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@idsia.ch

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

May 22, 2024

Abstract

In this work, we compute the lower bound of the integrality gap of the Metric Steiner Tree Problem (MSTP) on a graph for some small values of number of nodes and terminals. After debating about some limitations of the most used formulation for the Steiner Tree Problem, namely the Bidirected Cut Formulation, we introduce a novel formulation, that we named Complete Metric formulation, tailored for the metric case. We prove some interesting properties of this formulation and characterize some types of vertices. Finally, we define a linear program (LP) by adapting a method already used in the context of the Travelling Salesman Problem. This LP takes as input a vertex of the polytope of the CM relaxation and provides an MSTP instance such that (a) the optimal solution is precisely that vertex and (b) among all of the instances having that vertex as its optimal solution, the selected instance is the one having the highest integrality gap. We propose two heuristics for generating vertices to provide inputs for our procedure. In conclusion, we raise several conjectures and open questions.

Keywords: Steiner Tree · Integrality Gap · Combinatorial Optimization.

[*]Department of Mathematics "Felice Casorati", University of Pavia (Italy)
[†]Istituto Dalle Molle di studi sull'intelligenza artificiale (IDSIA USI-SUPSI) (Switzerland)
[‡]Faculty of Informatics, Università della Svizzera italiana, (Switzerland)

1 Introduction

Given an undirected, edge-weighted, connected graph $G = (V, E)$ with n nodes and cost c_{ij} on each edge $\{i, j\} \in E,\ i, j \in V$, and a subset of nodes $T \subset V$ of cardinality at least 2, the Steiner Tree Problem (STP) involves finding the minimum-cost tree that spans T. The STP is generally modeled and solved via integer linear programming, with many diverse publications released over the years. For a survey, see [GM93]. Among them, the *Directed cut formulation* has attracted much attention, as has an exceptional empirical performance [Lju21]. The core of the formulation consists in replacing each undirected edge $\{i, j\}$ with two arcs (i, j) and (j, i) and introducing a decision variable x_{ij} for each arc. For a given root node $r \in T$, the formulation is presented below

$$\min_{\mathbf{x}\in\{0,1\}^{2\times|E|}} \quad \sum_{\{i,j\}\in E} c_e(x_{ij} + x_{ji}) \tag{1a}$$

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

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

$$x_{ij} \in \{0, 1\}, \tag{1d}$$

where $\delta^-(W) := \{(i, j) \mid i \notin W,\ \ j \in W\}$. This model can be relaxed by replacing constrain (1d) with

$$0 \leq x_{ij} \leq 1. \tag{2}$$

We will abbreviate the ILP with DCUT and the relaxed version RDCUT. The optimal value of the DCUT formulation on a graph G with a set of terminals T will be denoted with $\mathrm{DCUT}(G, T)$, and similarly, we will use $\mathrm{DCUT}(G, T)$.

Note that, despite the exponential number of constraints (1c), it is well known that RDCUT can be optimized in polynomial time, as the oracle separating such constraints works in polynomial time, thanks to the max-flow-min-cut theorem [DF55]. To do so, we check for every $t \in T \setminus \{v\}$ if the minimal (r, t)-cut is less than one. If so, a violated cut inequality is found, otherwise, there is none and we can prove optimality. This separation routine has been improved through the years, e.g., by using faster algorithms for finding the minimum cut in a directed graph, such as [HO94].

On another hand, The STP is NP-Hard, and the corresponding decision problem is NP-Complete [Kar10]. The two well-known polynomial-time solvable cases are the *shortest path* ($|T| = 2$) and the *minimum spanning tree* ($|T| = n$). The best-known polynomial-time algorithm for the STS guarantees an approximation ratio of 1.39 [BGRS13]. Improving this bond is still an open problem.

In this framework, it becomes of interest the study of the *integrality gap*, which is the supremum among all the instances of the ratio between DCUT and RDCUT, namely

$$\alpha := \sup_{G=(V,E),\ T\subset V} \frac{\mathrm{DCUT}(G, T)}{\mathrm{RDCUT}(G, T)}. \tag{3}$$

The value α is unknown, and it has been proved to be between $\frac{36}{31} \cong 1.161$ [BGRS13] and 2. [GW95] The integrality gap is believed to be higher than $\frac{36}{31}$. Recall that, proving that $\alpha < 1.39$ would lead to a better approximation algorithm with respect to the state of art.

In this work, we restrict our attention on the *pseudo-metric* (pm) case, where (i) $c_{ij} \geq 0$, (ii) $c_{ij} = c_{ji}$, (iii) $c_{ij} \leq c_{ik} + c_{jk}$ and (iv) if $i = j$, then $c_{ij} = 0$. Note that if we add (v) if $c_{ij} = 0$, then $i = j$, we are in the metric case. We decided to restrict our attention to the pm case, as we can do it without loss of generality for the study of the integrality gap (see Section 2). Furthermore, some polynomial time approximation algorithms build on the metric closure (See, e.g.[KMB81]).

In this work, we analyze the integrality gap of the DCUT formulation for a fixed value of n and $t := |T|$. In Section 2, we show how to adapt a method already used for the study of the integrality gap in the context of the Travelling Salesman Problem [BB08] to the DCUT formulation. This methodology relies upon the vertices of the polytope of the RDCUT formulation. A close look at these vertices allows us to detail some limits of the DCUT for the complete metric case. In Section 3, we introduce the Complete Metric (CM) formulation and prove some interesting properties of this formulation. We adapt the methodology of [BB08] even to this case, showing the advantages of this formulation. In Section 4 we attack the problem of having vertices as input of our procedure. We observe that the exhaustive enumeration of vertices is intractable for $n \geq 6$. Hence, after demonstrating several properties of the vertices of the RCM polytope, we present two heuristic procedures for generating them. In Section 5, we present lower bounds for the CM formulation for $n \leq 10$. Additionally, we will compare the two heuristics we introduced, highlighting the strengths and weaknesses of each. Finally, we will provide insight into the properties that instances with high integrality gaps may have. Lastly, in the Appendix, we show some properties of the polytope of RCM when $|T| = 3$.

The remaining part of this section is devoted to fixing notations and abbreviations, even though some notations will be fixed from time to time in the paper.

Let F any integer linear programming formulation, then RF will denote the linear relaxation, namely the same integer constraints with the variable allowed to be continuous (real numbers). Through this manuscript, we will denote an STP instance with cost c and set of terminal T as STP(c, T). We will also denote with S the set of the so-called *Steiner nodes*, namely $V \setminus T$.

1.1 Related literature

The integrality gap for the DCUT formulation has been widely study. [KPT11] proved that the lower bound is at least $\frac{8}{7}$ showing that there exists one instance with precisely this gap (Skutella's graph). Later on, [BGRS13] improved this bound by showing a recursive family of instances, depending on a parameter p, having an integrality gap asymptotically tending to $\frac{36}{31}$.

2 Integrality gap for fixed n in the DCUT formulation

We present the general framework of the strategy we use to compute the gap, which is the same approach adopted in [BB08] and [EM08] adapted to the STP.

Let $P_{\text{DCUT}}(n, T)$ the polytope defined by Constraints (1b),(1c), (2). Consider the complete graph $K_n(c) = (V, E)$ having n nodes and $T \subset V$, and a cost function c on edges. Consider the quantities

$$\alpha_{c,T} := \frac{\text{DCUT}(K_n(c), T)}{\text{RDCUT}(K_n(c), T)} \tag{4}$$

$$\alpha_{n,t} := \sup_{c \text{ pm},\ T \subset V,\ |T|=t} \alpha_{c,T}. \tag{5}$$

The first one is the integrality gap of a given instance of pm STP, while the second one is the maximum integrality gap, once fixed both the cardinality of T and the number of nodes. Clearly,

$$\alpha = \sup_{n,t} \alpha_{n,t}. \tag{6}$$

As we have already mentioned, the case $t = 2, n$ can be solved in polynomial time. However, they may not have an integral formulation when coming to the DCUT formulation. In practice, for $t = n$ the polyhedron is integral [E+67], while [GM93] shows the same result for $t = 2$. However, the exact value for different values of t is still unknown. To compute it, we proceed as previously done in [BB08, EM08]. As already observed in [BB08], for a particular STP problem, if we divide all the costs $c_{ij}, i, j \in V$ of an instance $\text{STP}(c, T)$ for the optimal value $\text{DCUT}(K_n(c), T)$, we obtain another instance $\text{STP}(c', T)$, having an optimal value $\text{DCUT}(K_n(c'), T) = 1$ but the same set of optimal solutions. Hence, one can write

$$\alpha_{n,t} := \sup_{\substack{c \text{ pm},\ T \subset V,\ |T|=t, \\ \text{DCUT}(K_n(c),T)=1}} \frac{1}{\text{RDCUT}(K_n(c), T)},$$

that in turn becomes

$$\frac{1}{\alpha_{n,t}} := \inf_{\substack{c \text{ pm},\ T \subset V,\ |T|=t, \\ \text{DCUT}(K_n(c),T)=1}} \text{RDCUT}(K_n(c), T). \tag{7}$$

