Exact and approximate formulations for the close-enough TSP

Domingo Araya[1], Gustavo Angulo[1†], Margarita Castro[1*†]

[1*]Department of Industrial and Systems Engineering, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Santiago, 7820436, Chile.

*Corresponding author(s). E-mail(s): margarita.castro@uc.cl;
Contributing authors: domingo.araya@uc.cl; gangulo@uc.cl;
†These authors contributed equally to this work.

Abstract

This work addresses the Close-Enough Traveling Salesman Problem (CETSP), a variant of the classic traveling salesman problem in which we seek to visit neighborhoods of points in the plane (defined as disks) rather than specific points. We present three exact formulations based on mixed-integer second-order cone programming (MISOCP): an arc-based formulation relying on big-M constraints, a perspective-based formulation that avoids them, and a sequence-based formulation. We also propose approximated mixed-integer linear formulations derived from polyhedral approximations of the second-order cone constraints. While MISOCP models yield optimal solutions, they are quite inefficient for larger instances. Thus, we propose decomposition schemes based on exact MISOCP models and their corresponding MILP approximations, along with several cuts derived from SOCP duality and classic integer L-shaped methods. Moreover, we show that the generalized Benders cuts previously proposed in the literature for a closely related problem are not valid, and illustrate the consequences of using them. We test all the proposed procedures over a wide range of instances and analyze the results for different instance features (i.e., number of points and radius size), and compare our best-performing procedure with an existing discretization-based method from the literature. We note that the decompositions based on the perspective formulation achieve the best overall performance, with dual cuts being most effective when the master problem provides weak distance estimates and enumerative cuts when it relies on the MILP approximation, illustrating the benefit of combining methodologies from the MILP and MISOCP literature.

Keywords: Traveling Salesman Problem, Mixed-Integer Second-Order Cone Programming, Polyhedral Approximation, Benders Decomposition

1 Introduction

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem that has been the subject of extensive study for several decades. Given a set of locations and pairwise distances, the goal is to find the shortest possible tour that starts and ends at a designated location while visiting every other location exactly once. The TSP has several applications in a variety of fields, such as logistics, genome sequencing, and data clustering [1], to name a few.

This work addresses the Close-Enough Traveling Salesman Problem (CETSP), a generalization of the TSP in which it is sufficient to visit a previously defined compact region, known as its neighborhood, that contains the exact location. Although the problem can be defined for arbitrary compact regions, we assume that the neighborhoods are closed disks centered at the location. This is a common assumption in the CETSP literature [2, 3], and corresponds to the type of neighborhoods that arise naturally in many applications, such as, distant meter readings via radio frequency identification technology [4], UAV routing in various contexts [3], and robot monitoring of wireless sensor networks [5].

Several generalizations of the CETSP have been proposed to capture additional practical requirements. For instance, Di Placido et al. [6] propose the generalized CETSP, a variation of the problem where each location has several neighborhoods instead of a single one, with neighborhoods given by disks of different radii, and a prize associated with each region which is redeemed by traversing it. Carrabs et al. [7] propose the multiple CETSP, in which there are several vehicles, and the objective is to minimize the length of the longest tour assigned to a vehicle. Cui et al. [8] introduce the CETSP with service time, in which the traveler must spend a certain amount of time on each neighborhood. Gao et al. [9] study a variant of the CETSP called the pickup-and-delivery TSP with neighborhoods (PDTSPN), which combines traditional pickup-and-delivery requirements with the flexibility of the CETSP. In this context, they propose a cut-generation scheme to solve the problem, based on generalized Benders decomposition [10]. An erratum to this work was later made available [11], correcting the proposed decomposition.

While there are several heuristic approaches to solving the CETSP [4, 12–15], there is a notable scarcity of exact algorithms with optimality guarantees. Mennell [12] introduced the first mathematical formulation of the problem, along with a heuristic approach based on finding non-empty neighborhood intersections, to subsequently solve a TSP over a selection of points from said intersections. Additionally, they introduced the *overlap ratio* as a measure of instance difficulty, defined as the ratio of the mean radius to the largest side of the smallest rectangle containing all neighborhoods. Gentilini et al. [16] presented a solution approach for this formulation based

on a traditional spatial branch-and-bound algorithm for mixed-integer non-linear programming (MINLP), with specific improvements for the CETSP to make the problem more tractable for MINLP solvers. Behdani and Smith [2] proposed a discretization scheme that generates lower and upper bounds on most test instances, and can be iteratively refined to make said bounds arbitrarily close. Building on these results, Carrabs et al. [17] introduced a novel discretization scheme that improves upon those of [2], generally providing better bounds in less time.

