right) && \forall v \in S, && \text{(15e)}\\
& 0 \leq x_a \leq 1, && \forall a \in A, && \text{(15f)}\\
& x_a \in \{0,1\}, && \forall a \in A. && \text{(15g)}
\end{aligned}
$$

By forgetting about (15g), we obtain the P_{CM} polytope. In particular, in our new formulation, the left-hand side of Constraint (13f) is multiplied by 2. This ensures that a non-terminal node is visited only if its outflow is at least 2. The idea is that, in a complete metric graph, if the inflow and the outflow of a non-terminal node are both equal to 1, then there exists an optimal solution with a smaller cost that avoids detouring in that node. The existence of such a solution is guaranteed by the property that the graph is metric and complete. Note that such a solution may not exist in a non-complete graph, for example, when $G = (V,E)$ with $V = \{0,1,2\}$, $E = \{\{0,2\},\{1,2\}\}$ and $T = \{0,1\}$. We also avoid adding the equivalent of Constraint (13g) because of the following lemma.

Lemma 1. *When dealing with positive costs, Constraint* (13g) *is redundant even for the simpler DCUT formulation.*

Before proving Lemma 1, we recall the *Multi Commodity Flow* (MCF) formulation [CT01]:

$$\text{(MCF)} \min \sum_{\{i,j\}\in E} c_e(x_{ij} + x_{ji}) \tag{16a}$$

$$\text{s.t.} \quad x_{ij} + x_{ji} \le 1, \qquad e = \{i,j\} \in E, \tag{16b}$$

$$f^t\left(\delta^-(v)\right) - f^t\left(\delta^+(v)\right) = \begin{cases} -1, & \text{if } v = r \\ 1, & \text{if } v = t \\ 0, & \text{otherwise,} \end{cases} \qquad v \in V,\ t \in T \setminus \{r\} \tag{16c}$$

$$f_{ij}^t \le x_{ij} \qquad \forall (i,j) \in A, \tag{16d}$$

$$f_{ij}^t, x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A. \tag{16e}$$

This formulation has a few interesting properties that we use in the following proof of Lemma 1.

Proof of Lemma 1. Let x_{ij} be an optimum vertex for the DCUT formulation with a positive cost c. By Theorem 3.2 of [CT01], in particular, because of the equivalence

$$\min\{c^\intercal x \mid x \in P_{\text{MCF}}(n,t)_{|x}\} = \min\{c^\intercal x \mid x \in P_{\text{DCUT}}(n,t)\},$$

there exists a configuration of variables f_{ij}^t such that $f_{ij}^t \le x_{ij}$ for every $t \in T,\ i,j \in V$ and $\sum_i f_{ij}^t - \sum_i f_{ji}^t = 0$ for every $t \in T,\ j \in S$. Because x_{ij} is optimal for strictly positive costs, we have that $x_{ij} = \max_{t\in T} f_{ij}^t$ and so there exists $t_{ij} \in T$ such that $x_{ij} = f_{ij}^{t_{ij}}$. Now, let $k \in S$. For every $a \in \delta^+(k)$, that is, for every $l \in V \setminus \{k\}$ we have that

$$\begin{aligned}
x_a &= x_{kl} && \text{by definition} \\
&= f_{kl}^{t_{kl}} && \text{by maximization} \\
&\le \sum_i f_{ki}^{t_{kl}} && \text{by nonnegativity} \\
&= \sum_i f_{ik}^{t_{kl}} && \text{by (16c)} \\
&\le \sum_i x_{ik} && \text{by (16d)} \\
&= x(\delta^-(k)) && \text{by definition}
\end{aligned}$$

which is equivalent to Constraint (13g). □

Before discussing how we use this formulation to retrieve information regarding the integrality gap of the DCUT formulation, we list some properties of the CM formulation that we consider interesting.

3.1 Properties of the complete metric formulation

We first show that for a particular configuration of complete metric graphs, namely, graphs with no triples of collinear points, the set of integer solutions of the SJ formulation coincides with the set of integer solutions of the CM formulation.

Lemma 2. *Let G be a complete metric graph with $c \in \mathbb{R}^{(n-1)\times n}$ defining the edge weights and let $x \in \{0,1\}^{(n-1)\times n}$ be an optimal solution of SJ with cost vector c. If*

$$c_{ij} < c_{ik} + c_{kj} \quad \forall \{i,j\},\{i,k\},\{j,k\} \in E, \tag{17}$$

then x is also an optimal solution for CM with the same cost vector. Moreover, if $y\ in\{0,1\}^{(n-1)\times n}$ is an optimal solution of CM for G, then it is also an optimal solution of SJ for G.

Proof. Suppose by contradiction that an optimal solution of SJ exists, which is not an optimal solution for CM. Because of the constraints that describe the two models, this solution x must verify

$$\sum_{i\neq j} x_{ij} \le \sum_{k\neq j} x_{jk}, \qquad 2 \cdot \sum_{i\neq j} x_{ij} > \sum_{k\neq j} x_{jk},$$

for a certain $j \in V \setminus T$. It follows that there exists $i, k \in V$ such that $x_{ij} = x_{jk} = 1$. Since we are in a complete graph, setting these two variables to zero and setting $x_{ik} = 1$ gives us a feasible solution, which is also of smaller cost because of hypothesis (17), which is in contradiction with the optimality of x.

Let y be an optimal solution for CM for G. Clearly, y is feasible for SJ. Suppose by contradiction that there exists z feasible for SJ such that $c^\intercal z < c^\intercal y$. For the first part of the proof, we have that z is an optimal solution for CM, which contradicts the optimality of y. □

Observation 2. Note that, without hypothesis (17), we can say that given a metric cost c and x an optimal solution of DCUT with c as the cost vector, there exists x' an optimal solution for DCUT with c as the cost vector such that x' is also an optimal solution for CM with the same cost vector. In particular, x' is chosen as one of the optimal solutions of DCUT that avoids detouring into non-terminal nodes, where detouring into a node means entering with one edge and exiting with one edge.

Observation 3. Note that Lemma 2 does not hold when replacing SJ with RSJ and CM with RCM. Take, for example, as graph G the metric completion of the instance `se03` of the SteinLib [KMV01]. We have that

$$\text{RSJ}(G, T) = 11 < 12 = \text{RCM}(G, T) = \text{SJ}(G, T).$$

We then have that $\text{SJ}(\cdot) = \text{CM}(\cdot)$ and $\text{RSJ}(\cdot) \leq \text{RCM}(\cdot)$, and so the integrality gap of the CM formulation is a lower bound for the integrality gap of the SJ formulation on complete metric graphs. Moreover, the bound is not always tight. The same holds for the DCUT formulation.

An interesting property of the CM formulation is connectedness. Constraints (13b) enforce that in an SJ solution, all the terminal nodes belong to the same connected component, but this is not guaranteed for non-terminal nodes. For the CM formulation instead, the following lemma holds.

Lemma 3. *The support graph of any feasible point of RCM is connected.*

Proof. It suffices to prove that no connected components without terminals exist since every terminal belongs to the same connected component because of Constraint (15b). Let x be a feasible point for CM and let $H \subset V$ be a connected component of x containing no terminals. Constraint (15e) implies

$$\sum_{i,j \in H} x_{ij} = \sum_{j \in H} \sum_{i \in H} x_{ij} \geq \sum_{j \in H} 2 \sum_{i \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ij}.$$

The only possibility is that $\sum_{i,j \in H} x_{ij} = 0$, so no connected component without terminals can be part of a feasible solution for CM. □

Note that Lemma 3 does not hold for the SJ formulation, as Example 1 shows.

Another interesting property of the CM formulation deals with constraint reduction. In this case, we can prove theoretical results on the number of edges in a CM solution and, consequently, on the number of Steiner nodes.

Lemma 4. *Let x be a feasible solution for the CM formulation for a graph with $|V| = n$ nodes and $|T| = t$ terminals. Then, x verifies*

$$\sum_{i,j} x_{ij} \leq \min(n - 1, 2t - 3). \tag{18}$$

Proof. Given x, let $G_x = (V_x, E_x)$ denote the corresponding support graph, that is $V_x = \{i \in V \,:\, x(\delta^+(i)) + x(\delta^-(i)) > 0\}$ and $E_x = \{e = \{i, j\} \in E \,:\, x_{ij} + x_{ji} > 0\}$. We know that G_x is acyclic because of Constraint (15d) and we also know that G_x is connected because of Lemma 3, so G_x is a tree. Since $|V_x| \leq n$, we have that $\sum_{i,j} x_{ij} \leq n - 1$. Now, we only need to prove that $\sum_{i,j} x_{ij} \leq 2t - 3$. We have that

$$\sum_{i,j} x_{ij} = \sum_{j} \sum_{i \neq j} x_{ij} = \sum_{j \in T} \sum_{i \neq j} x_{ij} + \sum_{j \in V \setminus T} \sum_{i \neq j} x_{ij} =$$
$$= \sum_{i \neq r} x_{ir} + \sum_{j \in T \setminus \{r\}} \sum_{i \neq j} x_{ij} + \sum_{j \in V \setminus T} \sum_{i \neq j} x_{ij} \leq 0 + (t - 1) + \frac{1}{2} \sum_{j \in V \setminus T} \sum_{k \neq j} x_{jk},$$

where the last inequality holds because of Constraint (15c), Constraint (15d) combined with Constraint (15b), and Constraint (15e), respectively. Note that only the last one gives us inequality since the others hold with equality. We can now rewrite

$$\sum_{j \in V \setminus T} \sum_{k \neq j} x_{jk} = \sum_{i,j} x_{ij} - \sum_{j \in T} \sum_{k \neq j} x_{jk}.$$

Combining this with the previous equation, we get that

$$\sum_{i,j} x_{ij} \leq t - 1 + \frac{1}{2}\sum_{i,j} x_{ij} - \frac{1}{2}\sum_{j \in T}\sum_{k \neq j} x_{jk}.$$

Rearranging the terms, we obtain

$$\frac{1}{2}\sum_{i,j} x_{ij} \leq t - 1 - \frac{1}{2}\sum_{j \in T}\sum_{k \neq j} x_{jk}$$

and hence, multiplying by 2

$$\begin{aligned}\sum_{i,j} x_{ij} &\leq 2t - 2 - \sum_{j \in T}\sum_{k \neq j} x_{jk} = \\ &= 2t - 2 - \sum_{k \neq r} x_{rk} - \sum_{j \in T \setminus \{r\}}\sum_{k \neq j} x_{jk} \leq 2t - 2 - 1 - 0 = 2t - 3,\end{aligned}$$

where the last inequality holds because $\sum_{k \neq r} x_{rk} \geq 1$ by taking $W = V \setminus \{r\}$ in Constraint (15b), and because $x_{jk} \geq 0$, respectively. $\square$

Observation 4. Let $t \leq \frac{n}{2} + 1$ and so $\min(n-1, 2t-3) = 2t - 3$. Then, if we consider the CM, our solution is a tree with at most $2t - 3$ edges, so it has $2t - 3 + 1 = 2t - 2$ nodes, t of which are terminals, leaving us with $t - 2$ Steiner vertices. Thus, it suffices to write Constraints (15b) only for

$$W = W_1 \sqcup W_2, \quad W_1 \subset T \setminus r,\ |W_1| \geq 1, \quad W_2 \subset V \setminus T,\ |W_2| \leq t - 2, \tag{19}$$

instead of writing it for any $W = W_1 \sqcup W_2$, $W_2 \subset V \setminus T$. For instance, in the case $(n, t) = (20, 5)$ we go from $(2^4 - 1) \times 2^{15} = 491520$ possible choices of W to just $(2^4 - 1) \times \sum_{i=0}^{3}\binom{15}{i} = 8640$, which is around 1.8% of the total.