Note that, for the integrality gap, the choice of the terminals is irrelevant and the only thing that matters is the *number* of terminal. To make it clearer, consider an instance $\text{STP}(c, T)$ where $|T| = t$. This can be re-mapped to $\text{STP}(c, \{1, \ldots, t\})$ trough a node-colored-edge-weighted graph isomorphism. More formally,

Definition 1 (Graph isomorphisms)**.** Let $G = (V, E)$, $H = (V', E')$ two undirected non-node-colored non-edge-weighted graphs, with $|V| = |V'| = n$. The two graphs are said to be *isomorphic*, and we will write it as $G \cong H$, if there exists a bijection $\sigma : V \to V'$ such that

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

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

$$w(\{i,j\}) = w'(\{\sigma(i), \sigma(j)\}) \quad \forall \{i,j\} \in E, \tag{9}$$

for the graphs to be isomorphic. This definition naturally extends to the case of directed graphs by simply taking into consideration arcs instead of edges. If the two graphs are node-colored graphs, with $\mathsf{c} : V \to C$ and $\mathsf{c}' : V' \to C'$ being the color functions of G and H respectively, it must also hold that

$$\mathsf{c}(i) = \mathsf{c}(j) \iff \mathsf{c}'(\sigma(i)) = \mathsf{c}'(\sigma(j)) \quad \forall i,j \in V. \tag{10}$$

Each instance of the STP naturally leads to and edge-weighted node-colored graph, where the colors are three: one for the root node, one for the terminal minus the root, and one for the potential Steiner nodes, while the edge-weight function is represented by the cost c.

Equation (7) implicitly states that the integrality gap does not depend on the set T, but solely on its cardinality. This observation can be more formally expressed in the following Lemma, making use of the definitions above.

Lemma 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*

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

Proof. First of all notice that $\sigma(K_n(c))$ is a complete metric graph. Now, let x be an optimal solution of $\text{DCUT}(K_n(c), T)$ and let y be an optimal solution for $\text{RDCUT}(K_n(c), T)$. Construct $x' = \tilde{\sigma}(x)$ and $y' = \tilde{\sigma}(y)$, where $\tilde{\sigma}$ is the same isomorphism but seen as a isomorphism between directed graphs, using the arcs of x and y. We have that x' and y' are feasible solution of $\text{DCUT}(\sigma(K_n(c)), \sigma(T))$ and $\text{RDCUT}(\sigma(K_n(c)), \sigma(T))$, respectively, and they share the same optimal value of x and y. Thus,

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

We now just need the reverse inequality to prove equality. To prove so, simply do the same reasoning by considering the isomorphism $\tau = \sigma^{-1} : \sigma(K_n(c)) \to K_n(c)$. $\square$

Note that (7) is an optimization problem having linear constraints but quadratic objective function, that can be re-written as follows:

$$
\begin{aligned}
\min_{\substack{\mathbf{X}\in\{0,1\}^{2\times|E|},\\ \mathbf{C}\in\mathbb{R}^{|E|}}} \quad & \sum_{\{i,j\}\in E} c_e(x_{ij}+x_{ji}) && && \text{(11a)}\\
\text{s.t.} \quad & x_{ij}+x_{ji}\le 1, && e=\{i,j\}\in E, && \text{(11b)}\\
& x\left(\delta^-(W)\right)\ge 1, && W\subset V\setminus\{1\},\ W\cap\{1,\dots,t\}\neq\emptyset, && \text{(11c)}\\
& 0\le x_{ij}\le 1 && \forall i,j\in V,\ i\neq j && \text{(11d)}\\
& c_{ij}\ge 0 && \forall\{i,j\}\in E, && \text{(11e)}\\
& c_{ij}\le c_{ik}+c_{jk} && \forall\{i,j\},\{i,k\},\{j,k\}\in E. && \text{(11f)}
\end{aligned}
$$

Constraints (11b) – (11d) ensures the feasibility of x, while Constraints (11e) – (11f) ensure the property of c being a pm. Our preliminary experiments show that this is intractable even for small values of n,t (e.g., $n\le 5$). Hence, we proceed as done in [BB08, BEM07, EM08] leveraging the vertex representation of $P_{\text{DCUT}}(n,T,)$. As we have fixed the terminal set being $T=\{1,\dots,t\}$, we will denote, from now onward with $P_{\text{DCUT}}(n,t) := P_{\text{DCUT}}(n,\{1,\dots,t\})$. Hence, this can be represented as the convex combination of its finite set of vertices $\{x^1,\dots,x^{k_{n,t}}\}$. Note that the number of vertices $k_{n,t}$ depends on both n and t. Recalling what has been done in [BB08, BEM07, EM08], and by observing that, from standard results of linear programming, for each cost c, there exists an optimal solution attained at a vertex, we can re-write (11a) – (11f) as a *linear program* for each vertex $\overline{x}$

$$
\begin{aligned}
\text{Gap}(\overline{x}) := \min_{\mathbf{C}\in\mathbb{R}^{|E|}} \quad & \sum_{\{i,j\}\in E} c_e(\overline{x}_{ij}+\overline{x}_{ji}) && && \text{(12a)}\\
\text{s.t.} \quad & c_{ij}\ge 0 && \forall\{i,j\}\in E, && \text{(12b)}\\
& c_{ij}\le c_{ik}+c_{jk} && \forall\{i,j\},\{i,k\},\{j,k\}\in E, && \text{(12c)}\\
& \text{the optimal solution of } c \text{ is attained at } \overline{x}. && && \text{(12d)}
\end{aligned}
$$

As done in [BB08, EM08], we observe that constraint (12d) can be encoded thanks to the complementary slackness conditions. Such conditions ensure that $\overline{x}$ belongs to the point minimizing the STP at cost c. Hence, we can rewrite the problem Gap as follows

$$
\begin{aligned}
\text{Gap}(\overline{x}) := \min_{\mathbf{X}\in\{0,1\}^{2\times|E|}} \quad & \sum_{\{i,j\}\in E} c_e(\overline{x}_{ij}+\overline{x}_{ji}) && && \text{(13a)}\\
\text{s.t.} \quad & c_{ij}\le c_{ik}+c_{jk} && \forall\{i,j\},\{i,k\},\{j,k\}\in E, && \text{(13b)}\\
& y_e+\sum_{(i,j)\in\delta^-(W)} z_W + d_{ij}\le c_e && \forall e=\{i,j\},(i,j)\in A, && \text{(13c)}\\
& y_e = 0 && \forall (i,j) \text{ s.t. } \overline{x}_{ij}+\overline{x}_{ji}<1,\ e=\{i,j\}\in E, && \text{(13d)}\\
& z_W = 0 && \forall W\subset V\setminus\{r\},\ W\cap T\neq\emptyset \text{ s.t. } \sum_{(i,j)\in\delta^-(W)} \overline{x}_{ij}>1, && \text{(13e)}
\end{aligned}
$$

n	t	time for vertices generation	# feas problems	gap
4	3	0.04	70/256	1.000
5	3	4563.57	3655/28345	1.000
5	4	2798.17	3645/24297	1.000

Table 1: Results obtained via Polymake for the DCUT formulation. Number of nodes, number of terminals, time for generating the vertices, number of feasible problems and maximum gap found.

$$d_{ij} = 0 \qquad \forall (i,j) \in A \text{ s.t } \overline{x}_{ij} = 1, \tag{13f}$$

$$y_e + \sum_{(i,j)\in\delta^-(W)} z_W + d_{ij} - c_e = 0 \qquad \forall e = \{i,j\}, (i,j) \in A \text{ s.t } \overline{x}_{ij} > 0, \tag{13g}$$

$$y_e, d_{ij}, d_{ji} \leq 0, \qquad \forall e = \{i,j\} \in E, \tag{13h}$$

$$z_W \geq 0 \qquad \forall W \subset V \setminus \{r\},\, W \cap T \neq \emptyset, \tag{13i}$$

$$c_{ij} \geq 0 \qquad \forall \{i,j\} \in E. \tag{13j}$$

where we will denote with A the sets of all oriented edges.

2.1 Vertices enumeration with Polymake and limits of the DCUT formulation

As already discussed in the previous section, we aim to solve one Gap problem for each vertex. Hence we need an exhaustive list of vertices of the polytope $P_{\text{DCUT}}(n,t)$ for each $n \geq 3$, for each $3 \leq t \leq n-1$. To do so, we use the software Polymake [GJ00], which is designed for managing polytope and polyhedron. We implement the Gap function in Python, using the commercial solver Gurobi 11.0.0 [Gur23] for the optimization part.

On each of the so obtained vertex, we compute the Gap problem to get the maximum possible value attained at each vertex. Table 1 reports this information. From these results, we can draw several conclusions. First of all, the software Polymake can only exhaustively generate vertices for $n \leq 5$. For all the cases we could analyze, the value of the gap is exactly 1. For larger values of n, the enumeration becomes computationally infeasible. 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 actually not compatible with the triangle inequality of the cost vector c nor with its non-negativity. We tackle both issues in this paper by (a) the design of a novel formulation tailored for the metric case and (b) the design of an heuristic procedure for the vertices enumeration.