To the best of our knowledge, the only exact algorithm specifically designed for this problem is a combinatorial *branch-and-bound* [3], where each node of the tree corresponds to a partial visit sequence, for which the visit points for each neighborhood are computed via second-order cone programming (SOCP). If the solution does not visit every neighborhood, the partial sequence is extended, creating new nodes of the branching tree. This approach provides good results on several benchmark instances but, in the worst case, has to consider every possible partial sequence. This algorithm was refined by Zhang et al. [18], who proposed a number of improvements that lead to better results in most instances of the problem. Gutow and Choset [19] further improved the algorithm by reusing information from parent nodes in the branching tree to speed up the solution of subproblems corresponding to the children nodes, while Deckerová et al. [20] and Fekete et al. [21] extended it to generalizations of the CETSP.

In this paper, we present three mixed-integer second-order cone programming (MISOCP) formulations for the CETSP, namely an arc-based formulation relying on big-M constraints, a perspective-based formulation that avoids them, and a novel sequence-based formulation, and compare the strength of their continuous relaxations. We also present approximated formulations based on mixed-integer linear programming (MILP) that provide arbitrarily close approximations to the feasible regions of the original formulations. Furthermore, we propose decomposition schemes for all MISOCP and MILP formulations to make the problem more computationally manageable and, for the MILP models, to close the gap left by the approximation factor, using both enumerative cuts and cuts derived from conic duality. We also provide examples that demonstrate that the cuts originally proposed by [9] are not always valid for the problem. Finally, we run extensive computational experiments on a variety of instances to measure and compare the performance of the formulations mentioned above, including a comparison with the discretization scheme of Behdani and Smith [2]. Our results show that the decompositions based on the perspective formulation achieve the best overall performance, with dual cuts being most effective when the master problem provides weak distance estimates, and enumerative cuts when it relies on the MILP approximation.

The remainder of the paper is organized as follows. Section 2 contains a formal definition of the CETSP, along with the notation that is used throughout the paper for the parameters of the problem. Section 3 presents the MISOCP formulations and compares their continuous relaxations, while Section 4 introduces the linear approximation scheme and presents the resulting formulations. Section 5 describes the decomposition schemes and the cuts used in each of them. In Section 6, we show that the cuts originally proposed by [9] based on generalized Benders decomposition are not valid.

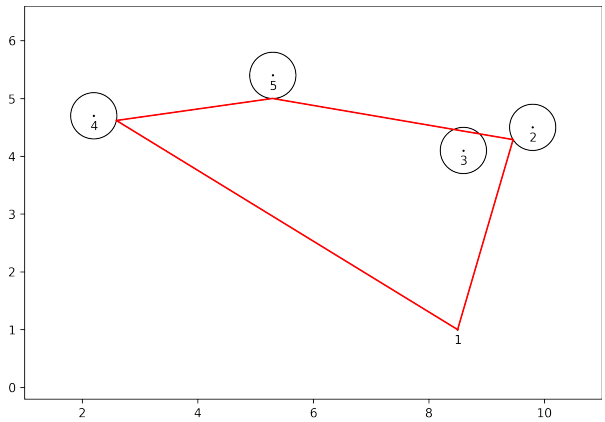


Fig. 1: An optimal CETSP tour with $|N| = 5$.

Lastly, Section 7 presents computational results, and Section 8 concludes with some final remarks.

2 Problem description

The CETSP can be formally stated as follows. Let $N = \{1, \ldots, n\}$ be a set of locations in a two-dimensional plane, with their corresponding coordinates $c_i = (c_i^x, c_i^y)$ for each $i \in N$. Each location $i \in N$ is covered by a closed disk centered in c_i with radius r_i, which we call its neighborhood. In this setting, the CETSP seeks the shortest tour that visits each neighborhood, with no restrictions on the order of visits. We consider a neighborhood to be visited if the intersection of the tour and the neighborhood is nonempty. Although the problem can be stated in terms of arbitrary norms, we consider Euclidean distances. We also assume $r_1 = 0$, so location 1 acts as the depot.

Behdani and Smith [2] proved that any optimal solution can be characterized by line segments between the neighborhoods. Hence, the problem reduces to finding a visit sequence along with the endpoints of the corresponding line segments, such that the tour over these points forms a Hamiltonian cycle of minimum length and each point is contained in its corresponding neighborhood. Figure 1 illustrates an optimal CETSP tour for an instance with 5 locations, exemplifying this property.

3 MISOCP formulations

We now present three MISOCP formulations for the CETSP. The first two models are derived by convexifying a natural formulation for the problem obtained by adapting the classic TSP formulation; the first uses a big-M convexification, while the second consists of a perspective reformulation. The third model focuses on the problem's sequential aspect and is, to the best of our knowledge, a novel formulation.

