The Integrality Gap of the Traveling Salesman Problem is 4/3 if the LP Solution Has at Most $n + 6$ Non-Zero Components

Tullio Villa[1,3], Eleonora Vercesi[2,3], Janos Barta[1,3], and Monaldo Mastrolilli[1,3]

[1] Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)
[2] Università della Svizzera Italiana (USI)
[3] Dalle Molle Institute for Artificial Intelligence USI-SUPSI

Abstract. We address the classical Dantzig–Fulkerson–Johnson formulation of the symmetric metric Traveling Salesman Problem and study the integrality gap of its linear relaxation, namely the Subtour Elimination Problem (SEP). This integrality gap is conjectured to be 4/3. We prove that, when solving a problem on n nodes, if the optimal SEP solution has at most $n + 6$ non-zero components, then the conjecture is true. To establish this result, we devise a new methodology that combines theoretical analysis and computational verification.

Keywords: Symmetric Traveling Salesman Problem · Linear Programming Relaxation · Integrality Gap

1 Introduction

Let $K_n = (V_n, E_n)$ be the complete graph on n nodes, and let $\boldsymbol{c} \in \mathbb{R}^{E_n}$ be a non-negative cost vector. The *Traveling Salesman Problem* (TSP) consists of finding a Hamiltonian tour of minimum total cost. This paper focuses exclusively on the *symmetric* formulation, where the graph K_n is directed, hence we use ij and ji interchangeably to denote the edge connecting nodes i and j. Additionally, we only consider *metric* cost vectors, satisfying the triangle inequalities. Henceforth, we simply use the term TSP to indicate the symmetric metric TSP.

The TSP is typically formulated as an Integer Linear Program (ILP). Among the most prominent formulations is the one of Dantzig-Fulkerson-Johnson (DFJ) [9], whose associated relaxed Linear Problem (LP) is known in the literature as the Subtour Elimination Problem (SEP), or Held-Karp relaxation:

$$\text{DFJ}: \min_{\boldsymbol{x} \in P^n_{\text{SEP}} \cap \mathbb{Z}^{E_n}} \sum_{e \in E_n} c_e x_e \ , \qquad \text{SEP}: \min_{\boldsymbol{x} \in P^n_{\text{SEP}}} \sum_{e \in E_n} c_e x_e \ .$$

The polytope on which they are described is the *subtour elimination polytope*:

$$P^n_{\text{SEP}} := \left\{ \boldsymbol{x} \in \mathbb{R}^{E_n} \mid \sum_{e \in \delta(v)} x_e = 2 \ \ \forall v \in V_n, \ \sum_{e \in \delta(S)} x_e \geq 2 \ \ \forall S \in \mathcal{S}, \ 0 \leq x_e \leq 1 \ \ \forall e \in E_n \right\} ,$$

where $\mathcal{S} := \{S \subseteq V_n \mid 3 \leq |S| \leq n-3\}$ and,for any subset $S \subseteq V_n$, $\delta(S)$ is the set of edges having exactly one node in S [4]. The term *subtour elimination constraints* refers to the second set of constraints in the definition of P^n_{SEP}.

For a given cost vector $\boldsymbol{c}$, we denote the optimal solutions of the integral and relaxed problem by $\text{TSP}(\boldsymbol{c})$ and $\text{SEP}(\boldsymbol{c})$, respectively. The *integrality gap* is the ratio of these two values; it can be evaluated for a single instance cost $\boldsymbol{c}$, as $\alpha(\boldsymbol{c})$, or considered in a worst case scenario, as α:

$$\alpha(\boldsymbol{c}) := \frac{\text{TSP}(\boldsymbol{c})}{\text{SEP}(\boldsymbol{c})} \ , \qquad \alpha := \sup_{\boldsymbol{c} \text{ metric}} \frac{\text{TSP}(\boldsymbol{c})}{\text{SEP}(\boldsymbol{c})} \ .$$

An alternative granularity of the integrality gap can also be defined by focusing on a specific vertex $\boldsymbol{x} \in P^n_{\text{SEP}}$. We denote this by $\text{Gap}(\boldsymbol{x})$, and it is given by:

$$\text{Gap}(\boldsymbol{x}) := \sup \left\{ \frac{\text{TSP}(\boldsymbol{c})}{\text{SEP}(\boldsymbol{c})} \ \mid \ \boldsymbol{c} \text{ metric} \ , \ \boldsymbol{x} \in \arg\min \text{SEP}(\boldsymbol{c}) \right\} \ .$$

The exact value of α is currently unknown; in order to determine it, various approaches have been explored, with increasing interest over the past decade. Wolsey [23] proposed an upper bound of $3/2$, which remains the best known to date. Already in 1995, Goemans [11] conjectured that the integrality gap was equal to $4/3$, and despite several attempts, nobody was able to disprove that conjecture. Later, [1,3] exploited exhaustive enumeration of the vertices of P^n_{SEP} to compute the exact value of the integrality gap for TSP instances with $n \leq 12$. Other than that, only results for specific subclasses of instances are available in the literature. Schalekamp, Williamson, and van Zuylen [21] conjectured that the maximum integrality gap is attained on *half-integer* instances, that is, cost vectors whose optimal SEP solutions have all the entries in $\{0, 1/2, 1\}$. In recent years, promising lines of research have examined subclasses of SEP solutions $\boldsymbol{x}$ characterized by specific properties of the so called *support graph*, defined as the undirected weighted graph $G_{\boldsymbol{x}} = (V_n, E_{\boldsymbol{x}})$ such that $ij \in E_{\boldsymbol{x}} \Leftrightarrow x_{ij} > 0$, and the weight on edge ij is given by x_{ij}. Notice that the number of edges in the support graph $G_{\boldsymbol{x}}$ is, by definition, the number of non-zero components of $\boldsymbol{x}$. Boyd and Carr [2] proved that the integrality gap is $4/3$ when the support graph of the SEP solution contains disjoint $1/2$ triangles. For graph-TSP instances[5], Boyd et al. [6] proved that the $4/3$-conjecture holds for cubic graphs. Mömke and Svensson [20] improved this result showing that the integrality gap is $4/3$ for graph-TSP restricted either to half-integral solutions or to a class of graphs that contains subcubic and claw-free graphs. Boyd and Sebő [5] proved that the integrality gap is at most ~ 1.4286 for instances having as SEP solution one of the so-called Boyd-Carr points (firstly defined in [2]). Jin et al. [16] proved the $4/3$-conjecture for instances having as SEP solution cycle-cut points, that is, points for which

[4] The brackets denoting singletons are omitted for brevity, e.g., $v = \{v\}$; this abuse of notation is used throughout this work, when the context does not lead to ambiguity.

[5] Graph-TSP is a subclass of metric TSP where the distance between the nodes is computed as the minimum number of edges separating them.

every non-singleton tight set can be written as the union of two tight sets. With a breakthrough result [18], Karlin, Klein, and Oveis Gharan gave an approximation algorithm for the general TSP; building on this work, the same authors were able to prove an integrality gap smaller than $3/2$ in the half-integer case [17]. This factor was improved in [12] and subsequently in the as-yet unpublished work [19].

Different approaches have also been attempted to improve the lower bound. In [1,14,15,24], families of TSP instances with high integrality gap, asymptotically tending to $4/3$ were shown. In [22], the authors proposed an approach to *heuristically* generate instances with a high integrality gap: no instance with an IG greater than $4/3$ was found.

Contribution and outlook. This work presents both a theoretical contribution and a novel methodology that opens new research directions in the study of the integrality gap. We prove the $4/3$-conjecture for all the vertices of P^n_{SEP} whose support graph contains at most $n+6$ edges. Remarkably, this class is rich enough to contain vertices that are not covered by any previously known result in the literature. Figure 1 shows an example of such a vertex.

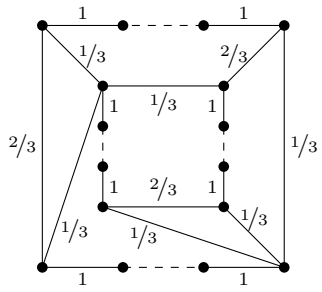

Fig. 1. Example of a vertex considered in this work but not in previous literature. Dashed edges may be replaced by arbitrarily long paths of edges of weight 1.

To establish our result, we define, for each integer k, the family $\mathcal{F}_k$ which comprises, across all polytopes P^n_{SEP} for $n \in \mathbb{N}$, the vertices whose support graphs contain exactly $n+k$ edges. Although the family $\mathcal{F}_k$ is infinite, we identify a finite subset $\mathcal{A}_k$ of its elements, which we call *ancestors*, that allows us to make overall considerations on the entire family. We devise the *Gap-Bounding (GB) algorithm* and prove that its application on these ancestors returns upper bounds on the integrality gap of all the costs whose SEP solution is a vertex of $\mathcal{F}_k$. The application of the GB algorithm on the elements of $\mathcal{A}_k$ with $k \leq 6$ yields a computer-aided proof of the desired result.

The methodology presented constitutes in itself a novel way to approach the integrality gap problem, combining theoretical analysis and computational verification. Unlike earlier computational proofs (see [1,3]) that were limited to enumerating vertices for a fixed dimension n, our framework fixes a small

parameter k and derives results valid for *all* n, hereby achieving a form of universality through a computational approach. To the best of our knowledge, this is the first time that a computer-aided proof is implied in bounding the integrality gap for *infinitely many* vertices.

We conclude by noting that the vast majority of instances used in the literature to establish lower bounds on the integrality gap are special cases of our result (see, e.g., [1,14,15,24][6]). Other families, however, fall outside the scope of our analysis: the donut instances proposed in [5] have a non-constant surplus of edges over nodes, and the lower bounds on their integrality gap provided in the article converge to $4/3$. Interestingly, for each donut instance, the mentioned lower bound is consistently dominated by the integrality gap of an instance of $\mathcal{F}_3$ with the same number of nodes[7]. This observation aligns with the empirical evidence that, for a small number of nodes $n \leq 12$, the value of the integrality gap of a vertex correlates with the number of edges in its support graph (more details in Section 7). Taken together, these insights motivate the following hypothesis, which highlights the significance of the class of vertices considered in this work.

Few Edges Hypothesis. *For every n, among the vertices of P^n_{SEP}, the maximum integrality gap is realized on a vertex with the minimum possible number of edges in the support graph, that is $n + 3$.*

In summary, our work contributes to three different fronts.

1. It establishes the $4/3$ conjecture for newly identified infinite classes of vertices.
2. It demonstrates the potential of computer-aided proofs to advance our understanding of the integrality gap across infinite vertex classes, proposing a flexible framework that can be adapted to other contexts.
3. It opens the compelling research direction of investigating the correlation between the integrality gap and the number of edges in the support graph. A proof of the Few Edges Hypothesis, together with the contribution of this work, would result in definitively solving the $4/3$ conjecture.

Outline. Section 2 introduces key definitions from the literature. Section 3 presents the concept of ancestors together with the procedure to retrieve them. Section 4 provides the theoretical basis for the GB algorithm, whose design is detailed in Section 5. Section 6 explains how to apply the GB algorithm to bound the integrality gap for vertices with up to $n + 6$ non-zero components. Section 7 explores future research directions. Proofs and technical details are included in the appendix.

[6] Counting nodes and edges in the constructions, one may see that the instances in [1,14,24] belong to $\mathcal{F}_3$ and the instances in [15] belong to $\mathcal{F}_6$.

[7] It can be checked comparing the explicit formulas for the lower bounds on the integrality gap in [1] and [5].

2 Background material

2.1 Preliminaries from graph theory

Before giving the list of definitions used in this work, we clarify that, to avoid confusion, given a graph $G = (V, E)$, the term *node* refers to the elements of V; the term *vertex* is reserved exclusively for the extreme points of P^n_{SEP}.