After discussing the properties that make the CM formulation interesting by itself, we now focus on commenting on its advantages in deducing information on the lower bounds of the DCUT.

First, we discuss why studying the complete metric case is not restrictive. In particular, we use the *metric closure* of a graph, defined below.

Definition 2 (Metric Closure of a Graph)**.** Let $G = (V, E)$ be an edge-weighted connected graph. We define the *metric closure* of G the complete metric graph $\bar{G} = (V, \bar{E})$ such that the weight of the edge $\{i, j\}$ in $\bar{G}$ is equal to the value of the shortest paths from i to j in G.

We now link the integrality gap of the DCUT formulation of a graph to the corresponding integrality gap of its metric closure.

Lemma 5. *Let $G = (V, E)$, $T \subset V$ be a Steiner instance, and let $\bar{G}$ be the Steiner instance corresponding to the metric closure of G. Then we have that*

$$\textit{DCUT}(G, T) = \textit{DCUT}(\bar{G}, T), \quad \textit{RDCUT}(G, T) = \textit{RDCUT}(\bar{G}, T). \tag{20}$$

Proof. Let x be a feasible solution for G. Then, it is also a feasible solution for $\bar{G}$, and because of the definition of metric closure, it is a feasible solution with a smaller cost. We have then that $\text{DCUT}(\bar{G}, T) \leq \text{DCUT}(G, T)$. Let now $\bar{x}$ be a feasible solution for $\bar{G}$. Reasoning in a non-oriented way, if we take every edge of $\bar{x}$ and substitute it with the corresponding shortest path in G, we obtain a subgraph of G that can be oriented as a feasible solution x of G, with a smaller cost. The cost is (non-strictly) smaller because we may take the same edge in different shortest paths. We then have that $\text{DCUT}(\bar{G}, T) \geq \text{DCUT}(G, T)$ and so $\text{DCUT}(\bar{G}, T) = \text{DCUT}(G, T)$.

For the same reasoning, we have that $\text{RDCUT}(\bar{G}, T) = \text{RDCUT}(G, T)$, with the exception that, when substituting an edge of $\bar{G}$ with the corresponding shortest path in G, since we are dealing with fractional solutions, if we have to take the same edge multiple times because it appears in multiple shortest paths, we have to take the minimum between 1 and the sum of all the values with which that edge appears. This choice preserves feasibility and does not produce a solution with a larger cost. $\square$

From this lemma, it follows that the integrality gap calculated with respect to the CM formulation only on metric graphs is a lower bound for the integrality gap of the DCUT formulation across all graphs.

3.2 The gap problem for the CM formulation

With this in mind, one can proceed as in Section A.1 and define a Gap problem for the CM formulation. Given $\overline{x}$ vertex of $P_{\text{CM}}(n,t)$, we define its gap as the linear problem of finding the cost vector that maximizes the integrality gap of a vertex $\overline{x}$, among those for which $\overline{x}$ is optimal. With this in mind, one can proceed as in Section A.1 and define a Gap problem for the CM formulation. Given $\overline{x}$ vertex of $P_{\text{CM}}(n,t)$, we define its gap as the linear problem of finding the cost vector that maximizes the integrality gap of a vertex $\overline{x}$, among those for which $\overline{x}$ is optimal. As the structure is nontrivial, we first write the constraints of the dual formulation, which comes from the simple application of the duality from problem (15b)–(15g). Table 2 shows the relation between the dual variables and the constraints.

Let $W(i,j)$ defined as

$$W(i,j) := \{W \mid W \subset V \setminus \{r\},\ W \cap T \neq \emptyset,\ (i,j) \in \delta^-(W)\}$$

Then, using the theory of LP duality, we can write the following.

$$y_{ri} + v_i + \sum_{w \in W(r,i)} z_w \le c_{ri}, \qquad i \in T \setminus \{r\}, \tag{21a}$$

$$y_{rj} + v_j + 2u_j + \sum_{w \in W(r,j)} z_w \le c_{rj}, \qquad j \in S, \tag{21b}$$

$$y_{ij} + v_j + \sum_{w \in W(i,j)} z_w \le c_{ij}, \qquad i,j \in T \setminus \{r\}, \tag{21c}$$

$$y_{ij} + v_j + 2u_j + \sum_{w \in W(i,j)} z_w \le c_{ij}, \qquad i \in T \setminus \{r\}, j \in S, \tag{21d}$$

$$q + y_{ir} \le c_{ir}, \qquad \forall i \in T \setminus \{r\}, \tag{21e}$$

$$q + y_{jr} \le c_{jr}, \qquad \forall j \in S, \tag{21f}$$

$$y_{ji} + v_i - u_j + \sum_{w \in W(j,i)} z_w \le c_{ji}, \qquad i \in T \setminus \{r\}, j \in S, \tag{21g}$$

$$y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i,j)} z_w \le c_{ij}, \qquad i,j \in S, \tag{21h}$$

$$q \text{ free},\ z \ge 0,\ v_j, u_j, y_{ij} \le 0. \tag{21i}$$

Note that we can merge some constraints, in particular, (21a) and (21c) are the same constraint where $i \in T$ and $j \in T \setminus \{r\}$. The same holds for (21b) and (21d) if $i \in T$ and $j \in S$. Lastly, we can drop constraint (21e) and (21f) as variable q is free, and it only appears in these constraints. Note that, referring to the primal formulation, this would imply deleting the variables $x_{ir},\ i \in V \setminus \{r\}$. Hence, the dual polytope can be rewritten as

$$y_{ij} + v_j + \sum_{w \in W(ij)} z_w \le c_{ij}, \qquad i \in T, j \in T \setminus \{r\}, \tag{22a}$$

$$y_{ij} + v_j + 2u_j + \sum_{w \in W(i,j)} z_w \le c_{ij}, \qquad i \in T,\ j \in S, \tag{22b}$$

$$y_{ji} + v_i - u_j + \sum_{w \in W(j,i)} z_w \le c_{ji}, \qquad i \in T \setminus \{r\}, j \in S, \tag{22c}$$

$$y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i,j)} z_w \le c_{ij}, \qquad i,j \in S, \tag{22d}$$

$$z \ge 0,\ v_j, u_j, y_{ij} \le 0. \tag{22e}$$

Given a vertex $x \in P_{\text{CM}}(n,t)$, we introduce variables $c_{ij},\ \{i,j\} \in E$ that satisfy the triangle inequality and non-negativity constraints. Adding them to the slackness compatibility conditions we obtain the following linear program with exponentially many variables and constraints:

$$\min \sum_{\{i,j\} \in E} c_e(\overline{x}_{ij} + \overline{x}_{ji}) \tag{23a}$$

Table 2: Association between dual variables and primal constraints

Primal constraints	Dual variables	Primal constraints	Dual variables
(15b)	z_W	(15c)	q
(15d)	v_j	(15e)	u_j
(15f)	y_{ij}		

$$
\begin{aligned}
\text{s.t.}\quad & c_{ij} \le c_{ik} + c_{jk} && \forall \{i,j\},\{i,k\},\{j,k\} \in E, && \text{(23b)}\\
& y_{ij} + v_j + \sum_{w \in W(ij)} z_w - c_{ij} = 0, && \forall i \in T, j \in T \setminus \{r\},\ x_{ij} > 0, && \text{(23c)}\\
& y_{ij} + v_j + 2u_j + \sum_{w \in W(i,j)} z_w - c_{ij} = 0, && \forall i \in T,\ j \in S,\ x_{ij} > 0, && \text{(23d)}\\
& y_{ji} + v_i - u_j + \sum_{w \in W(j,i)} z_w - c_{ji} = 0, && \forall i \in T \setminus \{r\}, j \in S,\ x_{ij} > 0, && \text{(23e)}\\
& y_{ij} + v_j + 2u_j - u_i + \sum_{w \in W(i,j)} z_w - c_{ij} = 0, && \forall i, j \in S,\ x_{ij} > 0, && \text{(23f)}\\
& z_W = 0, && \forall W \subset V \setminus \{r\},\ W \cap T \neq \emptyset\ x\left(\delta^-(W)\right) > 1, && \text{(23g)}\\
& v_j = 0, && \forall j \in V \setminus \{r\},\ x\left(\delta^-(j)\right) < 1, && \text{(23h)}\\
& u_j = 0, && \forall j \in S,\ 2x\left(\delta^-(v)\right) < x\left(\delta^+(v)\right), && \text{(23i)}\\
& y_{ij} = 0, && \forall (i,j) \in A,\ x_{ij} < 1, && \text{(23j)}\\
& c_{ij} \ge 0, && \forall \{i,j\} \in E. && \text{(23k)}
\end{aligned}
$$

4 Heuristic enumeration of nontrivial vertices

In this section, we present the theoretical results and algorithms used to enumerate vertices of the polytope $P_{\text{CM}}(n,t)$. We first introduce results linking polytopes of different dimensions, and then, relying upon these and other structural results, we present two different algorithms for vertices enumeration.

4.1 Avoiding redundancy

First, let us define a particular class of vertices that will be of interest for our results.

Definition 3. Let x be avertex of $P_{\text{CM}}(n,t)$. We will call x a *spanning vertex* if all of the nodes are part of the solution x, that is, $x(\delta^-(i)) + x(\delta^+(i)) > 0$ for all $i \in V$.

Note that Lemma 3 implies that every spanning vertex is connected. In an STP, Steiner nodes may or may not be part of an optimal solution. This holds for vertices of $P_{\text{CM}}(n,t)$, both integer and non-integer, that is, not all of the vertices are spanning vertices. Hence, we can consider whether a non-spanning vertex of $P_{\text{CM}}(n,t)$ can be seen as a spanning vertex of a polytope of a smaller dimension, and vice versa, that is, if a spanning vertex of $P_{\text{CM}}(n,t)$ can be seen as a vertex of a polytope of a larger dimension. The following results link vertices of $P_{\text{CM}}(n+1,t)$ with vertices of $P_{\text{CM}}(n,t)$ and vice versa. These results will be used in the enumeration of vertices to reduce the dimension of our search space by avoiding redundancy.

Lemma 6. *Let $x \in \mathbb{R}^{(n-1)\times n}$. Define $y \in \mathbb{R}^{n \times (n+1)}$ as*

$$
y_{ij} = \begin{cases} x_{ij}, & 1 \le i, j < n+1, \\ 0, & \text{otherwise.} \end{cases} \tag{24}
$$

Then, $x \in P_{CM}(n,t)$ if and only if $y \in P_{CM}(n+1,t)$.

Proof. Let $x \in P_{\text{CM}}(n,t)$. Note that y satisfies the domain constraints. Regarding Constraint (15b), we distinguish two cases. Let W be a set as described in (15b) for y. (a) If $n+1 \in W$, going from x to y adds the variables $x_{i,n+1}$ which are all zero so since x satisfies the constraint y satisfies it too. (b) If $n+1 \notin W$, going from x to y adds the variables $x_{n+1,j}$ which are all zero so since x satisfies the constraint y satisfies it too. Constraints (15c)—-(15d) are satisfied by y since x satisfies them, and we are only adding variables that take the value zero. Regarding Constraint (15e), if $j = n+1$, the constraint holds trivially since all the variables are zero. If $j \neq n+1$, going from x to y adds the variables $x_{i,n+1}, x_{n+1,j}$ which are all zero, so since x satisfies the constraint, y also satisfies it.