For notation purposes, we consider $A = \{(i, j) \in N \times N : i \neq j\}$ as the set of all ordered pairs of distinct locations and, similar to the classic TSP, we refer to each $(i, j) \in A$ as an arc. A natural way to formulate the CETSP, originally proposed by

Mennell [12], is to consider binary variables x_{ij} that indicate transitions between neighborhoods, and variables $p_i \in \mathbb{R}^2$ representing the coordinates of the representative point chosen for each neighborhood. Then, the problem is:

$$\min \sum_{(i,j)\in A} x_{ij}\|p_i - p_j\| \tag{1a}$$
$$\text{s.t. } x \in \mathcal{X}, \tag{1b}$$
$$\|p_i - c_i\| \le r_i, \qquad \forall i \in N, \tag{1c}$$
$$p_i \in \mathbb{R}^2, \qquad \forall i \in N, \tag{1d}$$

where $\mathcal{X}$ is the set of Hamiltonian cycle incidence vectors, i.e., $x \in \mathcal{X}$ satisfies degree constraints plus any subtour elimination constraint family (e.g., DFJ or MTZ formulations [22, 23]). The objective function (1a) minimizes the sum of the distances between points on the transitions used, while constraints (1b) ensure that x defines a tour and (1c) that each point belongs to its corresponding neighborhood. Note that setting $r_i = 0$ for all $i \in N$ recovers the Euclidean TSP.

Observe that (1a) as it stands is not jointly convex when relaxing integrality. Each term in the objective encodes a disjunction: either arc (i, j) is unused and contributes nothing, or it is used and contributes $\|p_i - p_j\|$. We present two convex alternatives for modeling this disjunction, the first of which relies on big-M constraints. Let M_{ij} be the maximum possible distance between neighborhoods i and j (i.e., $M_{ij} = r_i + \|c_i - c_j\| + r_j$) and consider new variables d_{ij} representing the distances associated with the realized transitions between neighborhoods. Then, the problem can be restated via the following formulation, which we call *arc-based formulation* (ABF):

$$\min \sum_{(i,j)\in A} d_{ij} \tag{ABF}$$
$$\text{s.t. } x \in \mathcal{X}, \tag{2a}$$
$$\|p_i - c_i\| \le r_i, \qquad \forall i \in N, \tag{2b}$$
$$\|p_i - p_j\| \le d_{ij} + M_{ij}(1 - x_{ij}), \qquad \forall (i,j) \in A, \tag{2c}$$
$$d_{ij} \ge 0, \qquad \forall (i,j) \in A, \tag{2d}$$
$$p_i \in \mathbb{R}^2, \qquad \forall i \in N. \tag{2e}$$

Constraints (2c) ensure that d_{ij} equals $x_{ij}\|p_i - p_j\|$ at an optimal solution; when x_{ij} is 0, the constraint is trivially satisfied and thus the distance variable takes value 0 (as the objective minimizes over d), while, when x_{ij} is 1, the constraint reduces to $\|p_i - p_j\| \le d_{ij}$.

A potential drawback of ABF is that it relies on big-M constraints, which are known to have poor relaxations and to lead to numerical issues [24]. Alternatively, we present a perspective reformulation of the nonconvex model, an approach known to provide tight relaxations for disjunctions of this type [25]. This approach has been successfully applied to related problems, such as the shortest path problem on graphs

of convex sets [26] and other combinatorial problems on such graphs, including the TSP [27]; the formulation below adapts this framework to the CETSP.

For each arc $(i,j) \in A$, we introduce variables p_i^{ij} and $p_j^{ij} \in \mathbb{R}^2$, intended to satisfy $p_i^{ij} = x_{ij} p_i$ and $p_j^{ij} = x_{ij} p_j$, thus acting as copies of the original point variables. Then, the objective term of each arc becomes $x_{ij}\|p_i - p_j\| = \|x_{ij}p_i - x_{ij}p_j\| = \|p_i^{ij} - p_j^{ij}\|$. Similarly, multiplying neighborhood constraints by x_{ij} yields $\|p_i^{ij} - x_{ij}c_i\| \leq x_{ij} r_i$, and an analogous constraint for the j entry. Note that these constraints force $p_i^{ij} = p_j^{ij} = 0$ when $x_{ij} = 0$, and reduce to the original constraints when $x_{ij} = 1$.

To handle the nonconvex definitions $p_i^{ij} = x_{ij} p_i$, we sum the new variables over the outgoing and incoming arcs of each location $i \in N$. Then, using the degree constraints over the x_{ij} variables, we get the linear equalities

$$p_i = \sum_{j:(i,j)\in A} p_i^{ij} = \sum_{k:(k,i)\in A} p_i^{ki},$$

which we use in place of the definitions. Since the previous constraints force the point copies to vanish on unused arcs, each sum reduces to the corresponding point on the single used arc, thus recovering $p_i^{ij} = x_{ij} p_i$ at integer solutions.