3 A novel formulation for the complete metric case

A stronger version of the DCUT formulation is proposed in the paper introducing the state-of-art solver for the (Graphic) STP, Scip-Jack [GKM+17]. More specifically, the formulation is as follows:

$$
\begin{aligned}
\min_{\mathbf{x}\in\{0,1\}^{2\times|E|}} \quad & \sum_{\{i,j\}\in E} c_e(x_{ij}+x_{ji}) && && \text{(14a)}\\
\text{s.t.} \quad & x\left(\delta^-(W)\right)\geq 1 && W\subset V\setminus\{r\},\ W\cap T\neq\emptyset, && \text{(14b)}\\
& x\left(\delta^-(r)\right)=0 && && \text{(14c)}\\
& x\left(\delta^-(v)\right)=1 && v\in T\setminus\{r\}, && \text{(14d)}\\
& x\left(\delta^-(v)\right)\leq 1 && v\in S, && \text{(14e)}\\
& x\left(\delta^-(v)\right)\leq x\left(\delta^+(v)\right) && \forall v\in S, && \text{(14f)}\\
& x\left(\delta^-(v)\right)\geq x_a && \forall a\in\delta^+(v), v\in S, && \text{(14g)}\\
& 0\leq x_a\leq 1 && \forall a\in A, && \text{(14h)}\\
& x_a\in\{0,1\}, && \forall a\in A, && \text{(14i)}
\end{aligned}
$$

where $\delta^+(W) := \{(i,j) \mid i\in W,\ j\notin W\}$. From now on, we will abbreviate this formulation with SJ. Constraints (14c)-(14e) 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 (14f) ensures that non-terminal nodes cannot be leaves of the solution. Constraint (14g) ensures that no flow generates from non-terminal nodes. However, this formulation is not specific for 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

$$
x_{ij}=\begin{cases}1, & \text{if } (i,j)\in\{(0,1),(2,3),(3,4),(4,2)\},\\ 0, & \text{else.}\end{cases} \tag{15}
$$

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 have come up with a stronger formulation, tailored for the complete metric case. This formulation is presented below.

$$
\min_{\mathbf{x}\in\{0,1\}^{2\times|E|}} \quad \sum_{\{i,j\}\in E} c_e(x_{ij}+x_{ji}) \tag{16a}
$$

$$
\begin{aligned}
\text{s.t.}\quad & x\left(\delta^-(W)\right) \geq 1 && W \subset V \setminus \{r\},\ W \cap T \neq \emptyset, && \text{(16b)}\\
& x\left(\delta^-(r)\right) = 0 && && \text{(16c)}\\
& x\left(\delta^-(v)\right) \leq 1 && v \in V\backslash r, && \text{(16d)}\\
& 2x\left(\delta^-(v)\right) \leq x\left(\delta^+(v)\right) && \forall v \in S, && \text{(16e)}\\
& 0 \leq x_a \leq 1 && \forall a \in A, && \text{(16f)}\\
& x_a \in \{0,1\}, && \forall a \in A, && \text{(16g)}
\end{aligned}
$$

In particular, in our new formulation, the left-hand side of Constraint (14f) 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 exist an optimal solution with a smaller cost that avoids detouring in that node. The existence of such a solution is guaranteed by the fact that the graph is metric and complete. As this formulation is only applicable when the graph is *metric* and complete, we will abbreviate it with CM (complete metric). 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 $T = \{0, 1\}$. Note also that we avoid adding the equivalent of Constraint (14g) because of the following lemma.

Lemma 2. *When dealing with positive costs, Constraint* (14g) *is redundant even for the simpler DCUT formulation.*

Before proving this Lemma, let us state another formulation, equivalent to the DCUT formulation, obtained by applying Benders decomposition or the max-flow-min-cut theorem. This formulation is the so called *Multi Commodity Flow* (MCF) formulation, presented below.

$$
\min_{\substack{\mathbf{X}\in\{0,1\}^{2\times|E|}\\ \mathbf{F}\in\{0,1\}^{2\times|E|\times t}}} \quad \sum_{\{i,j\}\in E} c_e(x_{ij} + x_{ji}) \tag{17a}
$$

$$
\begin{aligned}
\text{s.t.}\quad & x_{ij} + x_{ji} \leq 1, && e = \{i,j\} \in E, && \text{(17b)}\\
& f^t\left(\delta^-(v)\right) - f^t\left(\delta^+(v)\right) = \begin{cases} -1, & \text{if } v = r\\ 1, & \text{if } v = t\\ 0, & \text{otherwise,}\end{cases} && v \in V,\ t \in T \setminus \{r\} && \text{(17c)}\\
& f_{ij}^t \leq x_{ij} && && \text{(17d)}\\
& f_{ij}^t, x_{ij} \in \{0,1\}. && && \text{(17e)}
\end{aligned}
$$

This formulation is not computationally practical because of its large number of variables, but it has interesting properties that can be used in the proofs.

Proof of Lemma 2. 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

$$
\min\{c \cdot x \,|\, x \in P_{\mathrm{MCF}}(n,t)_{|x}\} = \min\{c \cdot x \,|\, x \in P_{\mathrm{DCUT}}(n,t)\},
$$

we have that there exists a configuration of varables f_{ij}^t, $t \in T$ terminals, $i, j \in V$, such that $f_{ij}^t \leq x_{ij}$ 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 optimum for strictly positive costs, we have that $x_{ij} = \max_t f_{ij}^t$ 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

$$\begin{aligned}
x_a &= x_{kl} && \text{by definition}\\
&= f_{kl}^{t_{kl}} && \text{by maximization}\\
&\leq \sum_i f_{ki}^{t_{kl}} && \text{by nonnegativity}\\
&= \sum_i f_{ik}^{t_{kl}} && \text{by (17c)}\\
&\leq \sum_i x_{ik} && \text{by (17d)}\\
&= x(\delta^-(k)) && \text{by definition}
\end{aligned}$$

which is equivalent to Constraint (14g). □

Before diving into how we use this formulation for retrieving information regarding the integrality gap of the DCUT formulation, we list some properties of the CM formulation that we retain of interest by themselves.

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 solution of the CM formulation.

Lemma 3. *Let G be a complete metric graph with c defining the edge weights and let x be an optimal solution of SJ with c as the cost vector. If*

$$c_{ij} < c_{ik} + c_{kj} \quad \forall \{i,j\}, \{i,k\}, \{j,k\} \in E, \tag{18}$$

then x is also an optimal solution for CM with the same cost vector. Moreover, if y is an optimal solution of CM for G, then it is also an optimal solution of SJ for G.

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

$$\begin{aligned}
\sum_{i \neq j} x_{ij} &\leq \sum_{k \neq j} x_{jk},\\
2 \cdot \sum_{i \neq j} x_{ij} &> \sum_{k \neq j} x_{jk},
\end{aligned}$$

for a certain $j \in V \setminus T$. It follows that there exist $i, k \in V$ 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 (18), which is in contradiction with the optimality of x.

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

Observation 1. Note that, without hypothesis (18), 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 2. Note that Lemma 3 does not hold true 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

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

We then have that $\mathrm{SJ}(\cdot) = \mathrm{CM}(\cdot)$ and $\mathrm{RSJ}(\cdot) \leq \mathrm{RCM}(\cdot)$, 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 true for the DCUT formulation.

An interesting property of the CM formulation is connectedness. Constraints (14b) enforce the fact that in a 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 true.

Lemma 4. *The support graph of any feasible point of RCM is a connected graph.*

Proof. It suffices to prove that no connected components without terminals exist since every terminal belongs to the same connected component because of Constraint (16b). So let x be a feasible point for CM and let $H \subset V$ be a connected component of x containing no terminals. We have that, because of Constraint (16e),

$$\sum_{i,j \in H} x_{ij} = \sum_{j \in H} \sum_{i \in H} x_{ij} \geq \sum_{j \in H} 2 \sum_{i \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ji} = 2 \sum_{i,j \in H} x_{ij}. \tag{19}$$

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

Note that Lemma 4 does not hold for the SJ formulation, as it is shown in Example 1.

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

$$\sum_{i,j} x_{ij} \leq \min(n-1, 2t-3). \tag{20}$$

Proof. Given x, let G_x denote the corresponding subgraph. We know that G_x is acyclic because of Constraint (16d) and we also know that G_x is connected because of Lemma 4, so G_x is a tree. Since the number of nodes of G_x is $\leq n$, we have that

$$\sum_{i,j} x_{ij} \leq n-1. \tag{21}$$

Now we only need to prove that $\sum_{i,j} x_{ij} \leq 2t-3$. We have that

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

where the last inequality holds because of Constraint (16c), Constraint (16d) combined with Constraint (16b), and Constraint (16e), respectively. Note that only the last one gives us the 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 fact 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 (16b), and because $x_{jk} \geq 0$, respectively. $\square$

Observation 3. 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 (16b) only for

$$W = W_1 \sqcup W_2, \quad W_1 \subset T \setminus r, \ |W_1| \geq 1, \quad W_2 \subset V \setminus T, \ |W_2| \leq t-2, \tag{22}$$

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 the advantages it leads in deducing information on the lower bounds of the DCUT.

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

Definition 2 (Metric Closure of a Graph)**.** Let $G = (V,E)$ 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 $\{ij\}$ in $\bar{G}$ is equal to the value of one of the shortest paths from i to j in the graph G.