Definition 1 (Hamiltonian walk, Hamiltonian tour). *Let G be an undirected graph. A* Hamiltonian walk *is a closed walk in G that visits every node at least once, possibly traversing the same edge multiple times. If every node is visited exactly once, we call the walk* Hamiltonian tour*. For a given walk $\boldsymbol{w}$, $w_{ij} \in \mathbb{N}$ denotes the* multiplicity *of ij in $\boldsymbol{w}$, that is, the number of times the walk $\boldsymbol{w}$ uses the edge ij (the analogous notation $t_{ij} \in \{0, 1\}$ is used for a tour $\boldsymbol{t}$).*

For brevity, in what follows the terms *walk* and *tour* always indicate Hamiltonian walks and tours. Moreover, we use the same literal to denote both the walk $\boldsymbol{w}$ (respectively the tour $\boldsymbol{t}$) and its *characteristic vector*, that is, the vector of w_{ij} (respectively t_{ij}) entries. When the underlying graph G is not specified, we assume it is the complete one.

Remark 1. Since we have interest in the shortest possible walks $\boldsymbol{w}$, we assume that every edge ij is traversed at most twice (i.e. $w_{ij} \in \{0, 1, 2\}$)[8]. Notice that, under this assumption, the number of walks considered on a graph becomes finite.

2.2 Preliminaries from the literature on the integrality gap

In this section, we collect from the literature (mainly from [1]) the definitions and results are used in this manuscript.

Theorem 1 ([1,4]). *Let $\boldsymbol{x} \in P^n_{SEP}$ be a vertex. Then $|E_{\boldsymbol{x}}| \leq 2n - 3$.*

Definition 2 (1-edge, 1-path, from [1]). *Let $\boldsymbol{x} \in P^n_{SEP}$ and let e be an edge of K_n; e is called* 1-edge *of $\boldsymbol{x}$ if $x_e = 1$. When $\boldsymbol{x}$ is fractional (that is, not a tour), we call* 1-path *of $\boldsymbol{x}$ a maximal path of 1-edges in the support graph $G_{\boldsymbol{x}}$; the nodes of degree 2 in the 1-path are called* internal nodes*, the two remaining nodes are called* end nodes*.*

Theorem 2 ([1]). *Let $\boldsymbol{x} \in P^n_{SEP}$ be a fractional vertex. Then $\boldsymbol{x}$ has at least three distinct 1-paths.*

We now introduce a construction presented in [1] and give it a name using the authors' initials.

[8] It is well known (see, e.g. [8]) that, when an edge is used strictly more than twice by a walk, it is possible to get rid of two copies of that edge to obtain a shorter walk.

Definition 3 (BB-move, from [1]). *Let $\boldsymbol{x} \in P^n_{SEP}$ and let ab be one of its 1-edges. We call* BB-move *the construction of a new point $\boldsymbol{x}' \in P^{n+1}_{SEP}$ defined as follows, where w is the new node added ($V_{n+1} = V_n \cup \{w\}$):*

$$\begin{array}{ll} x'_{ab} = 0 \ , & x'_e = 0 \quad \forall e \in \delta(w) \setminus \{aw, wb\} \ , \\ x'_{aw} = x'_{wb} = 1 \ , & x'_e = x_e \quad \forall e \notin \delta(w) \cup \{ab\} \ . \end{array}$$

We denote this construction by $\text{BB}(\boldsymbol{x}, ab) := \boldsymbol{x}'$. When it is clear from the context or not strictly necessary, we omit the edge ab in the notation: $\text{BB}(\boldsymbol{x})$.

Theorem 3 ([1]). *Let $\boldsymbol{x} \in P^n_{SEP}$ and let ab be one of its 1-edges. Then $\text{BB}(\boldsymbol{x}, ab)$ is a vertex of P^{n+1}_{SEP} if and only if $\boldsymbol{x}$ is a vertex of P^n_{SEP}.*

3 Families of vertices with a given number of edges

In this section, we study the vertices of P^n_{SEP} considering, in their support graph, the relation between the number of edges and the number of nodes.

Lemma 1. *Let $\boldsymbol{x}$ be a fractional vertex of P^n_{SEP}. Then $|E_{\boldsymbol{x}}| \geq n + 3$.*

Our aim now is to describe, for a given k, all possible fractional vertices with n nodes and $n + k$ edges. For this purpose, we define the families of vertices $\mathcal{F}_k$:

$$\mathcal{F}_k := \{\boldsymbol{x} \text{ fractional vertex } \mid \ \boldsymbol{x} \in P^n_{\text{SEP}} \text{ for some } n, \ |E_{\boldsymbol{x}}| = n + k\} \ .$$

Notice that Lemma 1 implies $k \geq 3$, hence $\mathcal{F}_1 = \mathcal{F}_2 = \emptyset$. Observe also that a BB-move increases both the number of nodes and edges by 1, thus preserving their difference k. Therefore, in virtue of Theorem 3, a vertex is in $\mathcal{F}_k$ if and only if its image under a BB-move is in $\mathcal{F}_k$ as well. This brings us to define the *ancestors*, the vertices of $\mathcal{F}_k$ that can not be obtained as results of a BB-move, that is, vertices without internal nodes in the 1-paths.

Definition 4 (Ancestor of order k). *An* ancestor of order k *is a vertex $\boldsymbol{x}$ of $\mathcal{F}_k$ with no node of degree 2. We denote the set of ancestors of order k as*

$$\mathcal{A}_k := \{\boldsymbol{x} \in \mathcal{F}_k \mid \boldsymbol{x} \text{ has no node of degree } 2\} \ .$$

Definition 5 (Successor). *Given a vertex $\boldsymbol{x}$, we call* successor *of $\boldsymbol{x}$ any vertex $\boldsymbol{x}'$ obtained by sequential applications of the* BB-move.

We can thus recover the whole family $\mathcal{F}_k$ by taking the elements of $\mathcal{A}_k$ and replacing any 1-edge with a 1-path of arbitrary length; $\mathcal{F}_k$ may be seen as the set of successors of $\mathcal{A}_k$. At this point, to give a complete description of $\mathcal{F}_k$, it only remains to retrieve all the elements of $\mathcal{A}_k$.

Lemma 2. *Let $\boldsymbol{x}$ be an ancestor in $\mathcal{A}_k$ and let n be its number of nodes. Then $k + 3 \leq n \leq 2k$.*

Therefore, the elements of $\mathcal{A}_k$ are finite and can be recovered from the lists of vertices of P^n_{SEP} with n up to $2k$ (provided we have them at our disposal). We simply need to scroll through the lists of vertices of P^n_{SEP} for all the $n = k+3, \ldots, 2k$, and extract the ones with exactly $n+k$ edges and no node of degree 2. Since, at the time being, an exhaustive list of fractional vertices (up to isomorphism[9]) is already available for n up to 12 [1,3], it is possible to completely determine $\mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6$ (up to isomorphism) by Lemma 2. For instance, Figure 2 shows all the ancestors in $\mathcal{A}_4$: as $k = 4$, they can be extracted from the lists of vertices of P^7_{SEP} and P^8_{SEP}. All the vertices of $\mathcal{F}_4$ are either of these shapes or with the 1-edges replaced with 1-paths of arbitrary length. Similarly, simply examining the structure of the sole ancestor in $\mathcal{A}_3$, one can deduce that the family $\mathcal{F}_3$ consists precisely of those vertices formed by two 1/2-triangles connected by three 1-paths. In virtue of this characterization, one may see that Conjecture 4.1 in [1] supports the Few Edge Hypothesis.

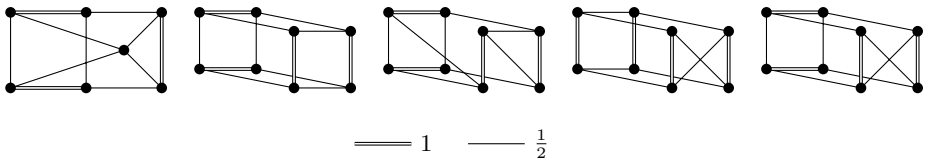

Fig. 2. Vertices of $\mathcal{A}_4$, up to isomorphism.

4 Redefining the Gap problem

In this section, we formulate a relaxed definition of Gap.

Definition 6 (Gap$^+$). *Let $\boldsymbol{x} \in P^n_{SEP}$ be a vertex. The* Gap$^+$ *of $\boldsymbol{x}$ is*

$$Gap^+(\boldsymbol{x}) := \sup \left\{ \frac{TSP(\boldsymbol{c})}{\boldsymbol{cx}} \;\middle|\; \boldsymbol{c} \text{ metric} \right\} .$$

By removing the requirement $\boldsymbol{x} \in \arg\min \text{SEP}(\boldsymbol{c})$, we obtain the inequality:

$$\text{Gap}^+(\boldsymbol{x}) \geq \text{Gap}(\boldsymbol{x}) . \tag{1}$$

Our objective is therefore to provide an upper bound on Gap$^+$, which consequently yields a valid upper bound for Gap. To compute the inverse of Gap($\boldsymbol{x}$), Benoit and Boyd, in [1], designed an LP named OPT($\boldsymbol{x}$). Adopting the same strategy, for a given a vertex $\boldsymbol{x}$, we consider a corresponding linear problem OPT$^+$($\boldsymbol{x}$),

[9] An isomorphism between vertices is a relabeling of their nodes.

defined as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{ij \in E_n} x_{ij} c_{ij} && && (2)\\
\text{subject to:} \quad & c_{ik} + c_{jk} - c_{ij} \ \geq 0 && \forall ij \in E_n,\ k \neq i, j, && (3)\\
& \sum_{ij \in E_n} t_{ij} c_{ij} \ \geq 1 && \forall \boldsymbol{t} \text{ tour}, && (4)\\
& c_{ij} \geq 0 && \forall ij \in E_n. && (5)
\end{aligned}$$

Constraints (3) force $\boldsymbol{c}$ to be a metric cost; constraints (4) normalize $\mathrm{TSP}(\boldsymbol{c}) = 1$; minimizing $\boldsymbol{cx}$ is then equivalent to maximizing $1/\boldsymbol{cx} = \mathrm{TSP}(\boldsymbol{c})/\boldsymbol{cx}$, thus

$$\frac{1}{\mathrm{OPT}^+(\boldsymbol{x})} = \mathrm{Gap}^+(\boldsymbol{x}) \ . \tag{6}$$

The following lemma is the first result that studies the behavior of Gap^+ under the application of the BB-move.

Lemma 3. *Let $\boldsymbol{x} \in P^n_{SEP}$. Then $Gap^+(\,\mathrm{BB}(\boldsymbol{x})\,) \geq Gap^+(\boldsymbol{x})$. That is, the* BB*-move is Gap^+-increasing.*

5 The Gap-Bounding algorithm

This section is entirely devoted to finding a way to bound the Gap^+ for all the vertices of a given family $\mathcal{F}_k$. To this end, we aim to contain the increase in Gap^+ originated by an iterative application of a BB-move.

We begin by outlining the central idea of our approach. Our goal is to provide a lower bound on OPT^+, which results in an upper bound on Gap^+ by (6). Assume that we are given a vertex $\boldsymbol{x}_0$ alongside with an optimal solution of $\mathrm{OPT}^+(\boldsymbol{x}_0)$ and a corresponding optimal solution $\boldsymbol{\mu}^0$ of the dual problem $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x}_0)$. When considering a successor $\boldsymbol{x}'$ of $\boldsymbol{x}_0$, we intend to construct a feasible dual solution $\boldsymbol{\mu}'$ for $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x}')$ starting from $\boldsymbol{\mu}^0$: succeeding in this task will directly result in a lower bound for $\mathrm{OPT}^+(\boldsymbol{x}')$, as guaranteed by duality theory.

Indeed, for every vertex $\boldsymbol{x}$, equations (1), (6), and duality theory yield

$$\mathrm{Gap}(\boldsymbol{x}) \leq \mathrm{Gap}^+(\boldsymbol{x}) = \frac{1}{\mathrm{OPT}^+(\boldsymbol{x})} = \frac{1}{\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})} \leq \frac{1}{L} \ , \tag{7}$$

where L is any lower bound on $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})$. In our case, L will be a bound on the objective value attained by $\boldsymbol{\mu}'$ (elaborated from $\boldsymbol{\mu}^0$). This strategy is visually represented in Figure 3: it will be the skeleton of the Gap-Bounding algorithm, discussed in detail in Section 5.2.

Clearly, the crucial point in this procedure is to find a "good-enough" feasible dual solution, so that the bound obtained is meaningful (e.g., the trivial dual solution of all zeros ultimately gives no bound on Gap^+).