Finally, by introducing distance variables d_{ij} and eliminating the p_i variables, we obtain the following formulation, which we call *perspective-based formulation* (PBF):

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} d_{ij} && && \text{(PBF)}\\
\text{s.t. } & x \in \mathcal{X}, && && \text{(3a)}\\
& \|p_i^{ij} - x_{ij} c_i\| \leq x_{ij} r_i, && \forall (i,j) \in A, && \text{(3b)}\\
& \|p_j^{ij} - x_{ij} c_j\| \leq x_{ij} r_j, && \forall (i,j) \in A, && \text{(3c)}\\
& \|p_i^{ij} - p_j^{ij}\| \leq d_{ij}, && \forall (i,j) \in A, && \text{(3d)}\\
& \sum_{j:(i,j)\in A} p_i^{ij} = \sum_{k:(k,i)\in A} p_i^{ki}, && \forall i \in N, && \text{(3e)}\\
& d_{ij} \geq 0, && \forall (i,j) \in A, && \text{(3f)}\\
& p_i^{ij}, p_j^{ij} \in \mathbb{R}^2, && \forall (i,j) \in A. && \text{(3g)}
\end{aligned}$$

Constraints (3b) and (3c) ensure that point copies of used arcs are located in their corresponding neighborhoods, and that point variables vanish for unused transitions, while (3d) defines the distances between visited points and (3e) ensures that the point associated with a neighborhood on its incoming arc equals the one of its outgoing arc.

Note that both ABF and PBF contain $\mathcal{O}(n^2)$ second-order conic constraints, which may limit their scalability as n grows. This issue motivates our third formulation, the *sequence-based formulation* (SBF), which determines which neighborhood is visited at each stage of the tour. In this formulation, variables $z_{ik} \in \{0,1\}$ indicate whether location $i \in N$ is visited in stage $k \in N$, $q_k \in \mathbb{R}^2$ represents the coordinates of the point visited in stage k, and $t_k \geq 0$ indicates the distance traveled in stage k. The

formulation is as follows:

$$\min \sum_{k\in N} t_k \tag{SBF}$$

$$\text{s.t.} \sum_{k\in N} z_{ik} = 1, \qquad \forall i \in N, \tag{4a}$$

$$\sum_{i\in N} z_{ik} = 1, \qquad \forall k \in N, \tag{4b}$$

$$z_{11} = 1, \tag{4c}$$

$$\|q_k - q_{k+1}\| \leq t_k, \qquad \forall k \in N, \tag{4d}$$

$$\left\| \sum_{i\in N} z_{ik} c_i - q_k \right\| \leq \sum_{i\in N} z_{ik} r_i, \qquad \forall k \in N, \tag{4e}$$

$$z_{ik} \in \{0,1\}, \qquad \forall i,k \in N, \tag{4f}$$

$$q_k \in \mathbb{R}^2,\ t_k \geq 0, \qquad \forall k \in N. \tag{4g}$$

The objective function of SBF minimizes the total distance traveled, constraint (4a) ensures that each neighborhood is visited at some stage, while constraint (4b) makes it so that one neighborhood is visited at each stage. Constraint (4c) fixes the first neighborhood to be visited (i.e., the depot), to eliminate symmetrical solutions given by shifting the sequence. Constraint (4d) defines the distance between representative points (taking $n+1$ to be equal to 1), and constraint (4e) ensures that the representative point of stage k is contained in the neighborhood visited in that stage. Finally, constraints (4f)–(4g) state the nature of the variables.

3.1 Relationship among MISOCP formulations

We now compare the strength of the continuous relaxations (i.e., relaxing variables' integrality) of the proposed MISOCP formulations. All proofs for the following propositions can be found in Appendix A. Our first result, formalized in Proposition 1, shows that the relaxation of PBF is at least as strong as that of ABF. Moreover, Proposition 2 shows, by means of an example, that this dominance can be strict.

Proposition 1. *Suppose ABF and PBF use the same constraint set $\mathcal{X}$. Then the projection of the continuous relaxation of PBF onto the (x,d)-space is contained in that of ABF. In particular, the optimal value of the continuous relaxation of PBF is at least that of ABF.*

Proposition 2. *There exist instances for which, for the same $\mathcal{X}$, the optimal value of the continuous relaxation of PBF is strictly larger than that of ABF.*

Therefore, PBF provides a never weaker, and in some cases strictly stronger continuous relaxation than ABF. The only observable drawback is that ABF has $2n$ point coordinates and n^2 conic constraints (n neighborhood constraints plus $n(n-1)$ distance constraints), while PBF uses $4n(n-1)$ point coordinates, $3n(n-1)$ conic constraints, and $2n$ linear continuity equalities.

Regarding SBF, it can be shown that its relaxation is incomparable to those of ABF and PBF. In order to prove such a result, we first show a closed form for the optimal value of the relaxation of SBF. Let $\bar{c}$ and $\bar{r}$ be the averages of the non-depot centers and radii:

$$\bar{c} = \frac{1}{n-1} \sum_{i \in N \setminus \{1\}} c_i, \qquad \bar{r} = \frac{1}{n-1} \sum_{i \in N \setminus \{1\}} r_i,$$

and denote by V^F the optimal value of the continuous relaxation of formulation F. Then, Proposition 3 uses both values to characterize the optimal value of the SBF relaxation. With this result, Proposition 4 illustrates the incomparability of the formulations through small examples.