Let $y \in P_{\text{CM}}(n+1,t)$ of the form (24). Note that x satisfies the domain constraints. Let W be a set as described in (15b) for x. Let $\hat{W} := W \cup \{n+1\}$. $\hat{W}$ is a set for which y satisfies the correspondent constraint (15b). In the $\hat{W}$ constraint, the only variables that appear are the ones appearing in the W constraint plus the variables $x_{i,n+1}$ which are all zero. Since the $\hat{W}$ constraint is satisfied by y, the W constraint is satisfied by x. Constraints (15c)–(15d) are clearly satisfied by x since y satisfies them. Regarding Constraint (15e), passing from y to x removes the variables $x_{i,n+1}, x_{n+1,j}$ which are all zero, so since y satisfies the constraint, x also satisfies it. □

The following lemmas show how to identify vertices of $P_{\text{CM}}(n+1,t)$ with the ones of $P_{\text{CM}}(n,t)$ and vice versa.

Lemma 7. *Let x be a vertex of $P_{CM}(n,t)$. Then*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i,j \neq n+1, \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

is a vertex of $P_{CM}(n+1,t)$.

Proof. The idea of the proof is to show by contradiction that if y is not a vertex, that x cannot be a vertex as well. In detail, we have that $y \in P_{\text{CM}}(n+1,t)$ because of Lemma 6. Let $P_{\text{CM}}(n+1,t)_0$ be the subpolytope of $P_{\text{CM}}(n+1,t)$ defined as

$$P_{\text{CM}}(n+1,t)_0 := \{z \in P_{\text{CM}}(n+1,t) : \; z_{i,n+1} = z_{n+1,j} = 0, \; 1 \leq i,j \leq n\}.$$

Let

$$\begin{aligned} \pi \colon P_{\text{CM}}(n+1,t)_0 &\twoheadrightarrow P_{\text{CM}}(n,t) \\ (z_{ij})_{i,j} &\mapsto (z_{ij})_{i,j \neq n+1} \end{aligned}$$

be the projection considering the first n nodes. Note that $\pi(y) = x$ and that π is an injective map. Note also that $\text{Im}(\pi) \subset P_{\text{CM}}(n,t)$ because of Lemma 6. By contradiction, suppose that there exist $a, b \in P_{\text{CM}}(n+1,t)$ such that $a \neq b$, $y = \frac{1}{2}a + \frac{1}{2}b$. We have that

$$y_{i,n+1} = y_{n+1,j} = 0 = \frac{1}{2}(a_{i,n+1} + b_{i,n+1}) = \frac{1}{2}(a_{n+1,j} + b_{n+1,j}).$$

Since $a, b \in P_{\text{CM}}(n+1,t)$, we have that $a_{i,n+1}, b_{i,n+1}, a_{n+1,j}, b_{n+1,j} \geq 0$ and so $a_{i,n+1}, b_{i,n+1}, a_{n+1,j}, b_{n+1,j} = 0$. Thus, $a, b \in P_{\text{CM}}(n+1,t)_0$ and we can define $c := \pi(a)$, and $d := \pi(b)$, and we have that $c, d \in P_{\text{CM}}(n,t)$, $c \neq d$, $x = \frac{1}{2}c + \frac{1}{2}d$, a contradiction. □

Lemma 8. *Let $t < n$ and let y be a vertex of $P_{CM}(n,t)$ of the form*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i \neq n \neq j, \\ 0, & \text{else,} \end{cases} \tag{26}$$

for $n \in V \setminus T$. Then x is a vertex of $P_{CM}(n-1,t)$.

Proof. We have that $x \in P_{\text{CM}}(n-1,t)$ because of Lemma 6. Let

$$\begin{aligned} i \colon P_{\text{CM}}(n-1,t) &\hookrightarrow P_{\text{CM}}(n,t) \\ (z_{i,j})_{i \neq n \neq j} &\mapsto ((z_{i,j})_{i \neq n \neq j}, 0, \ldots, 0) \end{aligned}$$

be the trivial immersion and note that $i(x) = y$. Note also that $\text{Im}(i) \subset P_{\text{CM}}(n,t)$ because of Lemma 6. By contradiction, suppose there exist $c, d \in P_{\text{CM}}(n-1,t)$ such that $c \neq d$, $x = \frac{1}{2}c + \frac{1}{2}d$. If we define $a := i(c)$, and $b := i(d)$, we have that $a, b \in P_{\text{CM}}(n,t)$, $a \neq b$, $y = \frac{1}{2}a + \frac{1}{2}b$, and so we have a contradiction. □

Note that the result above still holds if we replace the node n with any node $k \in V \setminus T$.

Observation 5. Let π and i be the maps introduced in the proofs of Lemma 7 and Lemma 8, respectively. Note that π is an injective map and $i(P_{\text{CM}}(n,t)) \subset P_{\text{CM}}(n+1,t)_0$, thus we have that π is also a surjective map and so it is bijective. Moreover, π is linear and sends vertices in vertices. In particular $P_{\text{CM}}(n+1,t)_0 \cong P_{\text{CM}}(n,t)$, where the isomorphism is given by the map π. Note that π is a surjective map because given an element $x \in P_{\text{CM}}(n,t)$, we have that $\pi(i(x)) = x$, and we can map $i(x)$ through π because $i(P_{\text{CM}}(n,t)) \subset P_{\text{CM}}(n+1,t)_0$. This implies that, in the aim of evaluating vertices of our polytopes, it is sufficient to evaluate vertices of $P_{\text{CM}}(n,t)$ to get all of the vertices of $P_{\text{CM}}(m,t)$, for every $m = t, t+1, \dots, n$. Alternatively, we can evaluate only the spanning vertices of $P_{\text{CM}}(n,t)$ for every n, t, since every non-spanning vertex can be seen as a spanning vertex of a polytope of a smaller dimension, applying the lemmas above iteratively. Note that we are only interested in non-isomorphic vertices because isomorphic vertices have the same integrality gap, see Lemma 1. Note also that the results presented above hold for the DCUT and the SJ formulations. The proof can be done in almost the same way.

Observation 6. As we have seen, the trivial way to go from a vertex of $P_{\text{CM}}(n+1,t)$ to a vertex of $P_{\text{CM}}(n,t)$ is removing zeros, and the trivial way to go from a vertex of $P_{\text{CM}}(n,t)$ to a vertex of $P_{\text{CM}}(n+1,t)$ is adding zeros. As one would expect, the trivial way to go from a vertex of $P_{\text{CM}}(n+1,t+1)$ to a vertex of $P_{\text{CM}}(n,t)$ and vice versa is the "dual" procedure of the previous one, that is, adding or removing one 1. Note that this can be done in different ways. More precisely, the following procedures start with a vertex of $P_{\text{CM}}(n,t)$ and return a vertex of $P_{\text{CM}}(n+1,t+1)$.

(a) Add an edge of weight 1 between a node v of indegree 1 and the new added terminal, see for example Figure 2b $\rightarrow$ Figure 3a and Figure 2a $\rightarrow$ Figure 3b.

(b) Same as (a), but substituting the outflow of v with the outflow of the newly added terminal, see for example Figure 2b $\rightarrow$ Figure 3c.

(c) Add an edge of weight 1 between the newly added terminal and the root, then swap the role of these two nodes, see Figure 2a $\rightarrow$ Figure 3d.

Reversing these procedures, when possible, allows us to go from a $P_{\text{CM}}(n+1,t+1)$ to a vertex of $P_{\text{CM}}(n,t)$. The proofs are similar to the ones presented above. For all of the procedures above, it is clear that the generated vertices are not isomorphic to the ones we start from.

Note that the above procedures do not change the integrality gap, as shown by the lemma below.

Lemma 9. *Let $\epsilon \geq 0$, $c \in \mathbb{R}^n_{\geq \epsilon}$ a cost vector of a minimization ILP instance, and let $x \in \{0,1\}^n$ be the variables of the LP. Denote by $\bar{x}, \hat{x}$ an optimal integer solution and an optimal solution of the LP relaxation, respectively. Suppose an index k exists such that $\hat{x}_k = 1$. Then, the instance $\tilde{c}$ defined as*

$$\tilde{c}_j = \begin{cases} c_j & j \neq k, \\ \epsilon & j = k, \end{cases} \tag{27}$$

has a greater or equal integrality gap than the instance c. Moreover, $\hat{x}$ is an optimum for the LP relaxation of the instance $\tilde{c}$.

Proof. Let $P \subset [0,1]^n$ be the polytope defined by the LP relaxation. For the second part, suppose by contradiction that there exists $y \in P$ such that $\tilde{c}^\intercal y < \tilde{c}^\intercal \hat{x}$. This can be rewritten as

$$\tilde{c}^\intercal y = \sum_{i=1}^n \tilde{c}_i y_i = \epsilon y_k + \sum_{i \neq k}^n c_i y_i < \epsilon \hat{x}_k + \sum_{i \neq k}^n c_i \hat{x}_i.$$

Thus,

$$\begin{aligned} c^\intercal y = c_k y_k + \sum_{i \neq k}^n c_i y_i < c_k y_k + \epsilon(\hat{x}_k - y_k) + \sum_{i \neq k}^n c_i \hat{x}_i = \\ = \epsilon(\hat{x}_k - y_k) + c_k y_k - c_k \hat{x}_k + \sum_{i=1}^n c_i \hat{x}_i = \\ = (\epsilon - c_k)(\hat{x}_k - y_k) + c^\intercal \hat{x} = (\epsilon - c_k)(1 - y_k) + c^\intercal \hat{x} \leq c^\intercal \hat{x}, \end{aligned}$$

which contradicts the optimality of $\hat{x}$. For the first part, we have

$$\text{IG}(\tilde{c}) = \frac{\text{ILP}(\tilde{c})}{\text{LP}(\tilde{c})} = \frac{\text{ILP}(\tilde{c})}{\tilde{c}^\intercal \hat{x}} = \frac{\text{ILP}(\tilde{c})}{\text{LP}(c) - c_k + \epsilon}.$$

We have that $\text{ILP}(\tilde{c}) \leq \tilde{c}^\intercal \bar{x} \leq c^\intercal \bar{x} = \text{ILP}(c)$. We now want to prove that $\text{ILP}(\tilde{c}) \geq \text{ILP}(c) - c_k + \epsilon$. Suppose by contradiction that there exists $\bar{y} \in P$ integer such that $\tilde{c}^\intercal \bar{y} < c^\intercal \bar{x} - c_k + \epsilon$. But then

$$\begin{aligned} c^\intercal \bar{y} &= c_k \bar{y}_k + \sum_{i \neq k}^n c_i \bar{y}_i = (c_k - \epsilon)\bar{y}_k + \tilde{c}^\intercal \bar{y} \\ &< (c_k - \epsilon)\bar{y}_k + c^\intercal \bar{x} - c_k + \epsilon = (\bar{y}_k - 1)(c_k - \epsilon) + c^\intercal \bar{x} \leq c^\intercal \bar{x}, \end{aligned}$$

which contradicts the optimality of $\bar{x}$. We then have that $\text{ILP}(c) \geq \text{ILP}(\tilde{c}) \geq \text{ILP}(c) - c_k + \epsilon$, which implies

$$\frac{\text{ILP}(c)}{\text{LP}(c) - c_k + \epsilon} \geq \text{IG}(\tilde{c}) \geq \frac{\text{ILP}(c) - c_k + \epsilon}{\text{LP}(c) - c_k + \epsilon} \geq \frac{\text{ILP}(c)}{\text{LP}(c)} = \text{IG}(c)$$

where the last inequality holds because $\text{ILP}(c) \geq \text{LP}(c) \geq 0$, $\text{LP}(c) - c_k + \epsilon \geq 0$, $c_k \geq \epsilon$. □

Observation 7. Let $\hat{x}$ be a fractional vertex of $P_{\text{CM}}(n,t)$ such that $\hat{x}_{ij} = 1$. Because of Lemma 9, the maximum integrality gap is reached by minimizing the value of c_{ij}, making node i and node j collapse onto each other. This can be done by choosing a sequence of values of ϵ such that $\epsilon \to 0$. The study of the gap of this vertex is then equivalent to the study of the gap of a vertex of a smaller dimension. Note that, if we restrict the study to the metric case, even if (27) might not define a metric cost, the result still holds because of Lemma 5.