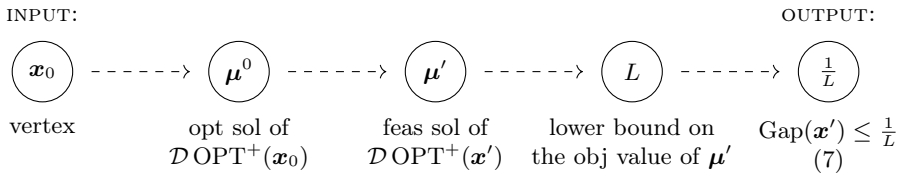

Fig. 3. Flowchart of the pipeline applied to bound Gap for a successor $\boldsymbol{x}'$ of $\boldsymbol{x}_0$.

5.1 Dual formulations of the OPT$^+$ problem

Given a vertex $\boldsymbol{x}$, the dual problem $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})$ is derived as follows.

$$\text{maximize} \quad \sum_{\boldsymbol{t}\ \text{tour}} \mu_{\boldsymbol{t}} \tag{8}$$

$$\text{subject to:} \quad \sum_{k\neq i,j} (-\lambda_{ijk} + \lambda_{ikj} + \lambda_{jki}) + \sum_{\boldsymbol{t}\ \text{tour}} t_{ij}\mu_{\boldsymbol{t}} \le x_{ij} \quad \forall ij \in E_n, \tag{9}$$

$$\lambda_{ijk} \ge 0 \qquad \forall ij \in E_n,\ k \neq i,j, \tag{10}$$

$$\mu_{\boldsymbol{t}} \ge 0 \qquad \forall \boldsymbol{t}\ \text{tour on } K_n. \tag{11}$$

Notice that the abuse of notation $ij \equiv ji \in E$ induces also $\lambda_{ijk} \equiv \lambda_{jik}$; this fact does not involve the third index: $\lambda_{ijk} \not\equiv \lambda_{kji}$ [10].

When considering a vertex $\boldsymbol{x}$ and one of its successors $\boldsymbol{x}'$, the task of producing a feasible solution of $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x}')$ starting from an optimal solution of $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})$ is anything but trivial. In what follows, we design an equivalent formulation and make use of it instead: $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$. Its main purpose is to get rid of $\boldsymbol{\lambda}$ variables: this is achieved at the price of considering walks on $G_{\boldsymbol{x}}$ (see Definition 1 and Remark 1) in place of tours. For a given vertex $\boldsymbol{x}$, $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ is thus defined as:

$$\text{maximize} \quad \sum_{\boldsymbol{w}\ \text{walk on}\ G_{\boldsymbol{x}}} \mu_{\boldsymbol{w}} \tag{12}$$

$$\text{subject to:} \quad \sum_{\boldsymbol{w}\ \text{walk on}\ G_{\boldsymbol{x}}} w_{ij}\mu_{\boldsymbol{w}} \le x_{ij} \qquad \forall ij \in E_{\boldsymbol{x}}, \tag{13}$$

$$\mu_{\boldsymbol{w}} \ge 0 \qquad \forall \boldsymbol{w}\ \text{walk on } G_{\boldsymbol{x}}. \tag{14}$$

The equivalence of the two formulations $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})$ and $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ (that is, they have the same optimal value) is stated in the following lemma.

Lemma 4. *$\mathcal{D}\ OPT^+(\boldsymbol{x}) = \mathcal{D}\ OPT^{II}(\boldsymbol{x})$ for every vertex $\boldsymbol{x}$ of P^n_{SEP}.*

5.2 Bounding the Gap on successors

The goal of this section is to derive, for a generic successor $\boldsymbol{x}'$ of $\boldsymbol{x}_0$, a feasible solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}')$, starting from an optimal solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)$; this serves as a bound on $\mathrm{Gap}^+(\boldsymbol{x}')$ (see Figure 3, with $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$ in place of $\mathcal{D}\,\mathrm{OPT}^+$).

[10] With the symbol $\equiv$, we do not mean that two distinct variables have the same value; rather, we mean that the two notations coincide: they denote the same variable.

Let $\boldsymbol{x}_0$ be a vertex of P^n_{SEP} and let $\boldsymbol{\mu}^0 \in \arg\max \mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}_0)$. We begin by considering a 1-edge ab of $\boldsymbol{x}_0$ and applying d consecutive BB-moves (see Definition 3): we insert nodes $a_1, \ldots, a_d$ and obtain, by Theorem 3, the vertices $\boldsymbol{x}_k := \text{BB}(\boldsymbol{x}_{k-1},\ a_{k-1}b)$ for $k = 1, \ldots, d$ (where $a_0 := a$). We aim to design a feasible solution for $\mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}_d)$.

An assignment $\boldsymbol{\mu}^d$ for $\mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}_d)$ is indexed by the walks on $G_{\boldsymbol{x}_d}$. We construct walks $\boldsymbol{w}^d$ on $G_{\boldsymbol{x}_d}$, starting from the ones on $G_{\boldsymbol{x}_0}$, and assign weights to the corresponding components of $\boldsymbol{\mu}^d$. In this perspective, we divide the walks on $G_{\boldsymbol{x}_0}$ into three families, according to how many times the edge ab is traversed: we define $\mathcal{W}^{ab}_m := \{\boldsymbol{w} \text{ walk } \mid\ w_{ab} = m\}$ for $m = 0, 1, 2$.

If $\boldsymbol{w}^0 \in \mathcal{W}^{ab}_0$, we construct the following $d+1$ walks on $G_{\boldsymbol{x}_d}$. For each $k = 0, \ldots, d$, the walk $\boldsymbol{w}^d_k$ retraces $\boldsymbol{w}^0$ but, when it arrives at a, it deviates to pass through $a_1, a_2, \ldots, a_k$ and back, and when it arrives at b, it deviates to $a_d, a_{d-1}, \ldots, a_{k+1}$ and back. We set $\mu^d_{\boldsymbol{w}^d_k} := \frac{1}{d+1}\mu^0_{\boldsymbol{w}^0}$ for all $k = 0, \ldots, d$.

If $\boldsymbol{w}^0 \in \mathcal{W}^{ab}_1$, we construct the walk $\boldsymbol{w}^d$ on $G_{\boldsymbol{x}_d}$ that retraces $\boldsymbol{w}^0$ but replaces ab with $aa_1, a_1a_2, \ldots, a_{d-1}a_d, a_db$. We set $\mu^d_{\boldsymbol{w}^d} := \mu^0_{\boldsymbol{w}^0}$.

If $\boldsymbol{w}^0 \in \mathcal{W}^{ab}_2$, we construct the walk $\boldsymbol{w}^d$ on $G_{\boldsymbol{x}_d}$ that retraces $\boldsymbol{w}^0$ but replaces the double passage on ab by passing twice through $aa_1, a_1a_2, \ldots, a_{d-1}a_d, a_db$. We set $\mu^d_{\boldsymbol{w}^d} := \mu^0_{\boldsymbol{w}^0}$.

Notice that these constructions cover all the possible walks $\boldsymbol{w}^d$ on $G_{\boldsymbol{x}_d}$. Figure 4 illustrates the new assignment $\boldsymbol{\mu}^d$ for $d = 2$.

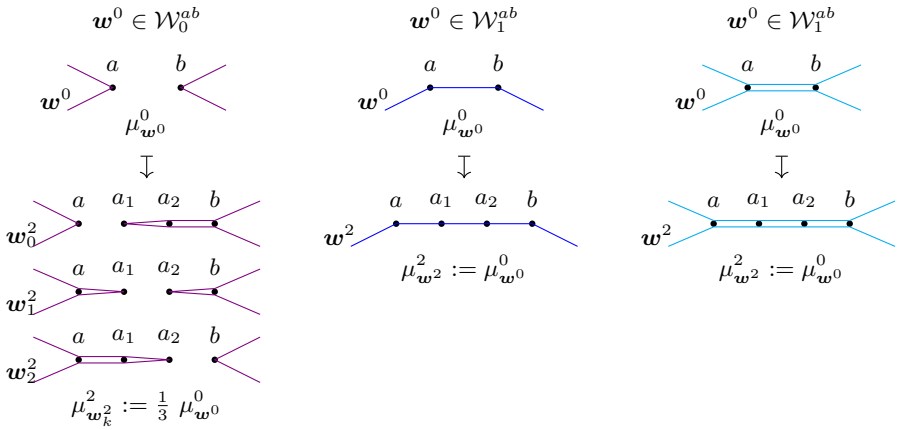

Fig. 4. Variables $\boldsymbol{\mu}^2$ for $\mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}_2)$ constructed from variables $\boldsymbol{\mu}^0$ for $\mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}_0)$.

In order to extend the construction for a generic successor of $\boldsymbol{x}_0$, we sequentially repeat the previous procedure for each 1-edge $e_1, \ldots, e_p$ of $\boldsymbol{x}_0$, expanding them into 1-paths with $d_1, \ldots, d_p$ internal nodes. This yields an assignment $\boldsymbol{\mu}'$ for $\mathcal{D}\,\text{OPT}^{\text{II}}(\boldsymbol{x}')$ such that: (i) $\boldsymbol{\mu}'$ attains the same objective value as $\boldsymbol{\mu}^0$; (ii) $\boldsymbol{\mu}'$ satisfies all the constraints (13), except those associated to the 1-edges, which

nonetheless remain bounded by fixed constants. These constants are the same for all the edges in the same 1-path of $\boldsymbol{x}'$, originated from the same 1-edge e_h of $\boldsymbol{x}_0$:

$$C(\boldsymbol{x}_0, e_h) := 2 \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{e_h}} \mu^0_{\boldsymbol{w}^0} \; + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{e_h}} \mu^0_{\boldsymbol{w}^0} \; + \; 2 \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{e_h}} \mu^0_{\boldsymbol{w}^0} \; .$$

Crucially, they only depend on the initial vertex $\boldsymbol{x}_0$, not on the lengths of the 1-paths of $\boldsymbol{x}'$. Rescaling by the factor $C^*(\boldsymbol{x}^0) := \max_{h=1,\dots,p}\{C(\boldsymbol{x}_0, e_h)\}$, we thus obtain the feasible solution $\boldsymbol{\mu}^* := \frac{1}{C^*(\boldsymbol{x}^0)}\boldsymbol{\mu}'$, whose objective value accordingly becomes $\frac{1}{C^*(\boldsymbol{x}^0)}\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)$. This value plays the role of L in eq. (7), yielding $\mathrm{Gap}^+(\boldsymbol{x}') \;\leq\; C^*(\boldsymbol{x}^0) \cdot \frac{1}{\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)}$, and ultimately, by eq. (6) and Lemma 4,

$$\mathrm{Gap}^+(\boldsymbol{x}') \;\leq\; C^*(\boldsymbol{x}^0) \cdot \mathrm{Gap}^+(\boldsymbol{x}_0) \; .$$

Notice that we expect $C^*(\boldsymbol{x}^0)$ to be greater than or equal to 1, otherwise we would obtain $\mathrm{Gap}^+(\boldsymbol{x}_d) < \mathrm{Gap}^+(\boldsymbol{x}_0)$, contradicting Lemma 3.

We are finally able to design the Gap-Bounding (GB) algorithm as Algorithm 1; its main purpose is clarified in the next theorem.