Proposition 3. *The optimal value of the continuous relaxation of SBF is*

$$V^{SBF} = 2 \max \{0, \|c_1 - \bar{c}\| - \bar{r}\}.$$

Proposition 4. *For any formulation of $\mathcal{X}$, the continuous relaxation of SBF is incomparable with those of ABF and PBF.*

3.2 Symmetry-breaking inequalities

Since the Euclidean distance is symmetric, a tour and its reverse have the same cost, so every optimal tour has a reversed counterpart with the same value, and branch-and-bound may explore both. A common way of handling such symmetries is to add inequalities that exclude all but one solution from each set of symmetric solutions [28]. In our case, reversing a tour exchanges the successor and the predecessor of the depot, so requiring the successor to have the smaller index keeps exactly one orientation of each tour. For ABF and PBF, the successor and predecessor of the depot are given by $\sum_{j \in N \setminus \{1\}} j\, x_{1j}$ and $\sum_{i \in N \setminus \{1\}} i\, x_{i1}$, respectively, which yields

$$\sum_{j \in N \setminus \{1\}} j\, x_{1j} + 1 \leq \sum_{i \in N \setminus \{1\}} i\, x_{i1}. \tag{5}$$

For SBF, as the depot is fixed at the first stage by (4c), its successor and predecessor are the locations visited at stages 2 and n, which yields

$$\sum_{i \in N \setminus \{1\}} i\, z_{i2} + 1 \leq \sum_{i \in N \setminus \{1\}} i\, z_{in}. \tag{6}$$

Since both members of each reversal pair have the same cost, and exactly one of them satisfies these inequalities, adding them leaves the optimal value unchanged. We add these inequalities to all formulations, including those of this section, the approximated formulations in Section 4, and the master problems in Section 5.

4 Approximated MILP formulations

Given the inherent complexity of solving the previous problems due to their second-order cone constraints and the superior performance of state-of-the-art MILP solvers, it may help to approximate these conic constraints with polyhedral representations. For this purpose, we rely on the work of Ben-Tal and Nemirovski [29], who propose a linear approximation of conic constraints that yields a feasible region arbitrarily close to the original one, with a limited number of variables and constraints.

Formally, let $\epsilon > 0$ and $\mathbf{L}^2 = \{(y, w) \in \mathbb{R}^2 \times \mathbb{R} : w \geq ||y||\}$ be the 3-dimensional Lorentz cone. We say that a polyhedron $P_\epsilon \subseteq \mathbb{R}^2 \times \mathbb{R}$ is an ϵ-approximation of $\mathbf{L}^2$ if $\mathbf{L}^2 \subseteq P_\epsilon$ and every point $(y, w) \in P_\epsilon$ is such that $||y|| \leq (1 + \epsilon)w$. In this context, Ben-Tal and Nemirovski [29] show that $P_{\epsilon(\nu)}$ is an $\epsilon(\nu)$-approximation of $\mathbf{L}^2$, where $\nu \in \mathbb{Z}_+$ is a construction parameter that determines the precision, $\epsilon(\nu)$ is given by

$$\epsilon(\nu) = \left(\cos\left(\frac{\pi}{2^{\nu+1}}\right)\right)^{-1} - 1,$$

and the polyhedron $P_{\epsilon(\nu)}$ is defined as:

$$\begin{aligned}
P_{\epsilon(\nu)} := \Bigg\{ & (y, w) \in \mathbb{R}^2 \times \mathbb{R} : \\
& \xi^0 \geq |y_1|, && && \text{(7a)} \\
& \eta^0 \geq |y_2|, && && \text{(7b)} \\
& \xi^j = \cos\left(\frac{\pi}{2^{j+1}}\right)\xi^{j-1} + \sin\left(\frac{\pi}{2^{j+1}}\right)\eta^{j-1}, && \forall j \in \{1, \ldots, \nu\}, && \text{(7c)} \\
& \eta^j \geq \left|-\sin\left(\frac{\pi}{2^{j+1}}\right)\xi^{j-1} + \cos\left(\frac{\pi}{2^{j+1}}\right)\eta^{j-1}\right|, && \forall j \in \{1, \ldots, \nu\}, && \text{(7d)} \\
& \xi^\nu \leq w, && && \text{(7e)} \\
& \eta^\nu \leq \tan\left(\frac{\pi}{2^{\nu+1}}\right)\xi^\nu, && && \text{(7f)} \\
& \xi^j, \eta^j \in \mathbb{R}, && \forall j \in \{0, \ldots, \nu\} \Bigg\}. && \text{(7g)}
\end{aligned}$$

Thus, the approximation is given by the projection onto the (y, w)-space of the polyhedron defined by the system of inequalities (7a)-(7g). We note that the number of constraints and new variables grows linearly with ν; in particular, the equivalent linear system has $2(\nu+1)$ additional variables and $3(\nu+2)$ constraints. As noted by Vielma et al. [30], this slow growth is what allows a fixed relaxation to be used throughout a branch-and-bound search, at the cost of having to select the approximation quality a priori. Moreover, small values of ν already yield close approximations (e.g., setting $\nu = 3$ yields $\epsilon(\nu) < 0.02$), thus the resulting polyhedron $P_{\epsilon(\nu)}$ remains manageable in terms of dimensionality.