4.2 Two heuristics procedure for vertices enumeration

In the following, we state more properties of the CM formulation that permit the design of two different heuristic procedures, in particular, one general search and one dedicated to a specific class of vertices. We are only interested in spanning vertices (Observation 5).

4.2.1 The $\{1,2\}$-costs heuristic

The first procedure is based on the observation that, when looking for integer solutions of the CM formulation, it is enough to study only metric graphs with edge weights in the set $\{1,2\}$.

Theorem 1. *Let x be an integer point of $P_{CM}(n,t)$. Then, x is an optimal solution for the CM formulation with the metric cost $c_{ij} = 2 - (x_{ij} + x_{ji}) \in \{1,2\}$.*

Proof. Consider x and the STP instance given by the vector c defined in the statement. We want to prove that x is optimal. Let x' be the integer optimal solution for c and let s, s' be the number of Steiner nodes of x and x', respectively. Let us write $x_e = x_{ij} + x_{ji}$ and the same for x'. We will divide the proof into two cases. First, we will prove (i) that if $s' \geq s$, then necessarily $s' = s$ and $x = x'$. Then, we will prove (ii) that if $s' < s$ we get a contradiction.

(i) Since the optimal solution is a tree with t terminals and s and s' Steiner node, respectively, we have that

$$\sum_{e \in E} x_e = t + s - 1, \qquad \sum_{e \in E} x'_e = t + s' - 1.$$

Note that the definition of c implies that the cost is equal to 1 on the edges of the support graph of the solution and equal to 2 otherwise. Because of the definition of c, we have that

$$\sum_{e \in E} c_e x_e = t + s - 1.$$

Now let $I_0 = \{e : x_e = 0,\ x'_e = 1\}$, $I_1 = \{e : x_e = 1,\ x'_e = 0\}$, $I = \{e : x_e = x'_e\}$. We then have that

$$\sum_{e \in E} c_e x'_e = \sum_{e \in I_0} c_e x'_e + \sum_{e \in I_1} c_e x'_e + \sum_{e \in I} c_e x'_e = 2 \times |I_0| + |I| \geq |I_0| + |I| =$$
$$= t + s' - 1 \geq t + s - 1 = \sum_{e \in E} c_e x_e,$$

and they are equal if and only if $s' = s$ and $I_0 = \emptyset$, and since these two conditions imply $I_1 = \emptyset$, we have that $x = x'$.

(ii) Let $\mathcal{S}(x)$ and $\mathcal{S}(x')$ be the set of Steiner nodes of the solution x and x', respectively. Let $\mathcal{S} = \{s_1, \dots, s_k\} = \mathcal{S}(x) \setminus \mathcal{S}(x')$ and let z be the number of edges of the form $s_i s_j$. Note that $\mathcal{S} \neq \emptyset$, otherwise we would have $s' \geq s$. Now, x' is a tree with $s' + t$. Thanks to the hypothesis $s' < s$, we have $s' = s - k$, and hence x' is a tree with $s + t - k$ nodes, so $s + t - k - 1$ edges. On the other side, x has $s + t - 1$ edges, all of them of cost 1, while all of the other edges have cost 2. We now have to evaluate how many edges of cost 2 x' must have, given that it does not contain any node of the set $\mathcal{S}$. We have that c contains exactly $s + t - 1$ edges of cost 1, and the number of those edges that contain a node of $\mathcal{S}$ is $\left(\sum_{i=1}^{k} deg(s_i)\right) - z$.

Since we said that x' must contain $s+t-k-1$ edges, its cost is $E_1 + 2 \times E_2$, where E_1 is the number of edges of cost 1 and E_2 is the number of edges of cost 2, and we have that

$$E_1 \leq s+t-1-\left(\left(\sum_{i=1}^{k} deg(s_i)\right)-z\right), \quad E_2 = s+t-k-1-E_1,$$

and the minimum of $E_1 + 2 \times E_2$ is attained when E_1 is exactly equal to the rhs. The difference between the cost of x' and the cost of x, which is exactly $s+t-1$, is then at least

$$2\times\left(\left(\sum_{i=1}^{k} deg(s_i)\right)-k-z\right)-\left(\left(\sum_{i=1}^{k} deg(s_i)\right)-z\right)=$$
$$=\left(\sum_{i=1}^{k} deg(s_i)\right)-2k-z \geq 3k-2k-z \geq k-z \geq 1,$$

So, x' is not optimal, and we have a contradiction. Note that $deg(s_i) \geq 3$ because of Constraints (15d) and (15e), and $k-z \geq 1$ because the support graph associated to $\mathcal{S}$ as a subgraph of x is a forest since it is a subgraph of a CM solution, which is a tree.

□

Observation 8. Note that the generalization of Theorem 1 does not hold in general for the non-integer case, that is, if x is a non-integer point of $P_{\text{CM}}(n,t)$, then x is not necessarily an optimal solution for the CM formulation with the metric cost

$$c_{ij} = 2 - \mathbb{1}_{E_x}(\{i,j\}), \qquad E_x = \{e = \{i,j\} \in E \,:\, x_{ij} + x_{ji} > 0\}, \tag{28}$$

see, for example, the vertex depicted in Figure 4d. In this case, with the cost assignation (28), we have that the fractional vertex has a value of $11/2$ (multiply the number of edges by $1/2$), while the optimal value of the CM formulation for this instance is 5. Thus, the vertex shown in Figure 4d cannot be optimal for this instance. It still holds that the vertex mentioned above is an optimum of a metric graph where every edge weight is in the set $\{1,2\}$, namely setting the cost as in (28) but changing the cost of the two edges outflowing the root, setting them to 2 instead of 1. Note that in this case, the subgraph linked to edges with cost 1 is not connected, as the root represents a connected component.

The observation above, together with Theorem 1, lead us to formulate a heuristic search based on the generation of metric graphs with edge weights in the set $\{1,2\}$ and then solve the STP on those instances. The detailed procedure called OTC(n,t) as in One-Two-Costs is described in Algorithm 1.

Note that for computational reasons, we restricted our search to the generation of only connected graphs, and so to graphs with costs $\{1,2\}$ in which the subgraph regarding the edges of cost 1 spans all the nodes and is connected. We know this is a strong restriction, making the procedure unable to find some vertices, see Observation 8. Note also that we restrict our search to graphs $G=(V,E)$ with $n \leq |E| \leq n \cdot t - t^2$: the lower bound is given by the fact that we are only interested in non-integer vertices, and the upper bound was derived after a first set of computational experiments. In Section 5.2, we broadly discuss this choice.

4.2.2 Pure half-integer vertices

Guided by the literature, we narrow our research to a specific class of vertices, which is conjectured to exhibit the maximum integrality gap in other NP-Hard problems (see, e.g., the Symmetric Traveling Salesman Problem [BB08; BE07] and the Asymmetric Traveling Salesman Problem [Ell08]). In particular, we restrict our attention to vertices having their values in $\{0, \frac{1}{2}, 1\}$. Given a non-integer vertex x of $P_{\text{CM}}(n,t)$, we say that x is *half-integer* (HI) if $x_{ij} \in \{0, 1/2, 1\}\ \forall i,j \in V$ and we say that x is *pure half-integer* (PHI) if $x_{ij} \in \{0, 1/2\}\ \forall i,j \in V$. In the following section, we state and prove some properties of PHI vertices. The choice of focusing only on PHI vertices instead of HI vertices is motivated by Lemma 9 and Observation 7.

Lemma 10. *Let x be a pure half-integer solution of $P_{CM}(n,t)$, which is also a vertex of $P_{DCUT}(n,t)$ optimum for a metric cost. Then we have that $x_{ij} > 0 \Rightarrow x_{ji} = 0$.*

Proof. Since x is pure half-integer, we have $x_{ij} = 1/2$. Suppose by contradiction that $x_{ji} \neq 0$, and so by the same reasoning, $x_{ji} = 1/2$. Because of Lemma 3, we have that the set $\{i,j\}$ is not a connected component of x, namely, is not an isolated 2-cycle, and neither of the two nodes can be the root, as the root has inflow equal to 0 because of

Algorithm 1 $\{1,2\}$-costs vertices heuristic

procedure OTC(n,t)
$\mathbb{G} = \{G = (V, E) \mid G \text{ connected},\ |V| = n,\ n \le |E| \le n \cdot t - t^2\}$
$\mathbb{T} = \{T \mid T \subset \{1, \dots, n\},\ |T| = t\}$
for $G \in \mathbb{G}$ **do**
  for $T \in \mathbb{T}$ **do**
    for $r \in T$ **do**
      $G_{T,r} =$ node-colored graph with
        $\cdot\ G$ as its support graph
        $\cdot\ r$ colored as `root`
        $\cdot\ i$ colored as `terminal` $\forall i \in T \setminus \{r\}$
        $\cdot\ j$ colored as `steiner` $\forall i \notin T$
      if $G_{T,r} \not\cong H\ \forall H \in \mathfrak{G}$ **then**
        add $G_{T,r}$ to $\mathfrak{G}$
$\mathcal{V} = \emptyset$
for $G_{T,r} \in \mathfrak{G}$ **do**
  obtain the STP instance (G, T, r) from $G_{T,r}$ with $c_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \in G_{T,r} \\ 2 & \text{otherwise} \end{cases}$
  solve (15a)–(15e)
  if a solution x is found and it is a non-integer vertex of $P_{\text{CM}}(n,t)$ **then**
    add x to $\mathcal{V}$
return $\mathcal{V}$

Constraint (15c). Thus, there must exist a path from the root to the two nodes, and so there must exist an active arc going from a third node to one of the two nodes we are considering. Without loss of generality, let $x_{ki} > 0$, that implies $x_{ki} = 1/2$. Suppose $x_{ik} = 0$ since, otherwise, we can do the same reasoning for nodes $\{i, j, k\}$ and repeat it until we return to the root, which has no inflow. Now, we distinguish between two cases.