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

Lemma 6. *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*

$$DCUT(G,T) = DCUT(\bar{G},T), \quad RDCUT(G,T) = RDCUT(\bar{G},T). \tag{23}$$

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 $\mathrm{DCUT}(\bar{G},T) \leq \mathrm{DCUT}(G,T)$. 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 $\mathrm{DCUT}(\bar{G},T) \geq \mathrm{DCUT}(G,T)$ and so $\mathrm{DCUT}(\bar{G},T) = \mathrm{DCUT}(G,T)$.

For the same reasoning, we have that $\mathrm{RDCUT}(\bar{G},T) = \mathrm{RDCUT}(G,T)$, 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 do not produce a solution with a bigger cost. $\square$

3.2 The gap problem for the CM formulation

With this in mind, one can proceed as done in Section 2.1 and define a Gap problem even for the CM formulation. Given $\overline{x}$ vertex of $P_{\text{CM}}(n,t)$, we define Gap_{CM} 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 a little bit involved, we first write the constraints of the dual formulation, which comes from the simple application of the duality from problem (16b) – (16g). Table 2 shows the relation between the dual variables and the constraints

Let

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

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

$$y_{ri} + V_i + \sum_{w \in W(r,i)} z_w \leq c_{ri} \qquad i \in T \setminus \{r\} \quad (24a)$$

$$y_{rj} + V_j + 2U_j + \sum_{w \in W(r,j)} z_w \leq c_{rj} \qquad j \in S \quad (24b)$$

$$y_{ij} + V_j + \sum_{w \in W(i,j)} z_w \leq c_{ij} \qquad i,j \in T \setminus \{r\} \quad (24c)$$

$$y_{ij} + V_j + 2U_j + \sum_{w \in W(i,j)} z_w \leq c_{ij} \qquad i \in T \setminus \{r\}, j \in S \quad (24d)$$

$$R + y_{ir} \leq c_{ir} \qquad \forall i \in T \setminus \{r\} \quad (24e)$$

$$R + y_{jr} \leq c_{jr} \qquad \forall j \in S \quad (24f)$$

$$y_{ji} + V_i - U_j + \sum_{w \in W(j,i)} z_w \leq c_{ji} \qquad i \in T \setminus \{r\}, j \in S \quad (24g)$$

$$y_{ij} + V_j + 2U_j - U_i + \sum_{w \in W(i,j)} z_w \leq c_{ij} \qquad i,j \in S, \quad (24h)$$

$$R \text{ free},\ z \geq 0,\ V,U,y \leq 0 \quad (24i)$$

Note that we can merge some constraints, in particular, (24a) and (24c) are the same constraint where $i \in T$ and $j \in T \setminus \{r\}$. The same holds for (24b) and (24d) if $i \in T$ and $j \in S$. Lastly, we can drop constraint (24e) and (24f) as variable R 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

$$y_{ij} + V_j + \sum_{w \in W(ij)} z_w \leq c_{ij} \qquad i \in T, j \in T \setminus \{r\} \quad (25a)$$

$$y_{ij} + V_j + 2U_j + \sum_{w \in W(i,j)} z_w \leq c_{ij} \qquad i \in T,\ j \in S \quad (25b)$$

Primal constraints	Dual variables	Primal constraints	Dual variables
(16b)	z_W	(16c)	R
(16d)	V_j	(16e)	U_j
(16f)	y_{ij}		

Table 2: Association between dual variables and primal constraints

$$y_{ji} + V_i - U_j + \sum_{w \in W(j,i)} z_w \le c_{ji} \qquad i \in T \setminus \{r\}, j \in S \quad (25c)$$

$$y_{ij} + V_j + 2U_j - U_i + \sum_{w \in W(i,j)} z_w \le c_{ij} \qquad i, j \in S, \quad (25d)$$

$$z \ge 0,\ V, U, y \le 0 \quad (25e)$$

Given a vertex $x \in P_{\mathrm{CM}}(n,t)$, we want to write the gap problem, as done in (13a) – (13j). Hence, we introduce variable c_{ij} $\{i,j\} \in E$ and write the slackness compatibility condition as follows

$$y_{ij} + V_j + \sum_{w \in W(ij)} z_w - c_{ij} = 0 \qquad \forall i \in T, j \in T \setminus \{r\},\ x_{ij} > 0 \quad (26a)$$

$$y_{ij} + V_j + 2U_j + \sum_{w \in W(i,j)} z_w - c_{ij} = 0 \qquad \forall i \in T,\ j \in S,\ x_{ij} > 0 \quad (26b)$$

$$y_{ji} + V_i - U_j + \sum_{w \in W(j,i)} z_w - c_{ji} = 0 \qquad \forall i \in T \setminus \{r\}, j \in S\ x_{ij} > 0 \quad (26c)$$

$$y_{ij} + V_j + 2U_j - U_i + \sum_{w \in W(i,j)} z_w - c_{ij} = 0 \qquad \forall i, j \in S,\ x_{ij} > 0 \quad (26d)$$

$$z_W = 0 \qquad \forall W \subset V \setminus \{r\},\ W \cap T \neq \emptyset\ x\left(\delta^-(W)\right) > 1 \quad (26e)$$

$$V_j = 0 \qquad \forall j \in V \setminus \{r\},\ x\left(\delta^-(j)\right) < 1 \quad (26f)$$

$$U_j = 0 \qquad \forall j \in S,\ 2x\left(\delta^-(v)\right) < x\left(\delta^+(v)\right) \quad (26g)$$

$$y_{ij} = 0 \qquad \forall (i,j) \in A,\ x_{ij} < 1. \quad (26h)$$

As done for the DCUT formulation, we implement this optimization model in Python, using Gurobi 11.0.0 as an optimization solver and we list all the vertices using Polymake. We compute the Gap problem on each so-obtained vertex to get the maximum possible value attained at each vertex. Table 3 reports this information.

First of all, we can observe that the number of vertices generated is smaller, and all of them are actually feasible. As expected, the integrality gap si 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 is not able to generate vertices for $n \ge 6$. For this reason, we describe two heuristic procedures to generate a large number of vertices.

n	t	time for vertices generation	# feas problems	gap
4	3	0.732	4/4	1/1
5	3	44.62	5/5	1/1
5	4	37.01	44/44	1/1

Table 3: Results obtained via Polymake for the CM formulation. Number of nodes, number of terminals, time for generating the vertices, number of feasible problems and maximum gap found.

4 Vertices enumeration

In this section, we present some theoretical results and algorithmic procedures that we use to enumerate vertices of the polytope $P_{\mathrm{CM}}(n,t)$. We first introduce some 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

Let us call *spanning vertex* a vertex x of $P_{\mathrm{CM}}(n,t)$ such that it visits all of the nodes, that is $x(\delta^-(i)) + x(\delta^+(i)) > 0$ for all $i \in V$. Note also that Lemma 4 implies that every spanning vertex is also connected. In a STP, some of the potential Steiner nodes may or may not be part of an optimal solution. This clearly holds true for vertices of $P_{\mathrm{CM}}(n,t)$, both integer and non-integer, i.e., not all of the vertices are spanning vertices. Because of this fact, we can ask ourselves if a non-spanning vertex of $P_{\mathrm{CM}}(n,t)$ can be seen as a spanning vertex of a polytope of a smaller dimension, and vice versa, that is, if a spanning vertex of $P_{\mathrm{CM}}(n,t)$ can be seen as a vertex of a polytope of a bigger dimension. The following results link vertices of $P_{\mathrm{CM}}(n+1,t)$ with vertices of $P_{\mathrm{CM}}(n,t)$ and vice versa. These results will be used in the enumeration of vertices to reduce the dimension of our research space by avoiding redundancy.

Lemma 7. *Let $x \in \mathbb{R}^{(n-1)\times n}$. Define $y \in \mathbb{R}^{n\times(n+1)}$ as*

$$y_{ij} = \begin{cases} x_{ij}, & 1 \le i, j < n+1, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

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

Proof. Let $x \in P_{\mathrm{CM}}(n,t)$. Note that y satisfies the domain constraints. Regarding Constraint (16b), we have to distinguish between two cases. Let W be a set as described in (16b) 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 (16c) - (16d) are clearly satisfied by y since x satisfies them and we are only adding variables that take value zero. Regarding Constraint (16e), 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 $y \in P_{\mathrm{CM}}(n+1,t)$ of the form (27). Note that x satisfies the domain constraints. Let W be a set as descried in (16b) for x. Let $\hat{W} \coloneqq W \cup \{n+1\}$. $\hat{W}$ is a set for which y satisfies the correspondent constraint (16b). In the $\hat{W}$ constraint, the only variables that appear are the one 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 (16c) - (16d) are clearly satisfied by x since y satisfies them. Regarding Constraint (16e), 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. □

We stress that, according to the formulation of the previous lemma, the Lemma holds if we label the added node as a potential Steiner node.