Figure 2 illustrates the approximation's evolution as ν increases, showing cross-sections of $P_{\epsilon(\nu)}$ and the Lorentz cone with $w = 1$ for different values of ν. In each subfigure, the inner disk is the Lorentz cone cross-section, while the outer polygon is

its approximation. This figure shows that the approximation approaches the Lorentz cone as ν increases and is already quite close for small ν.

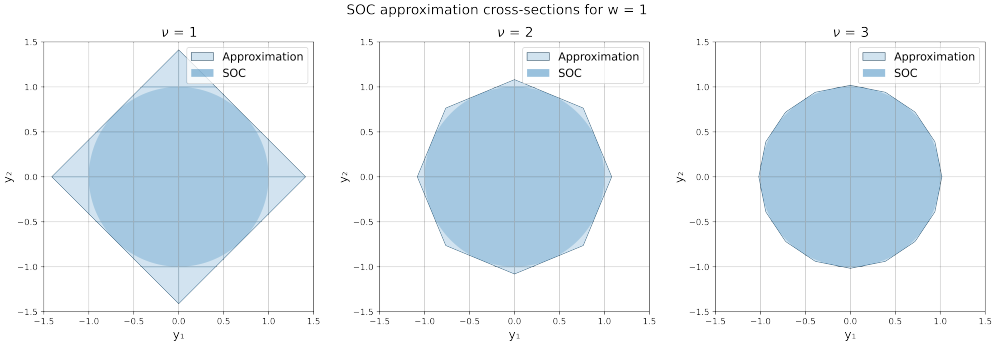


Fig. 2: Cross-sections of the Lorentz cone and its approximation for different values of ν

We use this approximation to transform our MISOCP models into MILP models. For ABF, the MILP approximation (ABF-A) is:

$$\min \sum_{(i,j)\in A} d_{ij} \tag{ABF-A}$$

$$\text{s.t. } (2a), (2d), (2e),$$

$$(p_i - c_i, r_i) \in P_{\epsilon(\nu)}, \qquad \forall i \in N, \tag{8a}$$

$$\left(p_i - p_j, d_{ij} + M_{ij}^{\epsilon(\nu)}(1 - x_{ij})\right) \in P_{\epsilon(\nu)}, \qquad \forall (i,j) \in A. \tag{8b}$$

The objective function is the same as in the original formulation; the first constraint group contains the unchanged constraints from the original model; and (8a) and (8b) ensure that the solution lies within the approximation of the conic constraints. Regarding the big-M constants, they are redefined as $M_{ij}^{\epsilon(\nu)} = \|c_i - c_j\| + (1 + \epsilon(\nu))(r_i + r_j)$ to account for the relaxation in the neighborhoods derived from the approximation factor.

Similarly, we apply this approximation to obtain a MILP formulation (PBF-A) for the perspective reformulation:

$$\min \sum_{(i,j)\in A} d_{ij} \tag{PBF-A}$$

$$\text{s.t. } (3a), (3e), (3f), (3g),$$

$$\left(p_i^{ij} - x_{ij}c_i, x_{ij}r_i\right) \in P_{\epsilon(\nu)}, \qquad \forall (i,j) \in A, \tag{9a}$$

$$\left(p_j^{ij} - x_{ij}c_j, x_{ij}r_j\right) \in P_{\epsilon(\nu)}, \qquad \forall (i,j) \in A, \tag{9b}$$

$$\left(p_i^{ij} - p_j^{ij}, d_{ij}\right) \in P_{\epsilon(\nu)}, \qquad \forall (i,j) \in A. \tag{9c}$$

As before, the objective function remains unchanged from PBF, the first constraint group contains the unchanged constraints, and (9a), (9b), and (9c) enforce that the solution lies in the approximation of the original feasible region.

Lastly, we apply this approximation to obtain a MILP formulation (SBF-A) for the sequence case:

$$\min \sum_{k \in N} t_k \tag{SBF-A}$$

$$\text{s.t. } (4a) - (4c), (4f) - (4g),$$

$$(q_k - q_{k+1}, t_k) \in P_{\epsilon(\nu)}, \qquad \forall k \in N, \tag{10a}$$

$$\left(\sum_{i \in N} z_{ik} c_i - q_k, \sum_{i \in N} z_{ik} r_i\right) \in P_{\epsilon(\nu)}, \qquad \forall k \in N. \tag{10b}$$

As in the previous formulations, the objective function remains unchanged from SBF. The first constraint group states the non-conic constraints, and (10b) with (10a) force the solution to lie in the polyhedral approximation of the original feasible region induced by the conic constraints.