(a) No other inflow is present in j, that is, $x_{aj} = 0\ \forall a \neq j$. Note that this implies that j is not a terminal since it has an inflow of $1/2$. Then x is not optimum. Consider x' that is equal to x on all the arcs but the arc (j, i), and set $x'_{ji} = 0$. For any nonnegative c, $c^\intercal x' < c^\intercal x$. Note that x' is feasible for the DCUT. Constraint (1b) is clearly satisfied. Constraint (1c) could not be verified by x' only for a set W for which $I \in W,\ j \notin W$, because then it appears the only variables that differ from x. Let us take one of these sets and define $\bar{W} = W \cup \{j\}$. We can write

$$\begin{aligned}
\sum_{(a,b)\in\delta^-(W)} x'_{ab} &= \sum_{\substack{(a,b)\in\delta^-(W)\\(a,b)\neq(j,i)}} x'_{ab} + x'_{ji} = \sum_{(a,b)\in\delta^-(\bar{W})} x'_{ab} - \sum_{a\in V\setminus W} x'_{aj} + x'_{ji} = \\
&= \sum_{(a,b)\in\delta^-(\bar{W})} x_{ab} - \sum_{a\in V\setminus W} x_{aj} + x'_{ji} = \\
&= \sum_{(a,b)\in\delta^-(\bar{W})} x_{ab} + 0 + x'_{ji} \ge 1 + 0 = 1,
\end{aligned}$$

where the inequality holds because x is feasible and W is a valid set. So we have that x' is feasible even for the constraints regarding the sets W for which $i \in W,\ j \notin W$ and so it is feasible for the DCUT. If x' is feasible for the CM, the proof is concluded. If x' is not feasible for the CM formulation, it is because of Constraint (15e) because x' satisfied all of the other constraints since x is feasible for CM. Regarding Constraint (15e), if x' is not feasible for the CM anymore, it is because the outdegree of j in x was exactly two, namely $x_{ji} = \frac{1}{2}$ and there exist d such that $x_{jd} = \frac{1}{2}$. Hence, we can build x'' from x' by removing arc x'_{ij} and x'_{jd} from x' and by adding the arc x''_{id}, avoiding the detour in j. This solution is feasible for the CM, and it holds $c^\intercal x'' \le c^\intercal x'$, for the non-negativity and the triangle inequality. Hence, $c^\intercal x'' < c^\intercal x$, from the relation between x and x' already proved. Hence, we can conclude that if the only inflow of the (Steiner) node j is x_{ij}, x is neither optimal for the CM nor for the DCUT.

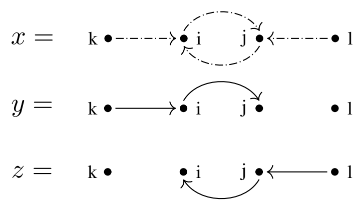
(b) The total inflow of j is 1, and so there exists l such that $x_{lj} = 1/2$. Suppose $x_{jl} = 0$ and suppose that both k and l have an inflow of 1. This will ensure the feasibility of the two points we are about to construct. Then x is not a vertex of $P_{\text{DCUT}}(n,t)$, because by setting

$$y_{ab} = \begin{cases} 0, & \text{if } a = l,\ b = j,\ \text{or } a = j,\ b = i, \\ 1, & \text{if } a = i,\ b = j,\ \text{or } a = k,\ b = i, \\ x_{ab}, & \text{else,} \end{cases}$$

$$z_{ab} = \begin{cases} 1, & \text{if } a = l,\ b = j,\ \text{or } a = j,\ b = i, \\ 0, & \text{if } a = i,\ b = j,\ \text{or } a = k,\ b = i, \\ x_{ab}, & \text{else,} \end{cases}$$

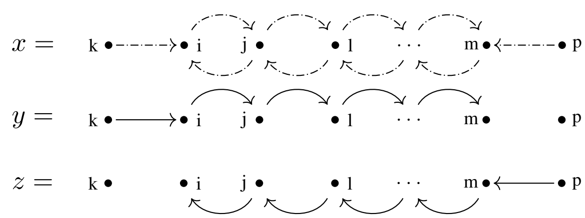
we have $y \neq z$, $x = \frac{1}{2}y + \frac{1}{2}z$, and $y, z \in P_{\text{DCUT}}(n,t)$ by an argument similar from the one above.

Visually, we can represent the three points as the following



where we draw only the interesting arcs. Note that dashed arcs represent a value of $1/2$ while full arcs represent a value of 1. If $x_{jl} \neq 0$, that is, $x_{jl} = 1/2$, then we can go backward until we find one node m such that there exists p for which $x_{pm} = 1/2,\ x_{mp} = 0$, and such a p exists because we can go back to the root with the same reasoning as above. Suppose that both k and p have an inflow of 1. We now do the same reasoning with y and z but consider the whole paths from p to i and from k to m instead of the paths from l to i and from k to j.

Visually, we can represent the three points as the following



where we draw only the interesting arcs. Note that dashed arcs represent a value of $1/2$ while full arcs represent a value of 1. If k or p do not have an inflow of 1, we can just go backward until we find a point with this property. If we do not find it, we go backward to the root. At this point, we can do the same reasoning with the paths as we did above.

□

We now focus on a particular type of PHI vertices, namely spanning vertices such that every Steiner node has indegree exactly one. We conjecture that every PHI spanning vertex has this property based on the following reasoning: First of all, because of Lemma 10, there are no loops of length 2, and so every edge can be oriented in only one way. Suppose there exists a Steiner node k such that $indeg(k) > 1$, and since the maximum inflow is 1 because of Constraint (15d) and we are dealing with pure half-integer solutions, we have that $indeg(k) = 2$. Then, regarding the MCF formulation, there exist $T_1, T_2 \subset T$, $T_1, T_2 \neq \emptyset$, and $i, j \in V$ such that $f_{ik}^{t_1} = f_{jk}^{t_2} = 1/2$, $\forall t_1 \in T_1, t_2 \in T_2$. We conjecture that is always possible to construct $y, z \in P_{\text{DCUT}}(n,t)$ such that $y \neq z$ and $x = \frac{1}{2}y + \frac{1}{2}z$, leading to a contradiction. In particular, y is derived by x by setting $f_{ik}^{t_1} = 1,\ f_{ik}^{t_2} = 0$, $\forall t_1 \in T_1, t_2 \in T_2$, and all the other variables are set accordingly to (16c), while z is derived by x by setting $f_{ik}^{t_1} = 0,\ f_{ik}^{t_2} = 1$, $\forall t_1 \in T_1, t_2 \in T_2$, and all the other variables are set accordingly to (16c).

We now derive some properties of these vertices that will be exploited in our heuristic search.

Lemma 11. *Let x be a pure half-integer solution of $P_{CM}(n,t)$, $t \geq 3$ that is also a vertex of $P_{DCUT}(n,t)$ and an optimum for a metric cost. Let x be a spanning vertex such that every Steiner node has indegree 1. Then, it holds that*

Algorithm 2 Pure half-integer vertices search

procedure PHI(n, t)
  $\mathbb{G} = \{G = (V, E) \mid G$ connected, $deg(i) \geq 2\, \forall i \in V,\ |V| = n,\ |E| = n + t - 2\}$
  di$\mathbb{G} = \emptyset$
  for $G = (V, E) \in \mathbb{G}$ **do**
    if $|\{i \in V \mid deg(i) = 2\}| \leq t$ **then**
      add to di$\mathbb{G}$ every non-isomorphic orientation of G s.t.
        · every edge can be oriented in only one way
        · every node has a maximum indegree of 2
  $\mathcal{V} = \emptyset$
  for di$G = (V, A) \in$ di$\mathbb{G}$ **do**
    if $|\{i \in V \mid indeg(i) = 0\}| = 1$ **then**
      if $|\{i \in V \mid indeg(i) = 1\}| = n - t$ **then**
        if $|\{i \in V \mid indeg(i) = 2\}| = t - 1$ **then**
          $x_{ij} = 1/2$ iff $(i, j) \in A$ is a solution of $P_{\text{CM}}(n, t)$ with
            · $\{r\} = \{i \in V \mid indeg(i) = 0\}$
            · $V \setminus T = \{i \in V \mid indeg(i) = 1\}$
            · $T \setminus \{r\} = \{i \in V \mid indeg(i) = 2\}$
          if x is a feasible vertex of $P_{\text{CM}}(n, t)$ **then**
            add x to $\mathcal{V}$
  return $\mathcal{V}$

- $|\{(i, j) \in A \mid x_{ij} > 0\}| = n + t - 2$,
- $3t - n - 4 \geq 0$.

Proof. For the first point, it suffices to count the incoming edges of each node. We have one incoming edge for each Steiner node and exactly two incoming edges for every terminal that is not the root since every terminal has an inflow exactly equal to one, and our edges have weights $1/2$. The total number of edges is $n - t + 2(t - 1) = n + t - 2$.

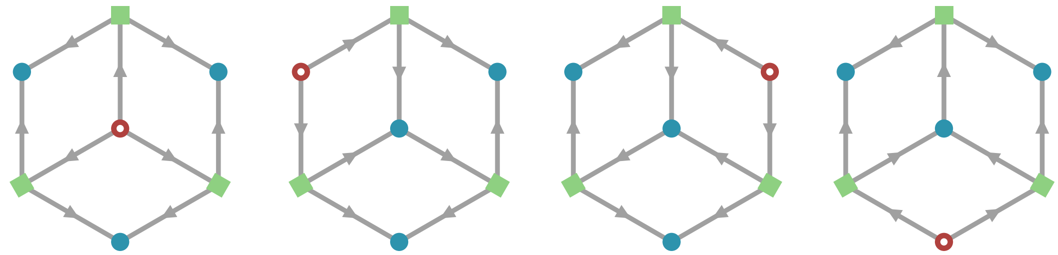
For the second point, because of Constraint (15b), we have that at least two edges exit from the root and at least two edges enter in every other terminal. Moreover, since in every Steiner node enters exactly one edge, at least two edges must come out. We then have that $2(n + t - 2) \geq 2t + 3(n - t)$ and so $3t - n - 4 \geq 0$. ☐

The properties stated above represent the core of the heuristic we now present. We generate all of the non-isomorphic connected undirected graphs such that every node is of degree at least 2 and with exactly $n + t - 2$ edges with the command `geng` of nauty [MP14]. For every generated graph, we generate all the non-isomorphic orientation of the edges, that can only be oriented in one way because of Lemma 10, and such that every node has a maximum indegree of 2 since we have Constraint (15d) and we are dealing with PHI solutions. This generation of digraphs can be done with the command `watercluster2` of nauty. The obtained digraph can be mapped into a spanning PHI vertex of $P_{\text{CM}}(n, t)$ for every feasible case. In particular, we have to check that: (i) There exist exactly $n - t$ nodes with in-degree 1 (Steiner nodes); (ii) There exists one node with in-degree 0 (root); (iii) There exist exactly $t - 1$ nodes of in-degree 2 (terminals). We filter all the generated graphs for these properties and then check if the remaining ones are vertices of $P_{\text{CM}}(n, t)$. This procedure called PHI(n, t), is illustrated in Algorithm 2.

Observation 9. Note how the PHI(n, t) can be generalized to vertex attaining values in the set $\{0, 1/m\}$ just by changing some values: the indegree of the terminal nodes must now be m, as well as the outdegree of the root, while the indegree of the Steiner nodes is again 1. This gives us a total number of edges of $n + (m - 1) \times t - m$. In addition, every node has degree at least $\min(3, m)$; if $m > 3$ the number of nodes with degree 3 is at most $n - t$; there must exist one node of indegree 0, $n - t$ nodes of indegree 1, and $t - 1$ nodes of indegree m.