Algorithm 1 Gap-Bounding algorithm

1: INPUT: $\boldsymbol{x} \in P^n_{\mathrm{SEP}}$ vertex
2: Solve $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ and retrieve an optimal assignment of variables $\{\mu_{\boldsymbol{w}}\}_{\boldsymbol{w}\text{ walk on }\boldsymbol{x}}$.
3: **for** each e 1-edge of $\boldsymbol{x}$ **do**
4: $\quad C(\boldsymbol{x}, e) \;\leftarrow\; 2 \sum_{\boldsymbol{w} \in \mathcal{W}_0^e} \mu_{\boldsymbol{w}} + \sum_{\boldsymbol{w} \in \mathcal{W}_1^e} \mu_{\boldsymbol{w}} + 2 \sum_{\boldsymbol{w} \in \mathcal{W}_2^e} \mu_{\boldsymbol{w}}$
5: **end for**
6: $C^*(\boldsymbol{x}) \;\leftarrow\; \max\{C(\boldsymbol{x}, e)\}_{e\text{ 1-edge of }\boldsymbol{x}}$
7: $\mathrm{Gap}^+(\boldsymbol{x}) \;\leftarrow\; {}^1\!/_{\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})}$
8: Return: $C^*(\boldsymbol{x}) \cdot \mathrm{Gap}^+(\boldsymbol{x})$

Theorem 4. *The Gap-Bounding algorithm, on input a vertex $\boldsymbol{x} \in P^n_{SEP}$, returns a value $GB(\boldsymbol{x})$ which is an upper bound for the gap of all the successors $\boldsymbol{x}'$ of $\boldsymbol{x}$: $Gap(\boldsymbol{x}') \leq GB(\boldsymbol{x})$.*

Proof. This whole section is the proof that $\mathrm{Gap}^+(\boldsymbol{x}') \leq C^*(\boldsymbol{x}) \cdot \mathrm{Gap}^+(\boldsymbol{x}) = \mathrm{GB}(\boldsymbol{x})$ for any successor $\boldsymbol{x}'$ of $\boldsymbol{x}$; equation (1) completes the argument. □

Notice that both the GB algorithm and Theorem 4 apply to all generic $\boldsymbol{x}$ vertices, without requiring that they be ancestors. The following lemma shows how to exploit this fact.

Lemma 5. *Let $\boldsymbol{x} \in P^n_{SEP}$ be a vertex and let $\boldsymbol{x}_0$ be its ancestor. Then $GB(\boldsymbol{x})$ gives a bound for the gap of all the successors $\boldsymbol{x}'$ of $\boldsymbol{x}_0$ (even for $\boldsymbol{x}'$ that are not successors of $\boldsymbol{x}$): $Gap(\boldsymbol{x}') \leq GB(\boldsymbol{x})$.*

6 Computational results

This section reports how the execution of the Gap-bounding algorithm yields the *computer-aided* proof of our main result: Theorem 5.

Our pipeline starts by extracting the ancestors of $\mathcal{A}_k$ for $k = 3, 4, 5, 6$ from the lists of vertices already available from [1,3][11], as explained in Section 3. Subsequently, the Gap-Bounding algorithm is applied on each vertex $\boldsymbol{x} \in \mathcal{A}_k$.

We inform that, for some ancestors, a straightforward application of the GB algorithm returned a value higher than $4/3$. To improve the bounds returned, we exploited Lemma 5 and executed the GB algorithm on some selected successors. Even though there is no guarantee that the GB algorithm applied to successors gives strictly better results, we thought it reasonable that more "information" (more nodes, edges, and dual variables) would lead to a more detailed analysis and a tighter bound. Eventually, this intuition led us to accomplish our objective.

Theorem 5. *$Gap(\boldsymbol{x}) \leq \frac{4}{3}$ for all the vertices of $\boldsymbol{x} \in \mathcal{F}_k$ with $k = 3, 4, 5, 6$.*

We implemented the GB algorithm in Python, employing the commercial software Gurobi [13] to solve the LPs. Our code is available here. The computation is relatively lightweight and can be easily performed on a standard laptop.

7 Conclusion

In this paper, the main result of proving the $4/3$-conjecture for vertices of P^n_{SEP} with at most $n + 6$ edges in their support graphs is presented. A natural continuation of this research consists of constructing $\mathcal{A}_7$ by exhaustive enumeration, to enlarge the set of ancestors on which to apply the GB algorithm and collect bounds for the whole family $\mathcal{F}_7$.

Besides this, the new *methodology* introduced is itself of independent interest, as it opens a new range of possible advancements in the field. Investigating novel transformations between vertices and analyzing their impact on the integrality gap (akin to the analysis we presented for the BB-move) may lead to interesting results on the integrality gap for diverse infinite families of vertices.

To conclude, we also mention a last research direction, motivated by an intriguing, computationally verified observation. Using data from [1], for instances with few nodes, we noticed a correlation between the number of edges in the support graphs of the vertices of P^n_{SEP} (with $6 \leq n \leq 12$) and their integrality gap. As shown in Figure 5, for a fixed $n \leq 12$, the integrality gap is higher on vertices with few non-zero components and appears to decrease as the number of edges in the support graph increases. Although this correlation is currently supported only for small values of n, it presents a fascinating avenue for further investigation. For example, the donut instances of [5] have an integrality gap tending to $4/3$, and they do not belong to the families of instances considered in our result; nevertheless, for each donut instance, the proposed lower bound on

[11] Vertices data can be found at this link, last visited 04.06.2025.

its integrality gap is strictly smaller than the integrality gap of a vertex of $\mathcal{F}_3$ with the same number of nodes. Should one succeed in proving that, for every n, the maximum integrality gap on the polytope P^n_{SEP} is always attained on the vertices with the smallest number of edges, such a result together with the contribution of this work would be the two key ingredients to definitely resolve the $4/3$-conjecture.

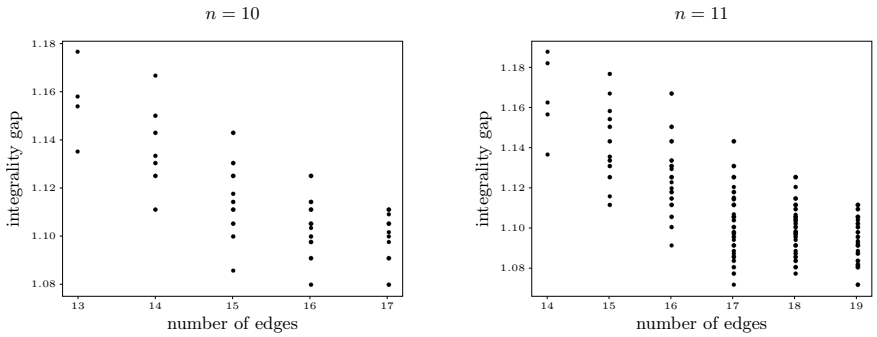

Fig. 5. Correlation between the number of edges in the support graph of a vertex and their integrality gap. Each plot collects all the vertices of P^n_{SEP} for a fixed n: $n = 10$ on the left, $n = 11$ on the right. The observed trend is the same for all small $n \leq 12$.

References

1. Benoit, G., Boyd, S.: Finding the Exact Integrality Gap for Small Traveling Salesman Problems. Mathematics of Operations Research **33**, 921–931 (11 2008). https://doi.org/10.1287/moor.1080.0337
2. Boyd, S., Carr, R.: Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. Discrete Optimization **8**(4), 525–539 (2011). https://doi.org/10.1016/j.disopt.2011.05.002, https://www.sciencedirect.com/science/article/pii/S1572528611000302
3. Boyd, S., Elliott-Magwood, P.: Structure of the Extreme Points of the Subtour Elimination Polytope of the STSP. In: RIMS Kokyuroku Bessatsu. pp. 33–47 (12 2010), https://api.semanticscholar.org/CorpusID:15182527
4. Boyd, S., Pulleyblank, W.R.: Optimizing over the subtour polytope of the travelling salesman problem. Mathematical Programming **49**, 163–187 (11 1990), https://doi.org/10.1007/BF01588786
5. Boyd, S., Sebő, A.: The salesman's improved tours for fundamental classes. Mathematical Programming **186**(1), 289–307 (2021)
6. Boyd, S., Sitters, R., van der Ster, S., Stougie, L.: TSP on Cubic and Subcubic Graphs. In: Integer Programming and Combinatoral Optimization. pp. 65–77. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
7. Conforti, M., Cornuéjols, G., Zambelli, G.: Integer programming models. Springer (2014)
8. Cornuejols, G., Fonlupt, J., Naddef, D.: The traveling salesman problem on a graph and some related integer polyhedra. Mathematical Programming **33**, 1–27 (09 1985), https://doi.org/10.1007/BF01582008
9. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America **2**(4), 393–410 (1954)
10. Gale, D., Kuhn, H.W., Tucker, A.W.: Linear Programming and the Theory of Games. In: Koopmans, T.C. (ed.) Activity Analysis of Production and Allocation, pp. 317–329. Wiley, New York, NY (1951)
11. Goemans, M.X.: Worst-case comparison of valid inequalities for the tsp. Mathematical Programming **69**(1), 335–349 (1995)
12. Gupta, A., Lee, E., Li, J., Mucha, M., Newman, H., Sarkar, S.: Matroid-based TSP rounding for half-integral solutions. Mathematical Programming **206**(1), 541–576 (2024)
13. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2024), https://www.gurobi.com
14. Hougardy, S.: On the integrality ratio of the subtour LP for Euclidean TSP. Operations Research Letters **42**(8), 495–499 (2014)
15. Hougardy, S., Zhong, X.: Hard to solve instances of the Euclidean Traveling Salesman Problem. Mathematical Programming Computation **13**, 51–74 (2021)
16. Jin, B., Klein, N., Williamson, D.P.: A 4/3-approximation algorithm for half-integral cycle cut instances of the TSP. Mathematical Programming **209** (02 2025). https://doi.org/10.1007/s10107-025-02193-5
17. Karlin, A.R., Klein, N., Oveis Gharan, S.: An improved approximation algorithm for TSP in the half integral case. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. pp. 28–39 (2020)
18. Karlin, A.R., Klein, N., Oveis Gharan, S.: A (slightly) improved approximation algorithm for metric TSP. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. p. 32–45. STOC 2021, Association for Computing Machinery (2021). https://doi.org/10.1145/3406325.3451009

19. Klein, N., Taziki, M.: Dual charging for half-integral tsp (2025), arXiv preprint arXiv:2507.17999
20. Mömke, T., Svensson, O.: Removing and Adding Edges for the Traveling Salesman Problem. J. ACM **63**(1), 2:1–2:28 (Feb 2016). https://doi.org/10.1145/2739008
21. Schalekamp, F., Williamson, D.P., van Zuylen, A.: 2-matchings, the traveling salesman problem, and the subtour LP: A proof of the Boyd-Carr conjecture. Mathematics of Operations Research **39**(2), 403–417 (2014)
22. Vercesi, E., Gualandi, S., Mastrolilli, M., Gambardella, L.M.: On the generation of metric TSP instances with a large integrality gap by branch-and-cut. Mathematical Programming Computation **15**(2), 389–416 (2023)
23. Wolsey, L.A.: Heuristic analysis, linear programming and branch and bound. In: Rayward-Smith, V.J. (ed.) Combinatorial Optimization II, pp. 121–134. Springer Berlin Heidelberg, Berlin, Heidelberg (1980). https://doi.org/10.1007/BFb0120913
24. Zhong, X.: Lower bounds on the integrality ratio of the subtour LP for the traveling salesman problem. Discrete Applied Mathematics **365**, 109–129 (2025)

A Auxiliary proofs

In this appendix, we discuss in detail all the proofs that have been removed from the main argument exposition. The self-contained proofs are presented consecutively, while a separate section is devoted to the more intricate arguments that warrant further discussion.

Proof (of Lemma 1). Consider three distinct 1-paths of $\boldsymbol{x}$ (they exist by Theorem 2) and let $\tilde{V}$ be the set of their end nodes, so that $|\tilde{V}| = 6$ and $\deg(v) \geq 3 \ \ \forall v \in \tilde{V}$. Then $|E_{\boldsymbol{x}}| = \frac{1}{2}\sum_{v\in V_n} \deg(v) = \frac{1}{2}\big(\sum_{v\in\tilde{V}} \deg(v) + \sum_{v\notin\tilde{V}} \deg(v)\big) \geq \frac{1}{2}\big(6{\cdot}3 + (n-6){\cdot}2\big) = n+3$. □

Proof (of Lemma 2). Theorem 1 gives the inequality $n + k = |E_{\boldsymbol{x}}| \leq 2n - 3$ which leads to $k + 3 \leq n$. The upper bound on n is derived from the fact that the minimum degree of the nodes of any ancestor $\boldsymbol{x}$ is 3, thus $n + k = |E_{\boldsymbol{x}}| = \frac{1}{2}\sum_{v\in V_n} \deg(v) \geq \frac{1}{2} \cdot 3\, n$ which simplifies to $n \leq 2k$. □

Proof (of Lemma 3). Let $\boldsymbol{x}_0 \in P^n_{\text{SEP}}$ be a vertex with a 1-edge ab. Let $\boldsymbol{x}_1 \in P^{n+1}_{\text{SEP}}$ be the result of the BB-move applied on ab: we denote w the node added and aw, wb the two 1-edges originated by this move. Consider a metric cost $\boldsymbol{c}^0$ which realizes the Gap^+ on $\boldsymbol{x}_0$, namely $\text{Gap}^+(\boldsymbol{x}_0) = \frac{\text{TSP}(\boldsymbol{c}^0)}{\boldsymbol{c}^0\boldsymbol{x}_0}$. We construct the metric $\boldsymbol{c}^1$ on V_{n+1} adding the node w at distance 0 form b: $c^1_{wb} = 0$, $c^1_{vw} = c^0_{vb}$ for all nodes $v \in V_n \smallsetminus b$ and $c^1_e = c^0_e$ for all edges $e \in E_n$. Clearly $\boldsymbol{c}^1$ is metric and $\boldsymbol{c}^1\boldsymbol{x}_1 = \boldsymbol{c}^0\boldsymbol{x}_0$.