The feasible regions obtained by these approximations are slightly larger than those of the original problem because we use an outer approximation for each conic constraint. Thus, models ABF-A, PBF-A, and SBF-A provide lower bounds for the original problem, and as such, the optimal solutions obtained by such models might be infeasible for the original problem. To obtain a feasible solution (and thus an upper bound), one can take the optimal visit sequence of either ABF-A, PBF-A, or SBF-A, fix the binary variables in the original formulation, and solve for the remaining variables using a SOCP model. Indeed, we use a similar idea to obtain optimal solutions from these MILP approximations, which we further detail in the following section.

5 Decomposition schemes

We now present decomposition schemes for all of our formulations. For MISOCP formulations, these decompositions aim to reduce the complexity inherent in these models, whereas for MILP approximations, they aim to close the optimality gap introduced by the linear approximation.

All decompositions have the following structure: (i) a master problem that is a relaxation of the original problem and, as such, estimates the distance traveled; (ii) a subproblem that computes the actual optimal distance for the visit sequence obtained in the master problem. Specifically, the master problem provides an integer solution associated with the tour sequence and an estimate of the tour length, while a variable $\theta \geq 0$ represents the difference between this estimate and the actual length. If the estimation underestimates the subproblem value, a cut is generated and added to the master problem as an affine function $g(\cdot)$ of the master variables, collected in a set $\mathcal{C}$ that is initially empty and imposed through constraints $\theta \geq g(\cdot)$. This procedure is

iterated within a branch-and-cut framework until optimality is certified by the master problem bounds and the feasible tour solutions.

This structure is related to the lifted LP branch-and-bound algorithm of Vielma et al. [31] (see also Vielma et al. [30]), which replaces the conic relaxation at each node with the polyhedral approximation of Ben-Tal and Nemirovski [29] and, whenever an integer solution is found, solves the original conic problem with the integer variables fixed to recover a feasible solution. The main difference is how the approximation gap is closed: the former branches on integer-feasible solutions, whereas our decompositions add cuts to the master problem, which, as discussed in the subsections below, also lets the subproblem be solved over a much smaller set of conic constraints.

We employ two different types of cuts. The first are enumerative cuts based on the integer L-shaped method [32], which apply to any of the decompositions. Given an integer master solution $\hat{x}$ with support $S(\hat{x})$, let $\hat{Q}$ denote the difference between the optimal length of its tour and the master's estimate. These cuts take the form

$$\theta \geq \hat{Q} - \frac{\hat{Q}}{\kappa} \sum_{(i,j)\in S(\hat{x})} (1 - x_{ij}), \tag{11}$$

where κ is the minimum number of entries of $S(\hat{x})$ in which any other feasible solution differs from $\hat{x}$. For arc variables, $\kappa = 3$, as two distinct directed tours cannot differ in only two arcs. For sequence assignment variables, $\kappa = 2$, as exchanging the locations visited at two stages changes exactly two entries. The right-hand side of (11) equals $\hat{Q}$ at $\hat{x}$, while it is nonpositive at any other feasible tour, where it is dominated by $\theta \geq 0$. For masters based on MILP approximations, $\hat{Q}$ provides a valid lower bound on the correction associated with the best approximate distance estimate for that tour. However, the cut need not capture this correction entirely, as $\hat{Q}$ depends on the current continuous master solution. Therefore, the same tour may need to be considered multiple times, potentially generating additional cuts. The second are dual cuts, obtained from the dual of the subproblem; these depend on the formulation and are presented in the corresponding subsections. In both cases, we add a cut whenever the master solution $(\hat{x}, \hat{\theta})$ satisfies $\hat{\theta} < \hat{Q}$.

5.1 Arc-based decompositions

We propose two decompositions based on ABF: ABF-D, whose master uses distance underestimators for each arc to estimate a tour's distance, and ABF-A-D, whose master uses the MILP approximation ABF-A to estimate that distance.

5.1.1 Decomposition with distance underestimators

In ABF-D, the master problem is given by

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} m_{ij} x_{ij} + \theta \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{ABF-D}$$

$$\theta \geq 0,$$
$$\theta \geq g(x), \qquad \forall g \in \mathcal{C},$$

where m_{ij} is an underestimator of the distance between neighborhoods i and j, for which we use the minimum possible distance between said neighborhoods (i.e., $m_{ij} = \max\{0, \|c_i - c_j\| - r_i - r_j\}$). Given an integer master solution $\hat{x}$, the subproblem is

$$Q^A(\hat{x}) = \min \sum_{(i,j)\in S(\hat{x})} (d_{ij} - m_{ij}) \tag{12a}$$
$$\text{s.t. } \|p_i - c_i\| \leq r_i, \qquad \forall i \in N, \tag{12b}$$
$$\|p_i - p_j\| \leq d_{ij}, \qquad \forall (i,j) \in S(\hat{x}), \tag{12c}$$
$$d_{ij} \geq 0, \qquad \forall (i,j) \in S(\hat{x}), \tag{12d}$$
$$p_i \in \mathbb{R}^2, \qquad \forall i \in N. \tag{12e}$$

Note that, since we instantiate the subproblem over $S(\hat{x})$, the subproblem has $2n$ conic constraints instead of the n^2 conic constraints in ABF. In case it is necessary, cuts (11) are added, with $\hat{Q} = Q^A(\hat{x})$ and $\kappa = 3$.