Lemma 8. *Let x be a vertex of $P_{CM}(n,t)$. Then*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i,j \neq n+1, \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

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 as well. To do so, we will rely on projection. In detail, We have that $y \in P_{\mathrm{CM}}(n+1,t)$ because of Lemma 7. Let $P_{\mathrm{CM}}(n+1,t)_0$ be the subpolytope of $P_{\mathrm{CM}}(n+1,t)$ defined as

$$P_{\mathrm{CM}}(n+1,t)_0 \coloneqq \{z \in P_{\mathrm{CM}}(n+1,t) : \ z_{i,n+1} = z_{n+1,j} = 0, \ 1 \leq i,j \leq n\}. \tag{29}$$

Let

$$\begin{aligned} \pi \colon P_{\mathrm{CM}}(n+1,t)_0 &\twoheadrightarrow P_{\mathrm{CM}}(n,t) \\ (z_{ij})_{i,j} &\mapsto (z_{ij})_{i,j \neq n+1} \end{aligned} \tag{30}$$

be the projection considering the first n nodes. Note that $\pi(y) = x$ and that π is an injective map. Note also that $\mathrm{Im}(\pi) \subset P_{\mathrm{CM}}(n,t)$ because of Lemma 7. By contradiction, suppose that there exist $a, b \in P_{\mathrm{CM}}(n+1,t)$ such that $a \neq b$, $y = \frac{1}{2}a + \frac{1}{2}b$. We have that

$$y_{i,n+1} = y_{n+1,j} = 0 = \frac{1}{2}(a_{i,n+1} + b_{i,n+1}) = \frac{1}{2}(a_{n+1,j} + b_{n+1,j}). \tag{31}$$

Since $a, b \in P_{\mathrm{CM}}(n+1,t)$, 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, $a, b \in P_{\mathrm{CM}}(n+1,t)_0$ and we can define

$$\begin{aligned} c &= \pi(a), \\ d &= \pi(b), \end{aligned} \tag{32}$$

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. $\square$

Lemma 9. *Let y be a vertex of $P_{CM}(n,t)$ of the form*

$$y_{ij} = \begin{cases} x_{ij}, & \text{if } i \neq k \neq j, \\ 0, & \text{else,} \end{cases} \tag{33}$$

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

Proof. We have that $x \in P_{\text{CM}}(n \setminus \{k\}, t)$ because of Lemma 7. Let

$$\begin{aligned} i\colon P_{\text{CM}}(n \setminus \{k\}, t) &\hookrightarrow P_{\text{CM}}(n,t) \\ (z_{i,j})_{i \neq k \neq j} &\mapsto ((z_{i,j})_{i \neq k \neq j}, 0, \dots, 0) \end{aligned} \tag{34}$$

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 7. By contradiction, suppose there exist $c, d \in P_{\text{CM}}(n \setminus \{k\}, t)$ such that $c \neq d$, $x = \frac{1}{2}c + \frac{1}{2}d$. If we define

$$\begin{aligned} a &= i(c), \\ b &= i(d), \end{aligned} \tag{35}$$

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. $\square$

Note that, even in this case, we are working on potential Steiner nodes.

Observation 4. 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

$$P_{\text{CM}}(n+1,t)_0 \cong P_{\text{CM}}(n,t). \tag{36}$$

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 true for the DCUT formulation as well as for the SJ formulations. The proof can be done in almost the same way.

Observation 5. 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 $P_{\mathrm{CM}}(n+1, t+1)$ to a vertex of $P_{\mathrm{CM}}(n, t)$ and vice versa is the "dual" procedure of the previous one, i.e., adding or removing one 1. Note that this can be done in different ways. More precisely, the following procedures start with avertex of $P_{\mathrm{CM}}(n, t)$ and return a vertex of $P_{\mathrm{CM}}(n+1, t+1)$:

(a) add an edge of weight 1 between a node v of indegree 1 and the new added terminal, see for example Figure 1a → Figure 2a and Figure 1b → Figure 2b;

(b) same as (a), but substituting the outflow of v with outflow of the new added terminal, see for example Figure 1a → Figure 2c;

(c) add an edge of weight 1 between the new added terminal and the root, then swap the role of this two nodes, see for example Figure 1b → Figure 2d.

Reversing these procedures, when possible, allow us to go from a $P_{\mathrm{CM}}(n+1, t+1)$ to a vertex of $P_{\mathrm{CM}}(n, t)$. 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.

4.2 Two heuristics procedure for vertices enumeration

In the following, we state some more properties of the CM formulation that permit the formulation of two different heuristic procedures, in particular, one general search and one dedicated to a particular class of vertices. Note that we are only interested in spanning vertices from now on (Observation 4).

4.2.1 The 1-2-costs heuristic

The first procedure is based on the following theorem, which states 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 node 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

$$\sum x_e = t + s - 1, \qquad \sum x'_e = t + s' - 1. \tag{37}$$

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

$$\sum c_e x_e = t + s - 1. \tag{38}$$

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

$$\begin{aligned} \sum c_e x'_e &= \sum_{I_0} c_e x'_e + \sum_{I_1} c_e x'_e + \sum_{I} c_e x'_e = \\ &= 2 \times |I_0| + |I| \geq |I_0| + |I| \\ &= t + s' - 1 \geq t + s - 1 = \sum c_e x_e, \end{aligned} \tag{39}$$

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 $\mathcal{S} = \{s_1, \ldots, s_k\} = \mathcal{S}(x) \setminus \mathcal{S}(x')$ 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 the fact 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

$$(\sum_i^k deg(s_i)) - z. \tag{40}$$

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

$$\begin{aligned} E_1 &\leq s + t - 1 - ((\sum_i^k deg(s_i)) - z), \\ E_2 &= s + t - k - 1 - E_1, \end{aligned} \tag{41}$$

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

$$\begin{aligned} 2 \times &((\sum_i^k deg(s_i)) - k - z) - ((\sum_i^k deg(s_i)) - z) = \\ &= (\sum_i^k deg(s_i)) - 2k - z \geq 3k - 2k - z \geq k - z \geq 1, \end{aligned} \tag{42}$$

and so x' is not optimal and we have a contradiction. Note that $deg(s_i) \geq 3$ because of Constraints (16d) and (16e), 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 6. Note that the generalization of Theorem 1 does not hold in general for the non-integer case, i.e., if x is a non-integer point of $P_{\mathrm{CM}}(n,t)$, then x is not necessarily an optimal solution for the CM formulation with the metric cost

$$c_{ij} = 2 - \mathbb{1}(x_{ij} + x_{ji} > 0), \tag{43}$$

see for example the vertex depicted in Figure 4a. In this case, with the cost assignation (43), 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 4a 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 (43) 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 an 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 connected graphs only, and so to graphs with costs $\{1, 2\}$ in which the subgraph regarding the edges of cost 1 spans all the node and is connected. We are aware that this is a strong restriction, making the procedure unable to find some vertices, see Observation 6. 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 5.1 we broadly discusses this choice.

4.2.2 Pure half-integer vertices

We now focus on a particular set of vertices, namely, vertices x such that $x_{ij} \in \{0, 1/M\}$. For simplicity, we restrict our attention to the case $M = 2$, and we call such vertices *pure half-integer* vertices. In particular, given a non-integer vertex x of $P_{\mathrm{CM}}(n,t)$, 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, we state and prove some properties of PHI vertices.

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

Algorithm 1 1-2-costs vertices heuristic

procedure OTC(n, t)
    $\mathbb{G} = \{G = (V, E) \mid G \text{ connected}, \ |V| = n, \ n \leq |E| \leq n \cdot t - t^2\}$
    $\mathbb{T} = \{T \mid T \subset \{1, \dots, n\}, \ |T| = t\}$
    $\mathfrak{G} = \emptyset$
    for $G \in \mathbb{G}$ **do**
        for $T \in \mathbb{T}$ **do**
            for $r \in T$ **do**
                $G_{T,r}$ = node-colored graph with G as its support graph, r colored as `root`, i colored as `terminal` $\forall i \in T \setminus \{r\}$, j colored as `steiner` $\forall i \notin T$
                if $H \ncong G_{T,r} \ \forall H \in \mathfrak{G}$ **then**
                    add $G_{T,r}$ to $\mathfrak{G}$
                end if
            end for
        end for
    end for
    $\mathcal{V} = \emptyset$
    for $G_{T,r} \in \mathfrak{G}$ **do**
        obtain the STP instance (G, T, r) from $G_{T,r}$ with $c_{ij} = 1$ if $\{i, j\} \in G_{T,r}$ and $c_{ij} = 2$ otherwise
        solve (16a) - (16e)
        if a solution x is found **then**
            if x is a non-integer vertex of $P_{\text{CM}}(n, t)$ **then**
                add x to $\mathcal{V}$
            end if
        end if
    end for
end procedure

Proof. Since x is pure half-integer, we have that $x_{ij} = 1/2$. Suppose by contradiction that $x_{ji} \neq 0$, and so by the same reasoning $x_{ji} = 1/2$. Because of Lemma 4, 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 Constraint (16c). 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 $x_{ik} = 0$, else, we can do the same reasoning for the nodes $\{i, j, k\}$ and repeat it until we get back to the root, which has no inflow. Now we have to distinguish between two cases.