We first show that $\text{TSP}(\boldsymbol{c}^0) \geq \text{TSP}(\boldsymbol{c}^1)$. Let $\boldsymbol{t}_0$ be a tour on K_n that is optimal for $\text{TSP}(\boldsymbol{c}^0)$: $\text{TSP}(\boldsymbol{c}^0) = \boldsymbol{c}^0\boldsymbol{t}_0$. Let $\boldsymbol{t}^+_0$ be the tour on K_{n+1} which retraces the steps of $\boldsymbol{t}_0$ but visits w immediately after b, that is, if the sequence of nodes visited by $\boldsymbol{t}_0$ is $b, v_2, \ldots, v_n$, then the sequence of nodes visited by $\boldsymbol{t}^+_0$ is $b, w, v_2, \ldots, v_n$. It is immediate to verify that $\boldsymbol{c}^1\boldsymbol{t}^+_0 = \boldsymbol{c}^0\boldsymbol{t}_0$: $\boldsymbol{t}^+_0$ uses the same edges of $\boldsymbol{t}_0$, except for aw and wb in place of ab, which still involve the same cost $c^1_{aw} + c^1_{wb} = c^0_{ab} + 0$. Therefore $\text{TSP}(\boldsymbol{c}^1) \leq \boldsymbol{c}^1\boldsymbol{t}^+_0 = \boldsymbol{c}^0\boldsymbol{t}_0 = \text{TSP}(\boldsymbol{c}^0)$.

On the other hand, $\mathrm{TSP}(\boldsymbol{c}^1)$ is always larger than $\mathrm{TSP}(\boldsymbol{c}^0)$: given an optimal tour $\boldsymbol{t}_1$ for $\mathrm{TSP}(\boldsymbol{c}^1)$, we can always cut out w to get a tour $\boldsymbol{t}_1^-$ with $\boldsymbol{c}^0\boldsymbol{t}_1^- \leq \boldsymbol{c}^1\boldsymbol{t}_1$ by triangle inequality, thus $\mathrm{TSP}(\boldsymbol{c}^0) \leq \boldsymbol{c}^0\boldsymbol{t}_1^- \leq \boldsymbol{c}^1\boldsymbol{t}_1 = \mathrm{TSP}(\boldsymbol{c}^1)$.

Putting both the inequalities together, we have $\mathrm{TSP}(\boldsymbol{c}^1) = \mathrm{TSP}(\boldsymbol{c}^0)$, which finally proves $\mathrm{Gap}^+(\boldsymbol{x}_1) \geq \frac{\mathrm{TSP}(\boldsymbol{c}^1)}{\boldsymbol{c}^1\boldsymbol{x}_1} = \frac{\mathrm{TSP}(\boldsymbol{c}^0)}{\boldsymbol{c}^0\boldsymbol{x}_0} = \mathrm{Gap}^+(\boldsymbol{x}_0)$. □

Proof (of Lemma 5). Let $\boldsymbol{x}'$ be a successor of $\boldsymbol{x}_0$. Let $e_1, \ldots, e_p$ be the 1-edges of $\boldsymbol{x}_0$ and let $d_1, \ldots, d_p$ and $d'_1, \ldots, d'_p$ be the lengths of the corresponding 1-paths of $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively. Consider the successor $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}_0$ whose 1-paths have $\max(d_1, d'_1), \ldots, \max(d_p, d'_p)$ internal nodes. Then $\tilde{\boldsymbol{x}}$ is a successor of $\boldsymbol{x}'$, hence $\mathrm{Gap}^+(\boldsymbol{x}') \leq \mathrm{Gap}^+(\tilde{\boldsymbol{x}})$ for Lemma 3, and $\tilde{\boldsymbol{x}}$ is a successor of $\boldsymbol{x}$ as well, hence $\mathrm{Gap}^+(\tilde{\boldsymbol{x}}) \leq \mathrm{GB}(\boldsymbol{x})$ for Theorem 4. This proves the lemma. □

A.1 Proof of Lemma 4

In Section 5.1, we introduced two different LPs: $\mathcal{D}\,\mathrm{OPT}^+$ (8) – (11) and $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$ (12) – (14); in Lemma 4, we claimed that they have the same objective value. This whole section is devoted to proving this equivalence.

We begin by giving a non-formal interpretation of the equivalence, as it may guide the understanding of the subsequent arguments. As it appears in the definition of the objective function (8), we shall think that the "important" variables of $\mathcal{D}\,\mathrm{OPT}^+$ are the $\boldsymbol{\mu}$, while the $\boldsymbol{\lambda}$ are just "auxiliary". An optimal dual assignment is an assignment of positive weights $\boldsymbol{\mu}$ to the tours, such that, for every edge ij, the sum of the weights of all the tours passing through ij gives exactly x_{ij} (because of complementary slackness). If we only considered the (positive) weights $\boldsymbol{\mu}$, we would obtain the undesired necessary condition that no tour shall use edges outside the support graph (where $x_{ij} = 0$); $\boldsymbol{\lambda}$ variables are meant to "correct" this restriction. In the constraints (9), the variable λ_{ijk} is subtracted from the edge ij and added to ik, jk, as if it is "erasing" the passage of a tour on ij and diverting it to pass through ik and jk instead. In the flavor of this interpretation, the new formulation $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$ has $\boldsymbol{\mu}$ variables for walks which stay on the support graph and may traverse the same edge multiple times; $\boldsymbol{\lambda}$ variables are no longer needed, as their corrections are already taken into account along walks.

In the subsequent proofs, we see how to pass from one formulation to the other, transforming the $\boldsymbol{\mu}$ variables of $\mathcal{D}\,\mathrm{OPT}^+(\boldsymbol{x})$, indexed by tours on K_n, into $\boldsymbol{\mu}$ variables of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$, indexed by walks on $G_{\boldsymbol{x}}$, and vice-versa. To convert a tour into a walk on $G_{\boldsymbol{x}}$, we substitute edges ij not in $G_{\boldsymbol{x}}$ with paths connecting i and j in $G_{\boldsymbol{x}}$[12]. To convert a walk into a tour, we consider the sequence of the nodes visited along the walk (having fixed an order and a starting point) and cut out repeated visits to the same nodes.

[12] This is always possible since $G_{\boldsymbol{x}}$ is connected, for subtour elimination constraints.

To accomplish the proof of Lemma 4, we introduce an intermediate formulation, $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}$. For a given vertex $\boldsymbol{x}$, $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ is defined as:

$$\text{maximize} \quad \sum_{\boldsymbol{w} \text{ walk}} \mu_{\boldsymbol{w}} \tag{15}$$

$$\text{subject to:} \quad \sum_{k \neq i,j} (-\lambda_{ijk} + \lambda_{ikj} + \lambda_{jki}) + \sum_{\boldsymbol{w} \text{ walk}} w_{ij}\mu_{\boldsymbol{w}} \leq x_{ij} \quad \forall ij \in E_n, \tag{16}$$

$$\lambda_{ijk} \geq 0 \qquad \forall ij \in E_n,\ k \neq i,j, \tag{17}$$

$$\mu_{\boldsymbol{w}} \geq 0 \qquad \forall \boldsymbol{w} \text{ walk on } K_n. \tag{18}$$

Since both tours on K_n and walks on $G_{\boldsymbol{x}}$ are special cases of walks on K_n, $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ is an extended version of both $\mathcal{D}\,\mathrm{OPT}^{+}$ and $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$.

Remark 2. Before proving the equivalence of $\mathcal{D}\,\mathrm{OPT}^{+}$, $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}$, and $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ (that is, they give the same optimal value), let us address an "exception" we may encounter along the proofs. In the following arguments, we often construct new walks by modifying pre-existing ones, adding or removing edges. In principle, it is not excluded that a new walk $\boldsymbol{w}'$ obtained this way passes through an edge ij more than twice. The problem arises because, as stated in Remark 1, we are not considering such walks in our formulations. Whenever this situation occurs, we shall then replace $\boldsymbol{w}'$ with another walk $\boldsymbol{w}''$ obtained removing two copies of ij; in addition, if a variable $\mu_{\boldsymbol{w}'}$ is assigned to $\boldsymbol{w}'$ in the $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}$ problem, we shall instead add the same amount to the variable $\mu_{\boldsymbol{w}''}$ and not consider $\mu_{\boldsymbol{w}'}$ at all. It is crucial to notice that this change preserves the same objective value as the "erroneous" construction. It preserves the feasibility of the assignment as well, since the value of the constraints (16) stays the same on all the edges but ij, on which it eventually decreases (since we dropped two copies of ij in $\boldsymbol{w}''$).

We are now ready to prove the two main lemmas of this section.

Lemma 6. *$\mathcal{D}\,OPT^{I}(\boldsymbol{x}) = \mathcal{D}\,OPT^{+}(\boldsymbol{x})$ for every vertex $\boldsymbol{x}$ of P^{n}_{SEP}.*

Proof. Since $\mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$ is a special case of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$, we immediately have $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}) \geq \mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$ and we only need to prove $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}) \leq \mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$. We prove it by showing that, for any given optimal solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$, we can compute a feasible solution of $\mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$ with the same objective value.

Let $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$ be an optimal solution for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$. Let $\boldsymbol{w}^0$ be a walk on K_n. If $\boldsymbol{w}^0$ is also a tour, then we can leave the variable $\mu_{\boldsymbol{w}^0}$ as it is; otherwise, if $\boldsymbol{w}^0$ is not a tour, in the aim of designing an assignment for $\mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$, we have to set $\mu_{\boldsymbol{w}^0}$ to 0. To do so without reducing the optimal value, we also need to adjust a couple of other variables. Let c be a node that is encountered more than once along $\boldsymbol{w}^0$ (it exists since $\boldsymbol{w}^0$ is not a tour), and let a and b be the nodes encountered immediately before and after c the second time it is traversed. Consider the walk $\boldsymbol{w}^1$ that retraces the steps of $\boldsymbol{w}^0$ but walks through the shortcut a, b instead of a, c, b, as illustrated in Figure 6. We design a new assignment of

variables $(\boldsymbol{\lambda}^1, \boldsymbol{\mu}^1)$, introducing, for simplicity, the constant $\delta^0 := \mu^0_{\boldsymbol{w}^0}$:

$$\begin{array}{ll} \lambda^1_{abc} := \lambda^0_{abc} + \delta^0, & \mu^1_{\boldsymbol{w}^0} := \mu^0_{\boldsymbol{w}^0} - \delta^0 = 0, \\ \lambda^1_{ijk} := \lambda^0_{ijk} \qquad \forall ijk \neq abc, & \mu^1_{\boldsymbol{w}^1} := \mu^0_{\boldsymbol{w}^1} + \delta^0, \\ & \mu^1_{\boldsymbol{w}} := \mu^0_{\boldsymbol{w}} \qquad\qquad \forall \boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1. \end{array}$$

It is immediate to check that the objective value does not change:

$$\begin{aligned} \sum_{\boldsymbol{w}} \mu^1_{\boldsymbol{w}} &= \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} \mu^1_{\boldsymbol{w}} \right) + \mu^1_{\boldsymbol{w}^0} + \mu^1_{\boldsymbol{w}^1} \\ &= \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} \mu^0_{\boldsymbol{w}} \right) + 0 + (\mu^0_{\boldsymbol{w}^1} + \mu^0_{\boldsymbol{w}^0}) \;=\; \sum_{\boldsymbol{w}} \mu^0_{\boldsymbol{w}} \;. \end{aligned}$$