5 Computational results

In this section, we aim to generate vertices of the P_{CM} polytope and, for any vertex, evaluate the maximum integrality gap that can be attained at that vertex. Recall that vertices of the P_{DCUT} polytope that are feasible for the CM formulation are also vertices of the P_{CM} polytope.



(a) First vertex. (b) Second vertex. (c) Third vertex. (d) Fourth vertex.

Figure 2: Fractional vertices of $(7, 4)$, all with integrality gap $10/9$. Hollow circle: root; Circles: Terminals; Square: Steiner node. Note that the second, third, and fourth vertex belong to the same class of isomorphism, while the first one belongs to another class.

Specifically, our approach involves: (i) running the two proposed heuristics for small values of n to produce vertices of P_{CM}, (ii) computing the maximum integrality gap associated with these vertices by solving the Gap problem, and (iii) extending our analysis beyond purely half-integer vertices.

Implementation details All the tests are executed on a desktop computer with a CPU 13th Gen Intel(R) Core(TM) i5-13600 and 16 GB of RAM. All the functions have been implemented in Python. For the optimization tasks, we use the commercial solver Gurobi 11.0 [Gur23].

5.1 Lower bounds for the integrality gap for $n \leq 10$

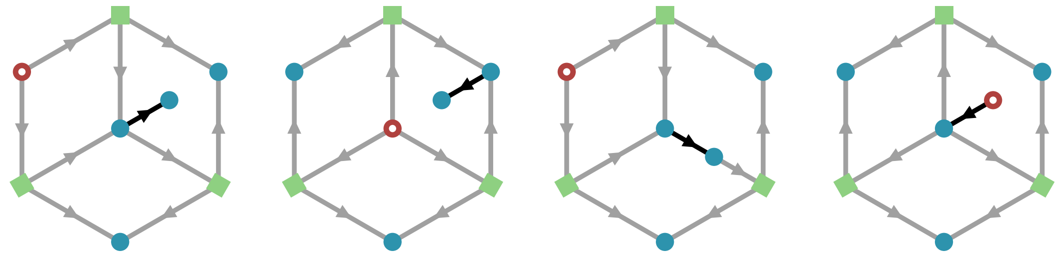
Our first set of algorithmic experiments aims at generating nontrivial vertices of P_{CM} having a large integrality gap. Table 3 and 4 present the lower bounds on the integrality gap and the number of nonintegral vertices we can compute with our two heuristics presented in Section 4. Recall that the vertices computed by the OTC procedures are filtered by isomorphism in post-processing. In the following paragraphs, we discuss the results for specific values of the number of vertices, starting from $n = 6$. Recall that results for $n \leq 5$ are presented in Table 1: we computed every vertex with Polymake, finding a maximum value of integrality gap equal to 1. Further details can be found in Appendix A.

$(n = 6)$ For $n = 6$, for any value of $t \leq 3 \leq n - 1$, the best lower bound we compute is always equal to 1. We conjecture that for $n = 6$, the CM and the DCUT formulation have a gap equal to 1.

$(n = 7)$ For $n = 7$, we compute four vertices attaining the gap of $\frac{10}{9}$ with the heuristic OTC; three of them belong to the same class of isomorphism. Figure 2 shows those four vertices, where Figures 2b, 2c, and 2d show the three isomorphic graphs. Moreover, these vertices are *pure half-integer*, e.g., $x_a \in \{0, \frac{1}{2}\}$. Although the directed support graph is the same for the four vertices, the arc orientation changes, and, in particular, the node labeled as "root" is different. Note that the PHI heuristic can only find two of the four vertices since it can find only one vertex for every class of isomorphism of node-colored edge-weighted directed graphs. On the contrary, the OTC heuristic may find more than one representative for the same class of isomorphism.

$(n = 8)$ The case $n = 8$ is more involved. The PHI does not find fractional vertices for the cases $t = 3, 4$. While the OTC does not find fractional vertices for $t = 3$, it finds again the vertices of $(7, 4)$ since it does not only find spanning vertices, as we already discussed. Both heuristics only find fractional vertices of integrality gap 1 for the case $t = 6$. For $t = 7$, only the PHI heuristic finds fractional vertices, all of them with an integrality gap of 1. The most interesting case is $t = 5$: the maximum integrality gap is depicted in Figure 4a while different values of integrality gap are depicted in Figure 4d and 4e. Note that the maximum integrality gap of this case for the PHI heuristic is $12/11$, while the maximum integrality gap for the OTC heuristic is again $10/9$: some of the vertices attaining this value are depicted in Figure 3. Note also how these vertices can be obtained from vertices of $(7, 4)$, as explained in Observation 6.

$(n = 9)$ For $n = 9$, we can only run the PHI heuristics, as the OTC heuristic runs out of memory. Note how no vertices for the case $t = 3, 4$ are found, accordingly to the second point of Lemma 11, while for the cases $t = 7, 8$ only vertices of integrality gap 1 are found. The maximum values of the integrality gap for $t = 5, 6$ are $10/9$ and $14/13$, as shown in Figure 4b and 4c, respectively. Different values of integrality gap are depicted in Figure 4. Notice how all the non-trivial values of integrality gaps found for $n \leq 9$ are of the form $\frac{2m}{2m-1}$, with $m = 5, 6, 7, 8, 9, 10, 11, 12$.



(a) First vertex. (b) Second vertex. (c) Third vertex. (d) Fourth vertex.

Figure 3: Fractional vertices of $(8, 5)$, all with integrality gap 10/9. Hollow circle: root. Circles: Terminals. Squares: Steiner nodes.

Table 3: PHI versus OTC heuristic. The third and sixth columns report the number of non-isomorphic vertices; the fourth and seventh columns report the maximum value of the gap obtained for these vertices; the fifth and eighth columns report the number of vertices attaining the maximum gap.

		PHI			OTC		
n	t	# vert.	max gap	# vert. max. gap	# vert.	max gap	# vert. max. gap
6	4	1	1	1	0	-	-
	5	7	1	7	0	-	-
7	4	2	10/9	2	11	10/9	2
	5	46	1	46	19	1	19
	6	71	1	71	8	1	8
8	4	0	-	-	19	10/9	2
	5	89	12/11	15	195	10/9	14
	6	1 070	1	1 070	239	1	239
	7	758	1	758	0	-	-

Table 4: Partial performances of the PHI heuristic for $n \geq 9$.

n	t	# vert.	max gap	# vert. max. gap
9	5	64	10/9	12
	6	4 389	14/13	200
	7	21 121	1	21 121
	8	8 987	1	8 987
10	5	15	10/9	7
	6	7 386	10/9	73
	7	155 120	16/15	2 653

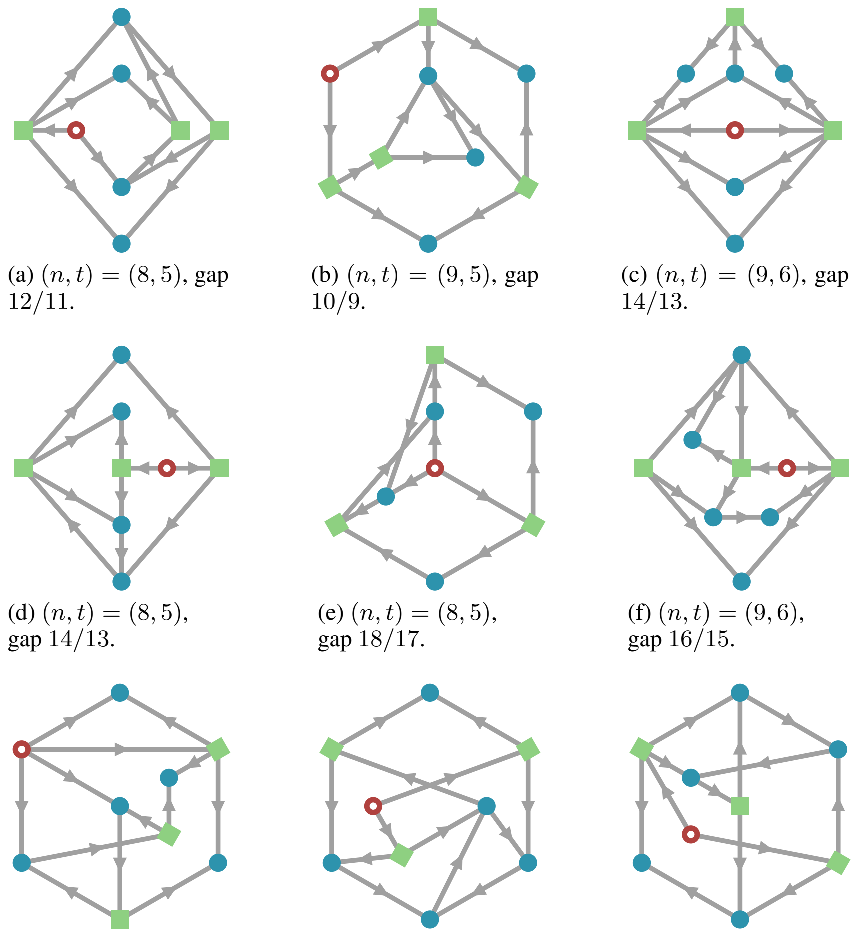
$(n \geq 10)$ For $n \geq 10$, even the PHI heuristic shows its limits. For $t \in \{8, 9\}$, the heuristic did not terminate within a timelit of 80 hours. The most interesting case we face is $t = 6$, where we found a vertex with an integrality gap of $19/18$. In this case, the value is of the form $\frac{2m+1}{2m}$, in contrast to what we found for $n \leq 9$.

For $n = 11$ and $n = 12$, we show that the cases $t \leq 5$ did not lead to any feasible PHI vertex. Tests with larger values of t were computationally currently untractable.

Note that no vertices with an integrality gap greater than 1 were found for the case $t = 3$. We focus on the theoretical aspects of this case in Appendix B.

5.2 A comparison between the two proposed heuristics

In this subsection, we discuss an in-depth comparison between the PHI and OTC heuristics. First, notice how neither of the two are exhaustive algorithms: at least one vertex can be found by the OTC heuristic but not by the PHI heuristic,



(a) $(n,t) = (8,5)$, gap $12/11$.
(b) $(n,t) = (9,5)$, gap $10/9$.
(c) $(n,t) = (9,6)$, gap $14/13$.
(d) $(n,t) = (8,5)$, gap $14/13$.
(e) $(n,t) = (8,5)$, gap $18/17$.
(f) $(n,t) = (9,6)$, gap $16/15$.
(g) $(n,t) = (9,6)$, gap $20/19$.
(h) $(n,t) = (9,6)$, gap $22/21$.
(i) $(n,t) = (9,6)$, gap $24/23$.

Figure 4: Fractional vertices of different gaps for different values of (n,t). The first three vertices attain the maximum gap for their respective value of (n,t).

and vice versa (see Figures 3a and 8). While the PHI heuristic is tailored for vertices with particular values, and so with a particular structure, the OTC is general enough to find different types of vertices; moreover, it remains an open question whether the heuristic becomes an exhaustive search by dropping the connectivity constraint.