(a) No other inflow is present in j, i.e., $x_{aj} = 0 \ \forall a \neq j$. 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 $x'_{ji} = 0$. Clearly, for any non negative c, $c^T x' < c^T x$. 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 differs from x. Let us take one of this set, and define $\bar{W} = W \cup \{j\}$. We can write

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

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 (16e) because x' satisifed all of the other constraint since x is feasible for CM. Regarding Constraint (16e), 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

$$c^T x'' \leq c^T x'$$

for the non-negativity and the triangle inequality. Hence

$$c^T x'' < c^T x,$$

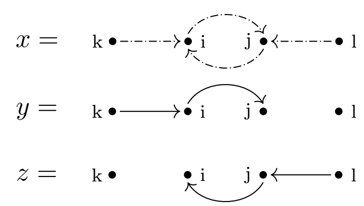
from the relation between x and x' already proved. Hence, we can conclude that if the only inflow of the (Steiner) node j is x_{ij}, x is neither optimal for the CM nor for the DCUT.

(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 also that both k and l have an inflow of 1. This will ensure feasibilty of the two points we are bout to construct. Then x is not a vertex of $P_{\text{DCUT}}(n,t)$, because by setting

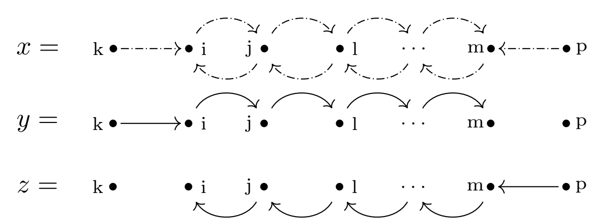
$$y_{ab} = \begin{cases} 0, & \text{if } a = l,\ b = j, \text{ or } a = j,\ b = i, \\ 1, & \text{if } a = i,\ b = j, \text{ or } a = k,\ b = i, \\ x_{ab}, & \text{else}, \end{cases} \tag{44}$$

$$z_{ab} = \begin{cases} 1, & \text{if } a = l,\ b = j,\ \text{or } a = j,\ b = i, \\ 0, & \text{if } a = i,\ b = j,\ \text{or } a = k,\ b = i, \\ x_{ab}, & \text{else,} \end{cases} \tag{45}$$

we have $y \neq z$, $x = \frac{1}{2}y + \frac{1}{2}z$, and $y, z \in P_{\mathrm{DCUT}}(n,t)$ 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 $x_{jl} \neq 0$, i.e., $x_{jl} = 1/2$, 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 considering 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 that has this property. If we do not find it, we go backward till 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 by the following reasoning. First of all, because of Lemma 10, there are no loops of length 2, ando so every edge can be oreinted 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 (16d) 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 (17c), 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 (17c).

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

Lemma 11. *Let $(x_{ij})_{ij}$ 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*

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

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 then $n - t + 2(t-1) = n + t - 2$.

For the second point, because of Constraint (16b) we have that at least two edges exit from the root and at least two edges enter in every other terminals. 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 has a degree of 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 (16d) 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 we check if the remaining ones are vertices of $P_{\text{CM}}(n,t)$. This procedure, called PHI(n,t), is illustrated in Algorithm 2.

Algorithm 2 Pure half-integer vertices search

procedure PHI(n, t)
  $\mathbb{G} = \{G = (V, E) \mid G \text{ connected}, \ deg(i) \geq 2 \, \forall i \in V, \ |V| = n, \ |E| = n + t - 2\}$
  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
    end if
  end for
  $\mathcal{V} = \emptyset$
  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 $P_{\mathrm{CM}}(n, t)$ 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 $P_{\mathrm{CM}}(n, t)$ **then**
            add x to $\mathcal{V}$
          end if
        end if
      end if
    end if
  end for
end procedure

Observation 7. 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 give 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 discuss the results we obtained with our heuristic procedure in terms of the number of vertices.

Implementation details. All the tests have been executed in parallel on 8 CPUs 13th Gen Intel(R) Core(TM) i5-13600, and 16 GB of RAM. All the functions have been implemented in Python. For the optimization problems, we use the commercial solver Gurobi [Gur23].

5.1 A comparison between the two proposed heuristics

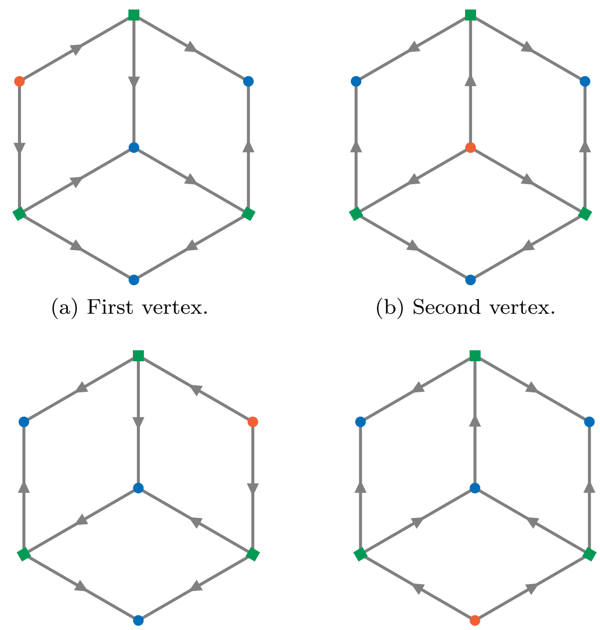
We now discuss the differences and similarities of the two heuristics. First of all, notice how neither of the two are exhaustive procedures: there exists a vertex that can be found by the one-two-cost (OTC) heuristic but not by the PHI heuristic, for example Figure 2a, and there exists also a vertex that can be found by the PHI heuristic but not by the OTC heuristic, see Observation 6. 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 can be an exhaustive search by dropping the connectivity constraint.

On the other hand, the OTC heuristic is very costly, even with the additional constraint of generating connected graphs only: for every generated graph, every possible assignation of the root, terminal nodes, and potential Steiner nodes needs to be computed, and for every assignation one must solve an LP. The solution of the LP does not guarantee to find fractional vertices with an integrality gap of 1, since it may return as an equivalent solution an integer vertex. In addition, this procedure does not guarantee to generate non-isomorphic solutions and so we have to filter the set of vertices by node-colored edge-weighted graph isomorphism. The procedure does not even guarantee to find spanning vertices only. The PHI heuristic is able never to generate isomorphic graphs, and so every vertex generated belongs to a unique class of isomorphism. In addition, no LP needs to be solved, since given an orientation of the vertices, the role of every node is uniquely determined. Lastly, note that in the heuristic OTC, we have applied the extra hypothesis that the number of edges is bounded above by $n \cdot t - t^2$ for efficiency purposes. Without this hypothesis, OTC is infeasible for $n \geq 8$. Table 4 results are obtained with this distinction. Note also that, even with this restriction, OTC is impractical for $n \geq 9$.

5.2 Lower bounds for the integrality gap for small values of n

Table 4 and Table 5 present the lower bounds on the integrality gap we found with the two heuristics, as well as the number of noninteger vertices. Note that the vertices found by the OTC procedures presented in the table are filtered by isomorphism. In this subsection, we discuss and comment the results we obtain.

$n = 6, 7$. For $n = 6$, for every value of $t \leq 3 \leq n - 1$, the best lower bound we found is equal to 1. We conjecture that for $n = 6$, both the CM and the DCUT formulation have a gap of 1. For $n = 7$, we found 4 vertices attaining the gap of 10/9 with the heuristic OTC, with three of them belonging to the same class of

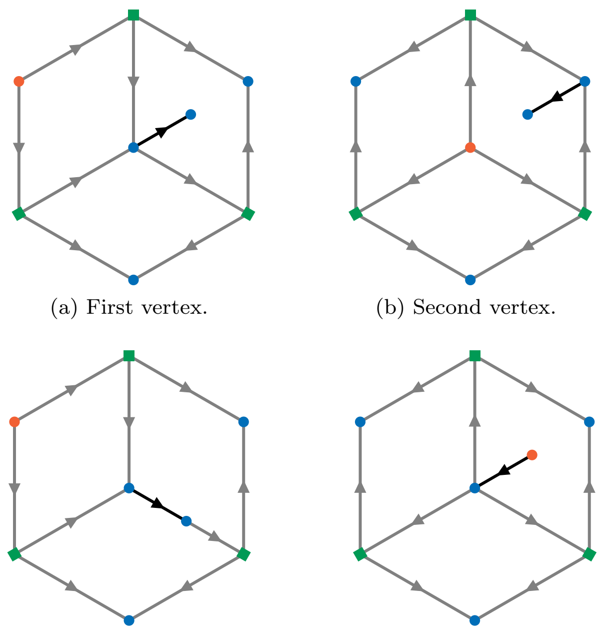


(a) First vertex. (b) Second vertex. (c) Third vertex. (d) Fourth vertex.

Figure 1: Fractional vertices of $(7,4)$ - all with integrality gap 10/9. Red circle: root. Blue circles: Terminals. Green square: Steiner node

isomorphism. These vertices are *pure half-integer*, e.g., $x_a \in \{0, \frac{1}{2}\}$. Figure 1 we present these vertices. We observe that, although the directed support graph is the same, the oriented one changes and, in particular, it changes the node that we label as “root”. Note that the heuristic PHI is only able to find two of them. This is given by the fact that PHI is able to generate only one vertex for every class of isomorphism of node-colored edge-weighted directed graphs, while the OTC may find more than one representative for the same class of isomorphism. Figure 1a, Figure 1c, and Figure 1d represent in fact three isomorphic graphs.