We now show that $(\boldsymbol{\lambda}^1, \boldsymbol{\mu}^1)$ is feasible for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}$, that is, all the constraints (16) are satisfied. In fact, the values of the constraints attained by $(\boldsymbol{\lambda}^1, \boldsymbol{\mu}^1)$ is exactly the same as the ones attained by $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$. We check this fact for the constraint associated with the edge ab; the equalities holds simply by definition of $(\boldsymbol{\lambda}^1, \boldsymbol{\mu}^1)$ and considering that the multiplicities of the walk $\boldsymbol{w}^1$ are given as $w^1_{ab} = w^0_{ab}+1$, $w^1_{ac} = w^0_{ac}-1$, $w^1_{cb} = w^0_{cb}-1$.

$$\begin{aligned} &\sum_{k \neq a,b} (-\lambda^1_{abk} + \lambda^1_{akb} + \lambda^1_{bka}) \;+\; \sum_{\boldsymbol{w}} w_{ab} \mu^1_{\boldsymbol{w}} = \\ &= \left(\sum_{k \neq a,b,c} (-\lambda^1_{abk} + \lambda^1_{akb} + \lambda^1_{bka}) \right) + (-\lambda^1_{abc} + \lambda^1_{acb} + \lambda^1_{cba}) \\ &\quad + \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} w_{ab} \mu^1_{\boldsymbol{w}} \right) + w^0_{ab} \mu^1_{\boldsymbol{w}^0} + w^1_{ab} \mu^1_{\boldsymbol{w}^1} \\ &= \left(\sum_{k \neq a,b,c} (-\lambda^0_{abk} + \lambda^0_{akb} + \lambda^0_{bka}) \right) + (-(\lambda^0_{abc} + \delta^0) + \lambda^0_{acb} + \lambda^0_{cba}) \\ &\quad + \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} w_{ab} \mu^0_{\boldsymbol{w}} \right) + 0 + w^1_{ab} (\mu^0_{\boldsymbol{w}^1} + \delta^0) \\ &= \left(\sum_{k \neq a,b} (-\lambda^0_{abk} + \lambda^0_{akb} + \lambda^0_{bka}) \right) \;-\; \delta^0 \\ &\quad + \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} w_{ab} \mu^0_{\boldsymbol{w}} \right) + w^1_{ab} \mu^0_{\boldsymbol{w}^1} + (w^0_{ab}+1) \delta^0 \\ &= \left(\sum_{k \neq a,b} (-\lambda^0_{abk} + \lambda^0_{akb} + \lambda^0_{bka}) \right) \;-\; \delta^0 \\ &\quad + \left(\sum_{\boldsymbol{w} \neq \boldsymbol{w}^0, \boldsymbol{w}^1} w_{ab} \mu^0_{\boldsymbol{w}} \right) + w^1_{ab} \mu^0_{\boldsymbol{w}^1} + w^0_{ab} \mu^0_{\boldsymbol{w}^1} + \delta^0 \\ &= \sum_{k \neq a,b} (-\lambda^0_{abk} + \lambda^0_{akb} + \lambda^0_{bka}) \;+\; \sum_{\boldsymbol{w}} w_{ab} \mu^0_{\boldsymbol{w}} \end{aligned}$$

For ac, bc and all other edges ij, the computation is similar: contributes brought by the new $\boldsymbol{\lambda}^1$ and $\boldsymbol{\mu}^1$ (i.e. $\pm\delta^0$) cancel out and give the same value of the constraint computed for $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$, as illustrated in Figure 6.

We apply the same argument on $\boldsymbol{w}^1$ and go on iteratively, considering the walks $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^s$, until every node is encountered just once along the last walk. Eventually, $\boldsymbol{w}^s$ is a tour and all the variables $\mu_{\boldsymbol{w}^0}, \mu_{\boldsymbol{w}^1}, \ldots, \mu_{\boldsymbol{w}^{s-1}}$ are 0.

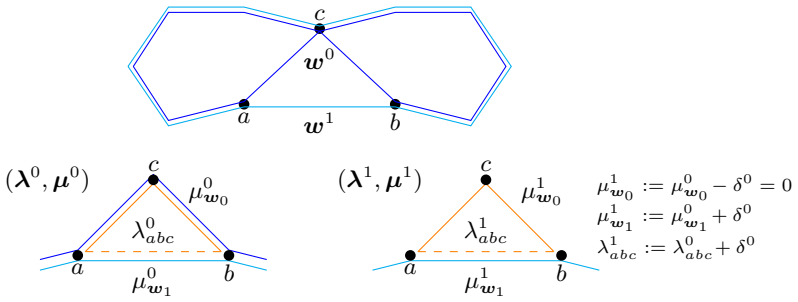

Fig. 6. Graphical representation of the proof of Lemma 6. Above: the walks $\boldsymbol{w}^0$ and $\boldsymbol{w}^1$: the latter is obtained from the former by cutting c out of the path acb. Below: a detail of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ variables, represented as triangles and walks with solid or dashed edges according to whether the variable is added or subtracted in the corresponding constraint. For both the assignments $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$ and $(\boldsymbol{\lambda}^1, \boldsymbol{\mu}^1)$, the constraint (16) attains the same value on every edge.

Applying this whole procedure for all the walks that are not tours yields a solution $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ which attains the optimal value, and with $\mu^*_{\boldsymbol{w}} = 0$ if $\boldsymbol{w}$ is not a tour; such a solution is in turn a feasible solution for $\mathcal{D}\,\mathrm{OPT}^{+}(\boldsymbol{x})$. This completes the proof. □

Lemma 7. *$\mathcal{D}\,OPT^{I}(\boldsymbol{x}) = \mathcal{D}\,OPT^{II}(\boldsymbol{x})$ for every vertex $\boldsymbol{x}$ of P^n_{SEP}.*

Proof. Since $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ is a special case of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$, we immediately have $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}) \geq \mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ and we only need to prove $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}) \leq \mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$. We prove it by showing that, for any given optimal solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$, we can compute a feasible solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ with the same objective value.

Let $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$ be an optimal solution for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$. Consider the polytope

$$P := \left\{ (\boldsymbol{\lambda}, \boldsymbol{\mu}) \;\middle|\; \begin{array}{l} (\boldsymbol{\lambda}, \boldsymbol{\mu}) \text{ optimal solution for } \mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}), \\ \sum_{ijk} \lambda_{ijk} \leq \sum_{ijk} \lambda^0_{ijk} \end{array} \right\}$$

$$= \left\{ (\boldsymbol{\lambda}, \boldsymbol{\mu}) \;\middle|\; \begin{array}{l} (\boldsymbol{\lambda}, \boldsymbol{\mu}) \text{ feasible solution for } \mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x}), \\ \sum_{\boldsymbol{w}} \mu_{\boldsymbol{w}} = \sum_{\boldsymbol{w}} \mu^0_{\boldsymbol{w}}, \\ \sum_{ijk} \lambda_{ijk} \leq \sum_{ijk} \lambda^0_{ijk} \end{array} \right\}.$$

P is a bounded polyhedron, hence it is compact. Consider the function $f : P \to \mathbb{R}$ which maps $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ into $f(\boldsymbol{\lambda}, \boldsymbol{\mu}) := \sum_{ijk} \lambda_{ijk}$. Since f is continuous on P compact, by the Weierstrass theorem, there exists at least one point $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in P$ where f attains its minimum value, i.e., $f(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \min_{(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in P} f(\boldsymbol{\lambda}, \boldsymbol{\mu})$. The following claim, whose proof is deferred at the end of the main argumentation, guarantees that the minimum of f is realized with $\boldsymbol{\lambda}^* = \mathbf{0}$.

Claim. For any $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ feasible solution for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ with $\boldsymbol{\lambda} \neq \mathbf{0}$, there exists another feasible solution $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$ for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ such that $\sum_{\boldsymbol{w}} \bar{\mu}_{\boldsymbol{w}} = \sum_{\boldsymbol{w}} \mu_{\boldsymbol{w}}$ and $\sum_{ijk} \bar{\lambda}_{ijk} < \sum_{ijk} \lambda_{ijk}$.

Moreover, for any edge ij not in $E_{\boldsymbol{x}}$, $x_{ij} = 0$ and the constraint (16) becomes $\sum_w w_{ij}\mu^*_{\boldsymbol{w}} \leq 0$, thus $\mu^*_{\boldsymbol{w}} = 0$ for every walk $\boldsymbol{w}$ with $w_{ij} > 0$. This ultimately means that $\lambda^*_{ijk} = 0$ for all ijk and $\mu^*_{\boldsymbol{w}} = 0$ for all the walks $\boldsymbol{w}$ which do not stay on the support graph $G_{\boldsymbol{x}}$; that is, the optimal solution $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$ is in turn a feasible solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ with the same objective value. □

Proof (of the claim). We divide the argument into two cases.

Case 1: exists an edge ab such that exist a node c with $\lambda_{abc} > 0$ and a walk $\boldsymbol{w'}$ with $w'_{ab}\mu_{\boldsymbol{w'}} > 0$ (the walk traverses ab at least once and the associated variable $\mu_{\boldsymbol{w}}$ is not zero). We aim to adjust the assignment by a small quantity δ that leaves λ_{abc} and $\mu_{\boldsymbol{w'}}$ grater or equal to 0: $\delta := \min(\lambda_{abc},\ \mu_{\boldsymbol{w'}})$. Let $\bar{\boldsymbol{w}}$ be the walk that retraces the steps of $\boldsymbol{w'}$ but drops once the edge ab and walks through a, c, b instead. We design a new assignment of variables:

$$\begin{aligned} \bar\lambda_{abc} &:= \lambda_{abc} - \delta\ , & & & \bar\mu_{\boldsymbol{w'}} &:= \mu_{\boldsymbol{w'}} - \delta\ , & \\ \bar\lambda_{ijk} &:= \lambda_{ijk} & \forall ijk \neq abc\ , & & \bar\mu_{\bar{\boldsymbol{w}}} &:= \mu_{\bar{\boldsymbol{w}}} + \delta\ , & \\ & & & & \bar\mu_{\boldsymbol{w}} &:= \mu_{\boldsymbol{w}} & \forall \boldsymbol{w} \neq \boldsymbol{w'}, \bar{\boldsymbol{w}}\ . \end{aligned}$$

It is immediate to check that the objective value stays the same and the sum of the $\boldsymbol{\lambda}$ variables strictly decreases:

$$\begin{aligned} \sum_{\boldsymbol{w}} \bar\mu_{\boldsymbol{w}} &= \sum_{\boldsymbol{w}\neq\boldsymbol{w'},\bar{\boldsymbol{w}}} \bar\mu_{\boldsymbol{w}}\ + \bar\mu_{\boldsymbol{w'}} + \bar\mu_{\bar{\boldsymbol{w}}} = \sum_{\boldsymbol{w}\neq\boldsymbol{w'},\bar{\boldsymbol{w}}} \mu_{\boldsymbol{w}}\ + \mu_{\boldsymbol{w'}} - \delta + \mu_{\bar{\boldsymbol{w}}} + \delta = \sum_{\boldsymbol{w}} \mu_{\boldsymbol{w}}\ , \\ \sum_{ijk} \bar\lambda_{ijk} &= \sum_{ijk\neq abc} \bar\lambda_{ijk}\ + \bar\lambda_{abc} = \sum_{ijk\neq abc} \lambda_{ijk}\ + \lambda_{abc} - \delta = \sum_{ijk} \lambda_{ijk} - \delta\ . \end{aligned}$$

Showing that the new assignment $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{w}})$ is feasible for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{I}}(\boldsymbol{x})$, as in the proof of Lemma 6, is just a matter of expand the new variables and check that the new weights and multiplicities balance out to give the same value as $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ for all the constraints.