Computationally, the OTC heuristic is highly demanding, even when limited to generating only connected graphs. For each generated graph, all possible assignments of the root, terminal nodes, and Steiner nodes must be considered, and an LP must be solved for every assignment. Moreover, there is no guarantee that the solution to the LP will be fractional; in fact, it may correspond to an equivalent integer solution. In addition, the OTC heuristic does not ensure that the generated solutions are non-isomorphic, necessitating a post-processing step to filter out isomorphic graphs based on node-colored edge-weighted graph isomorphism. The algorithm does not even guarantee finding spanning vertices. The PHI heuristic doesn't generate isomorphic graphs, and hence, every vertex generated belongs to a unique class of isomorphism. In addition, no LP needs to be solved since, given the orientation of the arcs, the role of every node is uniquely determined. Lastly, note that in the OTC heuristic, we have applied the extra bounds on the number of edges $n \cdot t - t^2$ derived after a first set of computational experiments. Without this hypothesis, OTC is untractable for $n \geq 8$. Table 3 results are obtained with this extra constraint. Note also that, even with the aforementioned bound on the number of edges, OTC is untractable for $n \geq 9$.

5.3 Beyond pure half-integer vertices

The computational results of the previous subsection show that the PHI heuristic is better than OTC in finding interesting vertices of the CM polytope. In addition, the PHI heuristic can be extended to enumerate all the vertices of the type $\{0, 1/m\}$, $m \in \mathbb{N}_{\geq 3}$. For example, an interesting case is $m = 4$, namely, when the vertices take value only in the set $\{0, 1/4\}$. Let us call these vertices *pure one-quarter* (POQ) vertices. In this case, our heuristic would work for each pair (n, t) as follows.

1. Generate all the non-isomorphic graphs having (i) every node of degree at least equal to 3, and (ii) exactly $n + 3t - 4$ arcs.
2. Filter the list of vertices computed at Step 1 by excluding all graphs with more than $n - t - 1$ nodes with degree 3. In POQ vertices, there are $n - t$ Steiner nodes that must have a minimum degree of 3, and the terminals have a minimum degree of 4.
3. For each oriented graph, we use `watercluster2` to get all possible orientations of edges, assuming that the maximum indegree must be equal to 4.
4. We filter out the list obtained at Step 3 and keep only the directed graphs having (i) exactly one node with in-degree 0 (the root), (ii) exactly $t - 1$ nodes with in-degree 4, (iii) exactly $n - t$ nodes with in-degree 1.

In [KPT11], the authors show that the integrality gap of the DCUT formulation is at least $\frac{8}{7}$ by exposing an instance leading to such a gap. The instance has 15 nodes and 8 terminals. The optimal vertex is of POQ type, and it originated from a personal communication between Martin Skutella and the authors of [KPT11]. This makes POQ vertices particularly relevant for our study. Figure 1 shows Skutella's graph. Note that solving the Gap function for the CM formulation leads to a gap equal to $\frac{8}{7}$. Hence, the maximum integrality gap for Skutella's graph is exactly $\frac{8}{7}$.

We defined a modified version of the PHI algorithm to find POQ vertices (for further details, see Appendix C), but we were not able to find any vertex with the above properties before the computation became intractable, that is $n \geq 8$.

6 Conclusion and future works

In this paper, we have studied the metric STP on graphs, focusing on computing lower bounds for the integrality gap for the DCUT and the CM formulations. We introduced a novel ILP formulation, the Complete Metric (CM) model, tailored for the complete metric Steiner tree problem. This formulation overcomes the limitations of the DCUT formulation in the metric case. For the CM formulation, we prove several structural properties of the polytope associated with its natural LP relaxation.

The core of our contribution presented in this paper is extending the Gap problem introduced in [BE05] for the symmetric TSP, to the metric Steiner tree problem. To speed up the search for suitable vertices, we designed two heuristics. Our heuristics outperform the exact method obtained as a natural extension of [BE05] to the Steiner problem and can generate nontrivial vertices for n up to 10. Note that exact methods got stuck already for $n = 6$.

We compare the performances of the two heuristics and their impact on providing insights into the exact value of the integrality gap. Although we cannot improve the bound of $\frac{10}{9}$ with $n \leq 10$, we find different structures of vertices leading to non-trivial gaps. By directly exploring vertices similar to those yielding the highest gaps for $n > 10$, we observed that these structures cannot be present for small values of n. Hence, we conjecture that with $n \leq 10$, the highest gap is $\frac{10}{9}$.

We retain that our study raises several interesting research questions. First, can an ad-hoc branching procedure be designed for the CM formulation? Second, can we improve the OTC heuristic by reducing the number of combinations we have to analyze without losing any of the outputs? Third, can we prove any further characterization of the vertices that reduce the effort for Polymake, similarly to what has been done in [BB08]? Lastly, can we enhance the design and implementation of the POQ heuristic to explore whether new lower bounds for the integrality gap are achievable for this type of vertices in higher dimensions?

We conclude this paper with three conjectures: First, we conjecture that for $t = 3$, the integrality gap is 1 for every n (See Appendix B for further details). Second, we conjecture that our OTC procedure, without the restriction on connectedness and the bound on the number of edges, is exhaustive and, hence, every vertex of $\mathrm{P}_{\mathrm{CM}}(n, t)$ can be obtained as an optimal solution of a $\{1, 2\}$-cost instance. Third, we conjecture that every spanning vertex x of $\mathrm{P}_{\mathrm{CM}}(n, t)$ with $x_{ij} \in \{0, 1/m\}$, $m \geq 2$, has an in-degree of 1 in every Steiner node, and hence our PHI is exhaustive for every pure half-integer spanning vertex.

Acknowledgments

Stefano Gualandi acknowledges the contribution of the National Recovery and Resilience Plan, Mission 4 Component 2 - Investment 1.4 - NATIONAL CENTER FOR HPC, BIG DATA AND QUANTUM COMPUTING (project code: CN_00000013) - funded by the European Union - NextGenerationEU. The work of Ambrogio Maria Bernardelli is supported by a Ph.D. scholarship funded under the"Programma Operativo Nazionale Ricerca e Innovazione" 2014–2020. The work of Eleonora Vercesi is supported by the Swiss National Science Foundation project n. 200021_212929 / 1 "Computational methods for integrality gaps analysis", project code: 36RAGAP

References

[AO21] Emmanuel Arrighi and Mateus de Oliveira Oliveira. "Three is enough for Steiner Trees". In: *19th International Symposium on Experimental Algorithms (SEA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2021.

[Ass+17] Benjamin Assarf, Ewgenij Gawrilow, Katrin Herr, Michael Joswig, Benjamin Lorenz, Andreas Paffenholz, and Thomas Rehn. "Computing convex hulls and counting integer points with Polymake". In: *Mathematical Programming Computation* 9 (2017), pp. 1–38.

[BB08] Genevieve Benoit and Sylvia Boyd. "Finding the exact integrality gap for small traveling salesman problems". In: *Mathematics of Operations Research* 33.4 (2008), pp. 921–931.

[BE05] Sylvia Boyd and Paul Elliott-Magwood. "Computing the integrality gap of the asymmetric travelling salesman problem". In: *Electronic Notes in Discrete Mathematics* 19 (2005), pp. 241–247.

[BE07] Sylvia Boyd and Paul Elliott-Magwood. "Structure of the extreme points of the subtour elimination polytope of the STSP". In: *Combinatorial Optimization and Discrete Algorithms* 23 (2007), pp. 33–47.

[BGT24] Jarosław Byrka, Fabrizio Grandoni, and Vera Traub. *The Bidirected Cut Relaxation for Steiner Tree has Integrality Gap Smaller than 2*. 2024. arXiv: `2407.19905 [cs.DS]`. URL: `https://arxiv.org/abs/2407.19905`.

[Byr+13] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. "Steiner tree approximation via iterative randomized rounding". In: *Journal of the ACM (JACM)* 60.1 (2013), pp. 1–33.

[CT01] Sunil Chopra and Chih-Yang Tsai. "Polyhedral approaches for the Steiner tree problem on graphs". In: *Steiner trees in industry*. Springer, 2001, pp. 175–201.

[Dij59] Edsger W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1 (1959), pp. 269–271. URL: `https://api.semanticscholar.org/CorpusID:123284777`.

[Edm+67] Jack Edmonds et al. "Optimum branchings". In: *Journal of Research of the National Bureau of Standards B* 71.4 (1967), pp. 233–240.

[Ell08] Paul Elliott-Magwood. "The integrality gap of the asymmetric travelling salesman problem". PhD thesis. University of Ottawa (Canada), 2008.

[Gam+17] Gerald Gamrath, Thorsten Koch, Stephen J Maher, Daniel Rehfeldt, and Yuji Shinano. "SCIP-Jack—a solver for STP and variants with parallelization extensions". In: *Mathematical Programming Computation* 9 (2017), pp. 231–296.

[GJ00] Ewgenij Gawrilow and Michael Joswig. "Polymake: a framework for analyzing convex polytopes". In: *Polytopes—combinatorics and computation*. Springer. 2000, pp. 43–73.

[GM93] Michel X Goemans and Young-Soo Myung. "A catalog of Steiner tree formulations". In: *Networks* 23.1 (1993), pp. 19–28.

[GMW97] Martin Grötschel, Alexander Martin, and Robert Weismantel. "The Steiner tree packing problem in VLSI design". In: *Mathematical Programming* 78 (1997), pp. 265–281.

[Gur23] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. URL: `https://www.gurobi.com`.

[HSV17] Stefan Hougardy, Jannik Silvanus, and Jens Vygen. "Dijkstra meets Steiner: a fast exact goal-oriented Steiner tree algorithm". In: *Mathematical Programming Computation* 9.2 (2017), pp. 135–202.

[Kar10] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

[KM98] Thorsten Koch and Alexander Martin. "Solving Steiner tree problems in graphs to optimality". In: *Networks: An International Journal* 32.3 (1998), pp. 207–232.

[KMV01] Thorsten Koch, Alexander Martin, and Stefan Voß. "SteinLib: An updated library on Steiner Tree problems in graphs". In: *Steiner Trees in Industry*. Ed. by Xiu Zhen Cheng and Ding-Zhu Du. Boston, MA: Springer US, 2001, pp. 285–325. ISBN: 978-1-4613-0255-1. DOI: `10.1007/978-1-4613-0255-1_9`. URL: `https://doi.org/10.1007/978-1-4613-0255-1_9`.

[KPT11] Jochen Könemann, David Pritchard, and Kunlun Tan. "A partition-based relaxation for Steiner trees". en. In: *Mathematical Programming* 127.2 (Apr. 2011), pp. 345–370. ISSN: 1436-4646. DOI: 10.1007/s10107-009-0289-2. URL: https://doi.org/10.1007/s10107-009-0289-2 (visited on 05/14/2024).

[Lju21] Ivana Ljubić. "Solving Steiner trees: Recent advances, challenges, and perspectives". In: *Networks* 77.2 (2021), pp. 177–204.

[MP14] Brendan D. McKay and Adolfo Piperno. "Practical graph isomorphism, II". In: *Journal of Symbolic Computation* 60 (2014), pp. 94–112. DOI: 10.1016/j.jsc.2013.09.003. URL: https://doi.org/10.1016/j.jsc.2013.09.003.

[Pey07] Sven Peyer. "Shortest paths and Steiner trees in VLSI routing". PhD thesis. Universitäts-und Landesbibliothek Bonn, 2007.

[Pri57] Robert Clay Prim. "Shortest connection networks and some generalizations". In: *The Bell System Technical Journal* 36.6 (1957), pp. 1389–1401.

[Vic20] Robert Vicari. *Simplex based Steiner tree instances yield large integrality gaps for the bidirected cut relaxation*. 2020. arXiv: 2002.07912 [cs.DS]. URL: https://arxiv.org/abs/2002.07912.