$n = 8$. The case $n = 8$ is more involving. While neither the PHI nor the OTC are able to find fractional vertices for the cases $t = 3$, the PHI heuristic does not find fractional vertices even for the case $t = 4$, the OTC finds again the vertices of $(7,4)$ since it does not only find spanning vertices, as we already discussed. Both heuristics only finds fractional vertices of integrality gap 1 for the case $t = 6$, while for the case $t = 7$ only the PHI heursitic is able to find fractional vertices, again of integrality gap 1. The most interesting case is $t = 5$. The maximum integrality gap is depicted in Figure 3a while different values of integrality gap are depicted in Figure 4a and Figure 4b. Note that the maximum integrality gap of this case for the PHI heuristic is 12/11, while the maximum intgerality gap for the OTC heuristic is again 10/9: some of



(a) First vertex. (b) Second vertex.

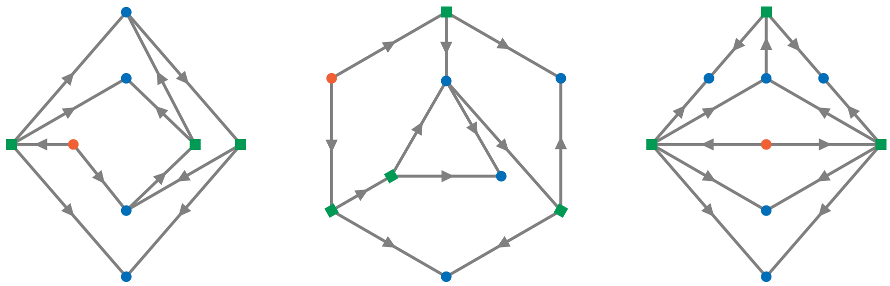
(c) Third vertex. (d) Fourth vertex.

Figure 2: Fractional vertices of $(8, 5)$ - all with integrality gap 10/9. Red circle: root. Blue circles: Terminals. Green square: Steiner node

the vertices attaining this value are depicted in Figure 2. Note also how these vertices can be obtained from vertices of $(7, 4)$, see Observation 5.

$n = 9$. For $n = 9$, we can only run the PHI heuristics, as the OTC heuristic goes 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 integrality gap for the cases $t = 5, 6$ are 10/9 and 14/13, depicted in Figure 3b and Figure 3c, rispectively. 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 $\frac{2m}{2m-1}$.

$n \geq 10$ For $n \geq 10$, even the PHI heuristic is too intense. For $t \in \{8, 9\}$, we weren't able to conclude the experiments within 80 hours. For $t \in \{3, 4, 5, 6, 7\}$, we can make similar claims with respect to the previous values of n. We face a new value of integrality gap in the case $t = 6$, that is, a value of 19/18. Note that in this case, the value is of the form $\frac{2m+1}{2m}$, in contrast to what we found for the cases $n \leq 9$. For $n = 11, 12$, we show that the cases $t \leq 5$ did not lead to any feasible PHI vertex. Test with bigger values of t were computationally infeasible.



(a) $(n, t) = (8, 5)$, gap 12/11.

(b) $(n, t) = (9, 5)$, gap 10/9.

(c) $(n, t) = (9, 6)$, gap 14/13.

Figure 3: PHI vertices of maximum gap for different values of (n, t).

		PHI			OTC		
n	t	# vert.	max gap	# vert. max. gap	# vert.	max gap	# vert. max. gap
6	3	0	-	-	0	-	-
	4	1	1/1	1	0	-	-
	5	7	1/1	7	0	-	-
7	3	0	-	-	0	–	
	4	2	10/9	2	11	10/9	2
	5	46	1/1	46	19	1/1	19
	6	71	1/1	71	8	1/1	8
8	3	0	-	-	0	-	-
	4	0	-	-	19	10/9	2
	5	89	12/11	15	195	10/9	14
	6	1070	1/1	1070	239	1/1	239
	7	758	1/1	758	0	-	-

Table 4: Comparison between the PHI heuristic and the OTC heuristic. For $n \leq 7$ we do not limit the number of edges to $n \cdot t - t^2$. We report the number of non-isomorphic vertices each heuristic can find e the maximum value of the gap. We also report how many vertices attained the maximum gap.

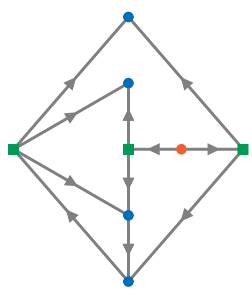
5.3 Beyond pure half-integer vertices

For the previous section, it seems that the PHI heuristic is more suitable for finding interesting vertices of the CM formulation. Interestingly, we observe that it can be extended to find all the vertices of the type $\{0, 1/M\}$. An interesting case is the one with $M = 4$, namely, vertices where the entries are only in $\{0, 1/4\}$. Let's call these vertices *pure one-quarter* (POQ). In this case, our heuristic would work as follows for each pair (n, t)

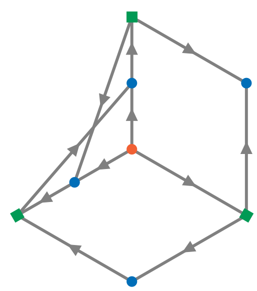
1. Generate all the non-isomorphic graphs having (i) every node of degree at least 3 and (ii) exactly $n + 3t - 4$.

n	t	# vert.	max gap	# vert. max. gap	n	t	# vert.	max gap	# vert. max. gap
	3	0	-	-		3	0	-	-
	4	0	-	-		4	0	-	-
	5	64	10/9	12		5	15	10/9	7
9	6	4389	14/13	200	10	6	7386	10/9	73
	7	21121	1/1	21121		7	155120	16/15	2653
	8	8987	1/1	8987		8	N.A	N.A	
						9	N.A	N.A	

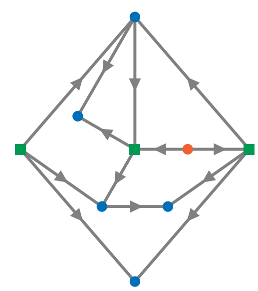
Table 5: Performances of the PHI heuristic for $n \geq 9$. N.A means that the time-limit has been hitted.



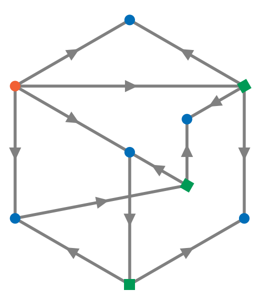
(a) $(n, t) = (8, 5)$, gap 14/13.



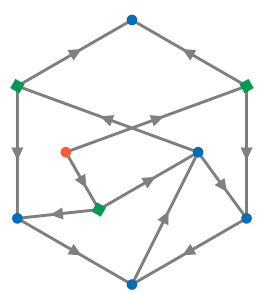
(b) $(n, t) = (8, 5)$, gap 18/17.



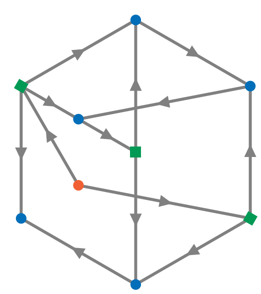
(c) $(n, t) = (9, 6)$, gap 16/15.



(d) $(n, t) = (9, 6)$, gap 20/19.



(e) $(n, t) = (9, 6)$, gap 22/21.



(f) $(n, t) = (9, 6)$, gap 24/23.

Figure 4: Fractional vertices of different gaps for different values of (n, t).

2. Filter this list by excluding all the graphs having more than $n-t-1$ nodes with degree 3 (In our graph, 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 of the so-oriented graphs, we use `watercluster2` to get all the possible orientations of edges, assuming that the maximum indegree must be equal to 4

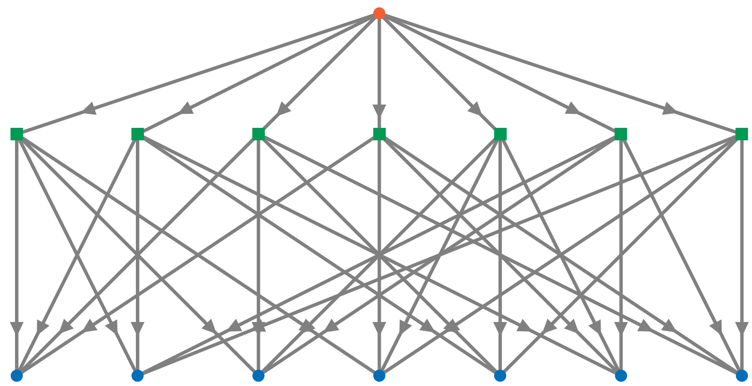


Figure 5: Skutella's graph. Note that this graph is a POQ vertex of $(n, t) = (15, 8)$ of integrality gap 8/7.

4. We filter out the list thus obtained 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 $n-1$ nodes with in-degree 1.

These vertices are particularly relevant. In [KPT11], it is shown that the integrality gap of the DCUT formulation is at least 8/7 by explicitly showing an instance leading to such a gap. The instance has 15 nodes and 8 terminals. The optimal vertex is of POQ type, and it's due to a personal communication between Skutella and the authors of [KPT11]. Figure 5 shows Skutella's graph. Note that, our heuristic would have been able to find such a graph. Note also that solving the Gap function for the CM formulation leads to a gap equal to 8/7. Hence the maximum gap we can have on the Skutella's vertex is exactly 8/7.