Case 2: for every edges ij, all the variables λ_{ijk} are 0 or all the walks $\boldsymbol{w}$ have $w_{ij}\mu_{\boldsymbol{w}} = 0$. Then, for every edge ij, or $\sum_{\boldsymbol{w}} w_{ij}\mu_{\boldsymbol{w}} \leq \sum_{\boldsymbol{w}} w_{ij}\mu_{\boldsymbol{w}} + \sum_{k\neq i,j}(-0 + \lambda_{ikj} + \lambda_{jki}) = \sum_{\boldsymbol{w}} w_{ij}\mu_{\boldsymbol{w}} + \sum_{k\neq i,j}(-\lambda_{ijk} + \lambda_{ikj} + \lambda_{jki}) \leq x_{ij}$, or $\sum_{\boldsymbol{w}} w_{ij}\mu_{\boldsymbol{w}} = 0 \leq x_{ij}$. This proves that we can remove all the $\boldsymbol{\lambda}$ variables (i.e., set them to 0) to obtain the new assignment $(\boldsymbol{0}, \boldsymbol{\mu})$, which is feasible and preserves all the $\boldsymbol{\mu}$ variables, thus preserving the objective value.

Notice that this completes the proof since the two cases analyzed are the logical negation of each other. Writing the two clauses with formal predicate logic, it is immediate to check that $\neg$ (Case 1) $\Leftrightarrow$ (Case 2), as we have:

$$\begin{aligned} &\text{Case 1:} \quad \exists ab \in E_n \colon (\ \exists c \in V_n : \lambda_{abc} > 0\ \wedge\ \exists \boldsymbol{w'} \text{ walk on } K_n : w'_{ab}\mu_{\boldsymbol{w'}} > 0\ )\ , \\ &\text{Case 2:} \quad \forall ij \in E_n \colon (\ \forall k \in V_n : \lambda_{ijk} = 0\ \vee\ \forall \boldsymbol{w} \text{ walk on } K_n : w_{ij}\mu_{\boldsymbol{w}} = 0\ )\ . \ \square \end{aligned}$$

Finally, Lemma 4 trivially follows, concatenating Lemma 6 and Lemma 7.

A.2 Proof of the soundness of the GB algorithm

In Section 5.2, we introduced the GB algorithm and proved its soundness relying upon claims that we only stated. In this section, we provide proofs of all the technical details, ultimately completing the demonstration of Theorem 4.

Recall how we designed the GB algorithm. Starting from a vertex $\boldsymbol{x}_0$ alongside with an optimal solution $\boldsymbol{\mu}^0$ for the dual problem $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)$, we considered the successor $\boldsymbol{x}_d$ obtained expanding a 1-edge ab of $\boldsymbol{x}_0$ to a 1-path with d internal nodes and we constructed an assignment $\boldsymbol{\mu}^d$ for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_d)$. To do so, for every walk $\boldsymbol{w}^0$ on $G_{\boldsymbol{x}_0}$, depending on its multiplicity on the edge ab, we considered different walks on $G_{\boldsymbol{x}_d}$: for $\boldsymbol{w}^0 \in \mathcal{W}_0^{ab}$, we considered $d{+}1$ walks $\boldsymbol{w}_k^d$ on $G_{\boldsymbol{x}_d}$ and set $\mu^d_{\boldsymbol{w}_k^d} := \frac{1}{d+1}\mu^0_{\boldsymbol{w}^0}$ for all $k = 0, \dots, d$; for $\boldsymbol{w}^0 \in \mathcal{W}_1^{ab}$ or $\boldsymbol{w}^0 \in \mathcal{W}_2^{ab}$, we considered one walk $\boldsymbol{w}^d$ on $G_{\boldsymbol{x}_d}$ and set $\mu^d_{\boldsymbol{w}^d} := \mu^0_{\boldsymbol{w}^0}$ (see Section 5.2).

Claim. (i) $\boldsymbol{\mu}^d$ attains the same objective value (in $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_d)$) as $\boldsymbol{\mu}^0$ (in $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)$); (ii) $\boldsymbol{\mu}^d$ satisfies all the constraints (13), except those associated to the edges of the 1-path $aa_1 \dots a_d b$, which nonetheless remain bounded by a fixed constant.

Proof. The proof of the first fact (i) is the following straightforward computation:

$$\begin{aligned}
\sum_{\boldsymbol{w}^d \text{ walk on } G_{\boldsymbol{x}_d}} \mu^d_{\boldsymbol{w}^d} &= \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{ab}} \Big(\sum_{k=0}^{d} \mu^d_{\boldsymbol{w}_k^d} \Big) + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{ab}} \mu^d_{\boldsymbol{w}^d} + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{ab}} \mu^d_{\boldsymbol{w}^d} \\
&= \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{ab}} (d{+}1) \frac{1}{d+1} \mu^0_{\boldsymbol{w}^0} + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{ab}} \mu^0_{\boldsymbol{w}^0} + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{ab}} \mu^0_{\boldsymbol{w}^0} \\
&= \sum_{\boldsymbol{w}^0 \text{ walk on } G_{\boldsymbol{x}_0}} \mu^0_{\boldsymbol{w}^0} \ .
\end{aligned}$$

To prove the second fact (ii), we consider the edges of $E_{\boldsymbol{x}_d}$ and study the value of the associated constraints (13).

For edges ij of $E_{\boldsymbol{x}_d}$ not in the 1-path $aa_1 \dots b$, the multiplicity of the new walks $\boldsymbol{w}^d$ are the same of the ones they were originated from, i.e., $w^d_{ij} = w^0_{ij}$, and the constraint is still satisfied:

$$\begin{aligned}
\sum_{\boldsymbol{w}^d \text{ walk on } G_{\boldsymbol{x}_d}} w^d_{ij}\, \mu^d_{\boldsymbol{w}^d} &= \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{ab}} \Big(\sum_{k=0}^{d} (w^d_k)_{ij}\, \mu^d_{\boldsymbol{w}_k^d} \Big) + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{ab}} w^d_{ij}\, \mu^d_{\boldsymbol{w}^d} + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{ab}} w^d_{ij}\, \mu^d_{\boldsymbol{w}^d} \\
&= \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{ab}} \Big((d{+}1)\ (w^0_{ij}\ \frac{1}{d+1} \mu^0_{\boldsymbol{w}^0}) \Big) + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{ab}} w^0_{ij}\, \mu^0_{\boldsymbol{w}^0} + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{ab}} w^0_{ij}\, \mu^0_{\boldsymbol{w}^0} \\
&= \sum_{\boldsymbol{w}^0 \text{ walk on } G_{\boldsymbol{x}_0}} w^0_{ij}\, \mu^0_{\boldsymbol{w}^0} \ \le\ (x_0)_{ij} \ =\ (x_d)_{ij} \ .
\end{aligned}$$

It only remains to study the edges of the 1-path and bound from above the value of the constraints associated with them. Consider, for instance, aa_1

(analogous computations apply to the other edges):

$$\begin{aligned}\sum_{\boldsymbol{w}^d \text{ walk on } G_{\boldsymbol{x}_d}} w^d_{aa_1}\,\mu^d_{\boldsymbol{w}^d} &= \sum_{\boldsymbol{w}^0\in\mathcal{W}_0^{ab}}\Big(\sum_{k=0}^{d}(w^d_k)_{aa_1}\,\mu^d_{\boldsymbol{w}^d_k}\Big) + \sum_{\boldsymbol{w}^0\in\mathcal{W}_1^{ab}} w^d_{aa_1}\,\mu^d_{\boldsymbol{w}^d} + \sum_{\boldsymbol{w}^0\in\mathcal{W}_2^{ab}} w^d_{aa_1}\,\mu^d_{\boldsymbol{w}^d}\\ &= \sum_{\boldsymbol{w}^0\in\mathcal{W}_0^{ab}}\Big(0 + d\cdot 2\,\frac{1}{d+1}\mu^0_{\boldsymbol{w}^0}\Big) + \sum_{\boldsymbol{w}^0\in\mathcal{W}_1^{ab}} 1\,\mu^0_{\boldsymbol{w}^0} + \sum_{\boldsymbol{w}^0\in\mathcal{W}_2^{ab}} 2\,\mu^0_{\boldsymbol{w}^0}\\ &\le \;2\sum_{\boldsymbol{w}^0\in\mathcal{W}_0^{ab}}\mu^0_{\boldsymbol{w}^0} \;+\; \sum_{\boldsymbol{w}^0\in\mathcal{W}_1^{ab}}\mu^0_{\boldsymbol{w}^0} \;+\; 2\sum_{\boldsymbol{w}^0\in\mathcal{W}_2^{ab}}\mu^0_{\boldsymbol{w}^0}\;.\end{aligned}$$

The value of the constraint is bounded from above by a quantity that does not depend on d. □

We denote by $C(\boldsymbol{x}_0, ab)$ the right-hand side of the final inequality in the previous proof, which is the constant mentioned in the claim:

$$C(\boldsymbol{x}_0, ab) := 2\sum_{\boldsymbol{w}^0\in\mathcal{W}_0^{ab}}\mu^0_{\boldsymbol{w}^0} + \sum_{\boldsymbol{w}^0\in\mathcal{W}_1^{ab}}\mu^0_{\boldsymbol{w}^0} + 2\sum_{\boldsymbol{w}^0\in\mathcal{W}_2^{ab}}\mu^0_{\boldsymbol{w}^0}\;.$$

It may be used to rescale $\boldsymbol{\mu}^d$ and get the feasible assignment $\boldsymbol{\mu}^* := \frac{1}{C(\boldsymbol{x}_0,ab)}\boldsymbol{\mu}^d$ for $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_d)$. Crucially, $C(\boldsymbol{x}_0, ab)$ depends only on $\boldsymbol{x}_0$: the dependency on d was eliminated in the last inequality by bounding $\frac{d}{d+1}$ by 1. Therefore, the constant found is the same for all the vertices obtained by expanding the 1-edge ab of $\boldsymbol{x}_0$ to a 1-path of arbitrary length.

We now need to extend the construction for a generic successor of $\boldsymbol{x}_0$. To this end, we repeat the previous procedure for all the 1-edges $e_1, \ldots, e_p$ of $\boldsymbol{x}_0$, sequentially expanding them to 1-paths with $d_1, \ldots, d_p$ internal nodes. Let $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^p$ be the vertices obtained along this process. Accordingly, let $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^p$ be the assignments of dual variables iteratively originated (starting from $\boldsymbol{\mu}^0$).

Claim. (i) $\boldsymbol{\mu}^p$ attains the same objective value (in $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}^p)$) as $\boldsymbol{\mu}^0$ (in $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}_0)$); (ii) $\boldsymbol{\mu}^p$ satisfies all the constraints (13), except those associated to the 1-edges, which nonetheless remain bounded by fixed constants.

Proof. It follows directly from the iterative application of the prior claim. □

The constants used as bounds for constraints (13) are computed sequentially at each step $h = 1, \ldots, p$ of the construction, starting from the preceding vertex $\boldsymbol{x}^{h-1}$ (with $\boldsymbol{x}^0 := \boldsymbol{x}_0$):

$$C(\boldsymbol{x}^{h-1}, e_h) := 2\sum_{\boldsymbol{w}^{h-1}\in\mathcal{W}_0^{e_h}}\mu^{h-1}_{\boldsymbol{w}^{h-1}} + \sum_{\boldsymbol{w}^{h-1}\in\mathcal{W}_1^{e_h}}\mu^{h-1}_{\boldsymbol{w}^{h-1}} + 2\sum_{\boldsymbol{w}^{h-1}\in\mathcal{W}_2^{e_h}}\mu^{h-1}_{\boldsymbol{w}^{h-1}}\;.$$

To prove the soundness of the GB algorithm, it only remains to show that the factors $C(\boldsymbol{x}^{h-1}, e_h)$ are not determined by the order in which the 1-edges are expanded: they can be computed directly from the starting vertex $\boldsymbol{x}^0$.

Claim. $C(\boldsymbol{x}^{h-1}, e_h) = C(\boldsymbol{x}^0, e_h)$.