A Enumerating vertices with Polymake

A.1 Enumerating vertices with Polymake and weakness of the DCUT formulation

As discussed in the previous section, we aim to solve the Gap problem on every vertex. Hence, we need an exhaustive list of vertices of the polytope $P_{\text{DCUT}}(n,t)$ for each $n \geq 3$, for each $3 \leq t \leq n-1$. We use the software Polymake [GJ00], designed for managing polytope and polyhedron. We implement the Gap function in Python, using the commercial solver Gurobi 11.0 [Gur23] to model and solve the Gap ILP model.

On every vertex, we solve the Gap problem to get the maximum possible value of the integrality gap associated with that vertex. Table 5 reports our computational results. From these results, we can draw several conclusions. First, Polymake can only exhaustively generate vertices for n up to 5. Second, for all these cases, the value of the integrality gap is exactly 1. For larger values of n, the enumeration becomes computationally untractable. Furthermore, by running the Gap problem on many vertices of the DCUT formulation, we observe that the problem turns out to be infeasible. By analyzing the minimum infeasibility set, we observe that many vertices of the DCUT formulation are incompatible with the triangle inequality of the cost vector c nor with its non-negativity. We tackle both issues in the following sections by (a) introducing a novel formulation tailored for the metric case and (b) designing two heuristic algorithms for enumerating nontrivial vertices.

Table 5: Number of feasible and optimal vertices for P_{DCUT} and P_{CM}. The column "time" reports the time of the generation in seconds, while the column "gap" reports the maximum gap obtained for the optimal vertices. While the DCUT polytope has several (feasible) vertices that cannot be optimal for any metric cost, the CM polytope does not suffer this issue (and it implicitly reduces the number of isomorphic vertices).

		P_{DCUT}				P_{CM}			
n	t	time	feasible	optimal	gap	time	feasible	optimal	gap
4	3	0.04	256	70	1.00	0.73	4	4	1.00
5	3	4563.57	28 345	3 655	1.00	44.62	5	5	1.00
5	4	2798.17	24 297	3 645	1.00	37.01	44	44	1.00

A.2 Enumerating vertices with Polymake for the CM formulation

We enumerate all the vertices of the CM formulation using the software Polymake [Ass+17]. Recalling what we have done in Section A.1, we compute the gap value for each vertex by implementing the model (23) using Gurobi 11.0.0 [Gur23].

Table 5 reports our computational results. First, we can observe that the number of vertices generated is smaller than the number of vertices of the DCUT formulation, and all of them are feasible. As expected, the integrality gap is 1 (Note that it must be a lower bound w.r.t the one of the DCUT formulation, which was 1). Note also that, even in this case, Polymake cannot generate vertices for $n \geq 6$. These limited results motivate the design of the two heuristic algorithms to generate nontrivial vertices introduced in the next section.

B The case with three terminal nodes

As we already mentioned, while the cases $t = 2$ and $t = n$ are trivial in terms of integrality gap, fractional vertices with an integrality gap greater than 1 exist for all of the formulations we have presented, starting from the case $t = 4$. In this section, we discuss the case of the STP with three terminals, proving the characterization of integer solutions and conjecturing the form of non-integer ones.

First, we define a class of graphs that will be useful for our goals and prove some additional characteristics of this class.

Definition 4 (Tristar)**.** A *tristar* is a tree with at least three nodes and at most three leaves.

Lemma 12 (Tristar characterization)**.** *A tristar with n nodes has either*

- *three leaves, one node of degree* 3 *and the remaining nodes of degree* 2*, or*
- *two leaves, and the remaining nodes of degree* 2*.*

Proof. Let $G = (V, E)$ be a tristar with n nodes and $t = |T| = |\{w_1, \dots, w_t\}|$ leaves. In particular, it is a tree, so the following equation holds true

$$2(n-1) = 2|E| = \sum_{v \in V} \deg(v).$$

Breaking the summation over V into two disjoint subsets we obtain

$$\begin{aligned} 2(n-1) = \sum_{v \in V} \deg(v) &= \sum_{v \in T} \deg(v) + \sum_{v \in V \setminus T} \deg(v) = \\ &= t + 2|V \setminus T| + \sum_{v \in V \setminus T} (\deg(v) - 2) = \\ &= t + 2(n-t) + \sum_{v \in V \setminus T} (\deg(v) - 2). \end{aligned}$$

Rearranging the terms we obtain

$$\sum_{v \in V \setminus T} (\deg(v) - 2) = t - 2.$$

Note that this also holds for any tree. Since $\deg(v) \geq 2$ for every $v \in V \setminus T$, a tristar with two leaves has two nodes of degree 1 (the two leaves) and $n-2$ nodes of degree 2, while a tristar with 3 leaves has exactly one node of degree 3, three nodes of degree 1 (the three leaves), and the remaining $n-4$ nodes of degree 2. $\square$

In the following theorem, we prove that the support graph of an optimal solution of the DCUT formulation for a metric (non-necessarily complete) connected graph with three terminal nodes is tristar.

Theorem 2. *The support graph of an optimal solution $\mathcal{T}$ of the DCUT of a metric graph with three terminal nodes is a tristar that has a subset of the set of terminal nodes as the set of leaves. In particular there exists a node $\mathtt{c} \in V$ such that the optimal solution $\mathcal{T} \supseteq T = \{u_1, v_1, w_1\}$ is the union of one of the shortest-$(u_1, \mathtt{c})$-path together with one of the shortest-$(\mathtt{c}, v_1)$-path and one of the shortest-$(\mathtt{c}, w_1)$-path., oriented accordingly with the choice of the root.*

Proof. Since the costs are positive, by optimality arguments, we have that $\mathcal{T}$ is a tree, and since $T \subset \mathcal{T}$, we have that $|\mathcal{T}| \geq 3$. By contradiction, assume that $\mathcal{T}$ has more than 3 leaves. Then a leaf $v \in \mathcal{T}$ exists such as $v \notin T$. Since the degree of v is 1, we can remove the only edge of $\mathcal{T}$ connected to v and v itself to obtain a new tree $\mathcal{T}'$. We have that $T \subset \mathcal{T}'$ and since we removed an edge with a positive cost, $\mathcal{T}$ was not an optimal solution of the DCUT, which is a contradiction.

We proved that $\mathcal{T}$ is a tristar. If a node of degree three exists, let us denote it with $\mathtt{c}$. If such a node does not exist, it means that $\mathcal{T}$ has only two nodes of degree one, and so one of the three terminal nodes has degree two: let us denote it with $\mathtt{c}$. Note that, given any tristar with the set of leaves being a subset of the set of terminal nodes, substituting any path between node $\mathtt{c}$ and one of the terminal nodes with one of the shortest paths between $\mathtt{c}$ and that terminal node gives us a solution with a less or equal cost. Note also that when $\mathtt{c}$ is one of the terminal nodes, the shortest path between that node and $\mathtt{c}$ is the empty set. $\square$

Note that such characterization has already been observed, without a formal proof, in [AO21]. Note also that we are only using that the costs are positive, the triangle inequality needs not to hold. Furthermore, the graph does not need to be complete either; it suffices that all the terminal nodes belong to the same connected component, which is a necessary hypothesis for the existence of a solution.

We can now characterize the integer solutions of the STP with three terminals for complete metric graphs.

Corollary 1. *In a complete metric graph $G = (V, E)$, there exists an optimal solution for the DCUT with $T = \{r, t_1, t_2\}$ of one of the two following form*

- $x_{\{r,t_i\}} = x_{\{t_i,t_j\}} = 1$*, with* (i, j) *a permutation of* $(1, 2)$*, or*
- $x_{\{r,\mathtt{c}\}} = x_{\{\mathtt{c},t_1\}} = x_{\{\mathtt{c},t_2\}} = 1$ *for some* $\mathtt{c} \in V \setminus T$*,*

where we are considering $x_{kl} = 0$ if not specified otherwise.

Proof. Since we are in a metric graph, one of the shortest paths between two nodes u and v is given by the edge $\{u, v\}$, which exists because the graph is complete. Thus, a tristar of minimum cost has either 2 or 3 edges, oriented as in the thesis. $\square$

In addition to these theoretical results, we conducted several numerical tests, and we could not produce a fractional point of $P_{\text{DCUT}}(n,3)$ with an integrality gap greater than one, nor could we find a non-integer optimal solution. This led us to formulate two different conjectures.

Conjecture 1. *Any vertex of $P_{DCUT}(n,3)$ optimum for a metric cost is an integer orientation of a tristar.*

Conjecture 2. *Given a metric graph, there exists an optimal solution which is an integer orientation of a tristar.*

Note in particular that Conjecture 1 implies Conjecture 2. Note also that the conjectures cannot be proven using total unimodularity of the constraint matrix because even if for the cases $(n,t) = (2,2),(3,2),(3,3)$ the constraint matrix is totally unimodular, it is not true for $(n,t) = (4,3),(5,t)$ with $t \le n$. The conjectures above are based not only on numerical tests but also on the fact that any integer solution is the union of two or three disjoint shortest paths, and the DCUT formulation for the shortest path, that is, the case $t = 2$, is integral. Note that there exist solutions of the DCUT with more than three terminal nodes, which are not disjoint unions of the shortest path from one node to the terminals. For example, let G be the complete graph with six nodes $V = \{1,2,3,4,5,6\}$ and let $T = \{1,2,3,4\}$, $r = 1$. If $c_{1,5} = c_{2,5} = c_{3,6} = c_{4,6} = 1$, $c_{5,6} = 1.1$, while all the other costs are equal to 2, the optimal solution x is given by $x_{1,5} = x_{5,2} = x_{6,3} = x_{6,4} = x_{5,6} = 1$ while all the other variables are equal to 0, and it cannot be seen as union of shortest path, even non necessarily disjoint.

C Pure one-quarter algorithm

We defined Algorithm 3: a modified version of the PHI algorithm to find POQ vertices exploiting the properties described in Subsection 5.3.

Algorithm 3 Pure one-quarter vertices search

procedure POQ(n,t)
$\mathbb{G} = \{G = (V,E) \mid G \text{ connected},\ deg(i) \ge 3\,\forall i \in V,\ |V| = n,\ |E| = n + 3t - 4\}$
di$\mathbb{G} = \emptyset$
for $G = (V,E) \in \mathbb{G}$ **do**
 if $|\{i \in V \mid deg(i) = 3\}| \le n - t$ **then**
  add to di$\mathbb{G}$ every non-isomorphic orientation of G s.t.
    · every edge can be oriented in only one way
    · every node has a maximum indegree of 4
$\mathcal{V} = \emptyset$
for di$G = (V,A) \in$ di$\mathbb{G}$ **do**
 if $|\{i \in V \mid indeg(i) = 0\}| = 1$ **then**
  if $|\{i \in V \mid indeg(i) = 1\}| = n - t$ **then**
   if $|\{i \in V \mid indeg(i) = 4\}| = t - 1$ **then**
    $x_{ij} = 1/4$ iff $(i,j) \in A$ is a solution of $P_{\text{CM}}(n,t)$ with
      · $\{r\} = \{i \in V \mid indeg(i) = 0\}$
      · $V \setminus T = \{i \in V \mid indeg(i) = 1\}$
      · $T \setminus \{r\} = \{i \in V \mid indeg(i) = 4\}$
    if x is a feasible vertex of $P_{\text{CM}}(n,t)$ **then**
     add x to $\mathcal{V}$
return $\mathcal{V}$