We run POQ algorithm for $6 \leq 8$, and for every $3 \leq t \leq n-1$ and we observe that no vertex with such properties.

5.4 Comparison between CM and SJ

In this section, we compare the two formulations from a computational perspective. More specifically, we compare the CM formulation with the SJ formulation on the complete metric instance. Note that, in the standard benchmark library, namely the SteinLib [KMV01] the number of instances that are both complete and metric is relatively small. In all of such instances, since distances have been calculated by rounding the Euclidean distance, the triangle inequality is not always satisfied, leading the CM to obtain only suboptimal solutions. Hence, before attacking the instances with the two formulations, we replace the complete graph with its *metric closure*, that is, the smallest metric space that contains the graph's vertex set and where the distance between any two vertices is defined as the length of the shortest path between them in the graph. Table 6 reports the mean runtime on 10 runs of Gurobi with different seeds, as

Instance Family	 Name	 n	 t	 CM (s)	 SJ (s)	 Gap CM (%)	 Gap SJ (%)
MC	mc3.stp	97	45	18.56	**4.94**	0.00	0.00
	mc2.stp	120	60	57.95	**21.47**	0.00	0.00
	mc13.stp	150	80	TL	TL	1.30	**1.19**
	mc8.stp	400	188	TL	TL	91.53	**73.71**
	mc7.stp	400	170	TL	TL	94.84	**87.74**
	mc11.stp	400	213	TL	TL	84.35	**67.80**
P4E	p455.stp	100	5	**39.07**	TL	**0.00**	4.52
	p456.stp	100	5	**61.96**	TL	**0.00**	1.35
	p459.stp	100	20	112.45	**100.05**	0.00	0.00
	p457.stp	100	10	**117.58**	TL	**0.00**	14.32
	p461.stp	100	50	TL	**201.87**	1.20	0.00
	p458.stp	100	10	TL	TL	**3.90**	16.92
	p460.stp	100	20	TL	TL	**6.26**	8.49
	p463.stp	200	10	TL	TL	51.51	**48.19**
	p464.stp	200	20	TL	TL	87.14	**69.22**
	p465.stp	200	40	TL	TL	85.97	**53.40**
	p466.stp	200	100	TL	TL	84.84	**39.07**
P4Z	p402.stp	100	5	2.30	**1.67**	0.00	0.00
	p403.stp	100	5	15.64	**17.93**	0.00	0.00
	p404.stp	100	10	21.15	**5.70**	0.00	0.00
	p401.stp	100	5	21.57	**14.79**	0.00	0.00
	p405.stp	100	10	23.87	**9.83**	0.00	0.00
	p406.stp	100	10	30.03	**8.71**	0.00	0.00
	p409.stp	100	50	35.97	**23.26**	0.00	0.00
	p408.stp	100	20	62.98	**35.07**	0.00	0.00
	p410.stp	100	50	135.79	**59.63**	0.00	0.00
	p407.stp	100	20	TL	**130.07**	0.83	**0.00**
X	brasil58.stp	58	25	TL	TL	**0.58**	1.58
	world666.stp	666	174	TL	TL	**93.27**	inf

Table 6: Comparison of the runtime of a naive implementation of the SJ and CM linear programming

well as the gap. We set, for each instance, a time-limit of 5 minutes and TL stands for time-limit reached. First of all, we observe that SJ performs slightly better than the CM formulation we introduced, however, some interesting comments can be made. For the family MC (instances randomly generated), and P4E/P4Z [CGR92] (randomly generated instances with euclidean weights) SJ performs better on the instances that can be solved to optimality and on the ones that hit the time-limit, achieving a better gap. On X, where the instances have been converted from TSPLIB [Rei91] problems, by selecting some nodes as Terminals, our routine seem to work better, also providing a feasible optimal solution on world666 when SJ is not able to do so.

We cannot claim at all that formulation CM outperforms formulation SJ. However, the result obtained on X may encourage further investigation, for example, in the design of an ad-hoc branching procedure.

6 Conclusion and future works

In this paper, we have studied the metric STP on graphs, focusing particularly on computing lower bounds for the integrality gap for the DCUT formulation. We introduced a novel approach tailored for the metric case to overcome the outlined limitations of the DCUT formulation in the metric case.

We establish interesting properties of this new formulation and of the polytope associated with its linear relaxation.

Central to our contribution is the extension of the Gap problem from the symmetric and asymmetric TSP to STP.

To facilitate the usage of the Gap problem, we propose two heuristic approaches for generating suitable vertices as inputs. Unfortunately, our heuristics are only able to generate vertices for $n \leq 10$, still outperforming exact methods that got stuck when $n \leq 5$.

We compare the performances of the two heuristics and the impact they have on providing insights into the exact value of the integrality gap. Although we were not able to surpass 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 most ambitious gap is actually $\frac{10}{9}$.

We retain that our study raises several interesting research questions. First, is it possible to design an ad-hoc branching procedure 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 some conjectures.

First, we conjecture that for $t = 3$, the integrality gap is 1. Second, we conjecture that our OTC procedure, without the restriction on connectedness and the bound on the number of edges, is exhaustive, hence, every vertex of $\mathrm{P_{CM}}(n, t)$ can be obtained as an optimal solution of a 1-2 cost. Third, we conjecture that every spanning vertex x of $\mathrm{P_{CM}}(n, t)$ with $x_{ij} \in \{0, 1/M\}$ has an indegree of 1 in every Steiner node, and so our PHI is exhaustive for every pure half-integer spanning vertex.

References

[AdOO21] 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.

[BB08] Genevieve Benoit and Sylvia Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.

[BEM07] Sylvia Boyd and Paul Elliott-Magwood. Structure of the extreme points of the subtour elimination polytope of the stsp. *Combinatorial Optimization and Discrete Algorithms*, 23:33–47, 2007.

[BGRS13] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM (JACM)*, 60(1):1–33, 2013.

[CGR92] S. Chopra, E.R. Gorres, and M.R. Rao. Solving the Steiner tree problem on a graph using branch and cut. *ORSA Journal on Computing*, 4:320–335, 1992.

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

[DF55] George Bernard Dantzig and Delbert Ray Fulkerson. *On the Max Flow Min Cut Theorem of Networks.* RAND Corporation, Santa Monica, CA, 1955.

[E+67] Jack Edmonds et al. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240, 1967.

[EM08] Paul Elliott-Magwood. *The integrality gap of the asymmetric travelling salesman problem.* PhD thesis, University of Ottawa (Canada), 2008.

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

[GKM+17] Gerald Gamrath, Thorsten Koch, Stephen J Maher, Daniel Rehfeldt, and Yuji Shinano. Scip-jack—a solver for stp and variants with parallelization extensions. *Mathematical Programming Computation*, 9:231–296, 2017.

[GM93] Michel X Goemans and Young-Soo Myung. A catalog of steiner tree formulations. *Networks*, 23(1):19–28, 1993.

[Gur23] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[GW95] Michel X Goemans and David P Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[HO94] JX Hao and James B Orlin. A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17(3):424–446, 1994.

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

[KMB81] Lawrence Kou, George Markowsky, and Leonard Berman. A fast algorithm for steiner trees. *Acta informatica*, 15:141–145, 1981.

[KMV01] Thorsten Koch, Alexander Martin, and Stefan Voß. Steinlib: An updated library on steiner tree problems in graphs. In Xiu Zhen Cheng and Ding-Zhu Du, editors, *Steiner Trees in Industry*, pages 285–325. Springer US, Boston, MA, 2001.

[KPT11] Jochen Könemann, David Pritchard, and Kunlun Tan. A partition-based relaxation for Steiner trees. *Mathematical Programming*, 127(2):345–370, April 2011.

[Lju21] Ivana Ljubić. Solving steiner trees: Recent advances, challenges, and perspectives. *Networks*, 77(2):177–204, 2021.

[MP14] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

[Rei91] G. Reinelt. TSPLIB — a traveling salesman problem library. *ORSA Journal on Computing*, 3:376 – 384, 1991.

A 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, we know that 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 reason about the case of the STP with three terminals, proving the characterization of integer solutions and conjecturing the form of non-integer ones.

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

Definition 3 (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 $t = |T| = |\{w_1, \ldots, w_t\}|$ 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 subset 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 true 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. □

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 cost 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 there exists a leaf $v \in \mathcal{T}$ such that $v \notin T$. 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 there exists a node of degree three, 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 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 c and one of the terminal nodes with one of the shortest paths between c and that terminal node gives us a solution with a less or equal cost. Note also that when c is one of the terminal nodes, the shortest path between that node and c is the empty set. □

Note that such characterization has been already observed in [AdOO21]. Note also that we are only using that the costs are positive, the triangle inequality needs not to hold. Furthermore, the graph needs not 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 are now able to characterize the integer solutions of the STP with three terminals for complete metric graphs.

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

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

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. □

After several numerical tests, we were not able to produce a fractional point of $P_{\text{DCUT}}(n, 3)$ with an integrality gap greater than one, neither we were able to 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 $t \leq n$. 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, i.e. the case $t = 2$, is integral. Note that there exist solutions of the DCUT with more than three terminal nodes which are not disjoint union of shortest path from one node to he 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.