Proof. Before digging into the main argument of the proof, we introduce a notation that prevents ambiguities. For $m = 0, 1, 2$, we use $\mathrm{E}_m^{e_h}(\boldsymbol{w}^{h-1})$ (eventually with subscript k when $m = 0$) to denote the new walks obtained when extending $\boldsymbol{w}^{h-1}$ ("E" for "extend") to pass through all the new nodes added in the expansion of e_h to a 1-path with d_h internal nodes (see Section 5.2). With this new notation, the new assignment of $\boldsymbol{\mu}$ variables becomes $\mu^h_{\mathrm{E}_0^{e_h}(\boldsymbol{w}^{h-1})_k} := \frac{1}{d_h+1}\mu^{h-1}_{\boldsymbol{w}^{h-1}}$ for all $k = 0, \ldots, d_h$, $\mu^h_{\mathrm{E}_1^{e_h}(\boldsymbol{w}^{h-1})} := \mu^{h-1}_{\boldsymbol{w}^{h-1}}$, or $\mu^h_{\mathrm{E}_2^{e_h}(\boldsymbol{w}^{h-1})} := \mu^{h-1}_{\boldsymbol{w}^{h-1}}$, according to whether $\boldsymbol{w}^{h-1} \in \mathcal{W}_0^{e_h}$, $\boldsymbol{w}^{h-1} \in \mathcal{W}_1^{e_h}$, or $\boldsymbol{w}^{h-1} \in \mathcal{W}_2^{e_h}$, respectively.

We also introduce an operator that mimics an *if* statement: we use $\langle \mathit{condition} \rangle$, which has value 1 if *condition* is true and 0 otherwise. $C(\boldsymbol{x}^{h-1}, e_h)$ can thus be rewritten compactly as

$$C(\boldsymbol{x}^{h-1}, e_h) = \sum_{\substack{\boldsymbol{w}^{h-1} \\ \text{walk on } G_{\boldsymbol{x}^{h-1}}}} \sum_{m=0,1,2} \theta_m \, \langle \boldsymbol{w}^{h-1} \in \mathcal{W}_m^{e_h} \rangle \, \mu^{h-1}_{\boldsymbol{w}^{h-1}} \, ,$$

where $\theta_0 := 2$, $\theta_1 := 1$, $\theta_2 := 2$.

To prove the claim, we adopt a recursive strategy: we show that the constant $C(\boldsymbol{x}^{h-1}, e_h)$ can be computed by swapping steps $h-1$ and h in the construction, namely $C(\boldsymbol{x}^{h-1}, e_h) = C(\boldsymbol{x}^{h-2}, e_h)$. Applying the same argument until the step 0 is reached proves the desired statement. For the sake of simplifying the indexing, we only consider the second step $h = 2$; the same argument is easily generalizable for the generic step $h \in \{2, \ldots, p\}$.

We start by unfolding the definition of $C(\boldsymbol{x}^1, e_2)$, considering how the walks $\boldsymbol{w}^1$ on $G_{\boldsymbol{x}^1}$ are originated from the walks $\boldsymbol{w}^0$ on $G_{\boldsymbol{x}^0}$:

$$\begin{aligned}
C(\boldsymbol{x}^1, e_2) &= \sum_{\substack{\boldsymbol{w}^1 \\ \text{walk on } G_{\boldsymbol{x}^1}}} \sum_{m=0,1,2} \theta_m \langle \, \boldsymbol{w}^1 \in \mathcal{W}_m^{e_2} \rangle \, \mu^1_{\boldsymbol{w}^1} \\
&= \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{e_1}} \left(\sum_{k=0}^{d_1} \Big(\sum_{m=0,1,2} \theta_m \, \langle \mathrm{E}_0^{e_1}(\boldsymbol{w}^0)_k \in \mathcal{W}_m^{e_2} \rangle \, \mu^1_{\mathrm{E}_0^{e_1}(\boldsymbol{w}^0)_k} \Big) \right) \\
&\quad + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{e_1}} \left(\sum_{m=0,1,2} \theta_m \, \langle \mathrm{E}_1^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \rangle \, \mu^1_{\mathrm{E}_1^{e_1}(\boldsymbol{w}^0)} \right) \\
&\quad + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{e_1}} \left(\sum_{m=0,1,2} \theta_m \, \langle \mathrm{E}_2^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \rangle \, \mu^1_{\mathrm{E}_2^{e_1}(\boldsymbol{w}^0)} \right) .
\end{aligned}$$

Notice that, when extending a walk on the 1-edge e_1, its multiplicity on e_2 is not affected. Therefore, for all $m = 0, 1, 2$, we have the equivalent conditions:

$$\begin{aligned}
&\boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \;\Leftrightarrow\; \mathrm{E}_0^{e_1}(\boldsymbol{w}^0)_k \in \mathcal{W}_m^{e_2} \;\Leftrightarrow\; \mathrm{E}_1^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \;\Leftrightarrow\; \mathrm{E}_2^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \, , \\
&\langle \boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \rangle = \langle \mathrm{E}_0^{e_1}(\boldsymbol{w}^0)_k \in \mathcal{W}_m^{e_2} \rangle = \langle \mathrm{E}_1^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \rangle = \langle \mathrm{E}_2^{e_1}(\boldsymbol{w}^0) \in \mathcal{W}_m^{e_2} \rangle \, .
\end{aligned}$$

With this observation, the constant $C(\boldsymbol{x}^1, e_2)$ may finally be rewritten as $C(\boldsymbol{x}^0, e_2)$, thus completing the proof:

$$\begin{aligned} C(\boldsymbol{x}^1, e_2) = & \sum_{\boldsymbol{w}^0 \in \mathcal{W}_0^{e_1}} \Big(\sum_{k=0}^{d_1} \Big(\sum_{m=0,1,2} \theta_m \, \langle \boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \rangle \, \frac{1}{d_1+1} \mu^0_{\boldsymbol{w}^0} \Big) \Big) \\ & + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_1^{e_1}} \Big(\sum_{m=0,1,2} \theta_m \, \langle \boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \rangle \, \mu^0_{\boldsymbol{w}^0} \Big) \\ & + \sum_{\boldsymbol{w}^0 \in \mathcal{W}_2^{e_1}} \Big(\sum_{m=0,1,2} \theta_m \, \langle \boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \rangle \, \mu^0_{\boldsymbol{w}^0} \Big) \\ = & \sum_{\substack{\boldsymbol{w}^0 \\ \text{walk on } G_{\boldsymbol{x}^0}}} \sum_{m=0,1,2} \theta_m \langle \, \boldsymbol{w}^0 \in \mathcal{W}_m^{e_2} \rangle \, \mu^0_{\boldsymbol{w}^0} \\ = & \, C(\boldsymbol{x}^0, e_2) \; . \end{aligned}$$

□

Finally, at the very end of the procedure, the last dual variables $\boldsymbol{\mu}^p$ are adjusted by the global factor

$$C^*(\boldsymbol{x}^0) := \max_{h=1,\dots,p} \{C(\boldsymbol{x}^0, e_h)\} = \max_{h=1,\dots,p} \{C(\boldsymbol{x}^{h-1}, e_h)\} \; .$$

The claims of this section guarantee that the rescaled assignment $\boldsymbol{\mu}^* := \frac{1}{C^*(\boldsymbol{x}^0)} \boldsymbol{\mu}^p$ is feasible and attains the objective value $\frac{1}{C^*(\boldsymbol{x}^0)} \mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}^0)$, therefore yielding, by eq. (6), eq. (7), and Lemma 4, the inequality:

$$\mathrm{Gap}^+(\boldsymbol{x}') \leq C^*(\boldsymbol{x}^0) \cdot \mathrm{Gap}^+(\boldsymbol{x}^0) \; .$$

This final inequality is the one used in the proof of Theorem 4, which is now complete in all its details.

B Implementation details for the GB algorithm

This section enriches the content of Section 6, discussing implementation details of the GB algorithm.

B.1 Details on the computation of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$

First, we briefly delve into the procedure of solving $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ (12) – (14), used as a subroutine of Algorithm 1.

It is not computationally efficient to solve the dual problem $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$ with all variables included from the outset. Instead, we adopt a well-known *row generation* approach to the primal: constraints are added incrementally, one at a time, until it can be proven that all remaining constraints are implied by those already included. This technique is analogous to the separation of subtour elimination constraints in the standard approach to solving the TSP (See, e.g.,

[7]). The LP to solve becomes $\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$, the primal of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$: variables (respectively constraints) of the former correspond to constraints (respectively variables) of the latter.

$$\begin{aligned}
\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}) := \ \text{minimize} \quad & \sum_{ij \in E_{\boldsymbol{x}}} x_{ij} c_{ij} & \\
\text{subject to:} \quad & \sum_{ij \in E_{\boldsymbol{x}}} w_{ij} c_{ij} \ \geq 1 & \forall \boldsymbol{w} \text{ walk on } G_{\boldsymbol{x}}, \\
& c_{ij} \geq 0 & \forall ij \in E_{\boldsymbol{x}}.
\end{aligned}$$

The constraints associated with the walks on $G_{\boldsymbol{x}}$ are added to the model sequentially, choosing the "most restrictive", i.e., the ones corresponding to the walks of minimal cost. These may be found by solving the following ILP, which is the version of the graph-TSP with costs on the edges.

$$\begin{aligned}
\text{minimize} \quad & \sum_{ij \in E_{\boldsymbol{x}}} c_{ij} w_{ij} & \\
\text{subject to:} \quad & \sum_{ij \in \delta(v)} w_{ij} \ = 2\ d_v & \forall v \in V_n, \\
& \sum_{ij \in \delta(S)} w_{ij} \ \geq 2 & \forall S \in \mathcal{S}, \\
& 0 \leq \ w_{ij} \ \leq 2 & \forall ij \in E_{\boldsymbol{x}}, \\
& w_{ij} \ \text{ integer} & \forall ij \in E_{\boldsymbol{x}}, \\
& d_v \ \text{ integer} & \forall v \in V_n.
\end{aligned}$$

Once the solution is proven to be optimal, we formulate $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}$ using only those variables $\mu_{\boldsymbol{w}}$ corresponding to tight constraints of $\mathrm{OPT}^{\mathrm{II}}$. In virtue of the theorem of complementary slackness (see, e.g., [10]), all remaining dual variables can be safely set to zero. At this point, the objective values of the primal and dual formulations coincide, and we recover the non-zero $\mu_{\boldsymbol{w}}$ values required for the estimates $C(\boldsymbol{x}, e)$.

B.2 Details on the obtained upper bounds

To prove Theorem 5, we executed the GB algorithm on all the ancestors of $\mathcal{A}_k$ with $k = 3, 4, 5, 6$. As anticipated in Section 6, these initial runs have not always returned the desired 4/3 bound. Consequently, to accomplish our objective, the GB algorithm was also applied to certain successors of these "unsatisfying" ancestors, in accordance with Lemma 5. Since there is some flexibility in choosing the next successor to explore, we adopted the following heuristic: given a vertex $\boldsymbol{x}$ with $\mathrm{GB}(\boldsymbol{x}) > 4/3$, we considered the successor $\boldsymbol{x}' = \textsc{bb}(\boldsymbol{x}, e)$ obtained by the application of a BB-move on the 1-edge e of $\boldsymbol{x}$ with the largest constants $C(\boldsymbol{x}, e)$ (as computed in Algorithm 1, line 4).

Notice that, when repeatedly applying the GB algorithm on subsequent successors, the procedure of solving $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x}')$ (subroutine of Algorithm 1, line 7) may benefit from the previous solution of $\mathcal{D}\,\mathrm{OPT}^{\mathrm{II}}(\boldsymbol{x})$, already at our disposal: instead of restarting the row generation technique from zero, we may take the optimal walks of $\boldsymbol{x}$, transform them into walks on $\boldsymbol{x}'$ as described at the beginning of Section 5.2, and initialize the model with the associated constraints.

Table 1 reports, for every $k = 3, 4, 5, 6$, the number of ancestors in $\mathcal{A}_k$ (up to isomorphism), the upper bound found on the Gap for the vertices in $\mathcal{F}_k$ (namely, the maximum of all the values returned by the GB algorithm applied on the ancestors in $\mathcal{A}_k$ and the selected successors), and the maximum number of additional consecutive iterations of the GB algorithm needed on the ancestor to achieve the final bound.

Table 1. Computational results of the application of the Gap-Bounding algorithm.

k	$\lvert\mathcal{A}_k\rvert$	upper bound on Gap for $\mathcal{F}_k$	max additional iterations of GB algorithm
3	1	4/3	0
4	5	4/3	2
5	44	4/3	5
6	715	4/3	10

We remark that the values given in the table are upper bounds: in the perspective of proving the conjecture, we were satisfied with 4/3, but in principle the actual Gap on a certain family may be even lower.