5.1.2 Decomposition with MILP approximation

The underestimators m_{ij} used in ABF-D can be very loose (e.g., zero for overlapping neighborhoods). To strengthen the master, we instead use the approximated formulation ABF-A, thus obtaining ABF-A-D, whose master problem is

$$\min \sum_{(i,j)\in A} d_{ij} + \theta \tag{ABF-A-D}$$
$$\text{s.t. } (2a), (2d), (2e), (8a), (8b),$$
$$\theta \geq 0,$$
$$\theta \geq g(x), \qquad \forall g \in \mathcal{C},$$

while, given a master solution $(\hat{x}, \hat{d})$ with $\hat{x}$ integer, the subproblem is

$$Q^A_\nu(\hat{x}, \hat{d}) = \min \left\{ \sum_{(i,j)\in S(\hat{x})} d_{ij} : (12b)\text{–}(12e) \right\} - \sum_{(i,j)\in A} \hat{d}_{ij},$$

where hats denote values fixed at the current master solution. When necessary, cuts (11) are added, with $\hat{Q} = Q^A_\nu(\hat{x}, \hat{d})$ and $\kappa = 3$.

Remark. *Dual cuts are not used in the decompositions based on ABF. In both of them, x only appears implicitly in the subproblem, through the definition of the support. Making it explicit requires carrying over the big-M constraints (2c), so the resulting cuts involve the big-M constants, and, in our preliminary experiments, they led to worse results than cuts (11).*

5.2 Perspective-based decompositions

As in Section 5.1, we present two decompositions based on PBF: PBF-D, whose master uses the distance underestimators m_{ij}, and PBF-A-D, whose master is PBF-A.

5.2.1 Decomposition with distance underestimators

In PBF-D, m_{ij} is defined as in ABF-D, and the master problem is given by

$$\begin{aligned} \min \ & \sum_{(i,j)\in A} m_{ij}x_{ij} + \theta && && \text{(PBF-D)} \\ \text{s.t. } & x \in \mathcal{X}, \\ & \theta \geq 0, \\ & \theta \geq g(x), && \forall g \in \mathcal{C}. \end{aligned}$$

Given an integer solution $\hat{x}$, the subproblem is

$$\begin{aligned} Q^P(\hat{x}) = \min \ & \sum_{(i,j)\in A} (d_{ij} - m_{ij}\hat{x}_{ij}) && && \text{(13a)} \\ \text{s.t. } & \|p_i^{ij} - \hat{x}_{ij}c_i\| \leq \hat{x}_{ij}r_i, && \forall (i,j) \in A, && \text{(13b)} \\ & \|p_j^{ij} - \hat{x}_{ij}c_j\| \leq \hat{x}_{ij}r_j, && \forall (i,j) \in A, && \text{(13c)} \\ & \|p_i^{ij} - p_j^{ij}\| \leq d_{ij}, && \forall (i,j) \in A, && \text{(13d)} \\ & \sum_{j:(i,j)\in A} p_i^{ij} = \sum_{k:(k,i)\in A} p_i^{ki}, && \forall i \in N, && \text{(13e)} \\ & d_{ij} \in \mathbb{R},\ p_i^{ij}, p_j^{ij} \in \mathbb{R}^2, && \forall (i,j) \in A, && \text{(13f)} \end{aligned}$$

where the distance variables are free, as their nonnegativity is implied by (13d).

Cuts (11) can also be used in this decomposition, with $\kappa = 3$. However, at an integer $\hat{x}$, constraints (13b) and (13c) force the copies of off-support arcs to vanish, and the subproblem reduces to that of ABF-D (see Appendix C), so $Q^P(\hat{x}) = Q^A(\hat{x})$. As the master problems also coincide, PBF-D with enumerative cuts is identical to ABF-D. The perspective reformulation is thus only useful here through its dual, which, unlike that of ABF, yields cuts that do not involve big-M constants.

This subproblem admits the following dual formulation:

$$\begin{aligned} Q^P(\hat{x}) = \max \ & -\sum_{(i,j)\in A} \hat{x}_{ij}\Big(m_{ij} + c_i^\top \alpha_{ij}^i + r_i\lambda_{ij}^i \\ & \qquad + c_j^\top \alpha_{ij}^j + r_j\lambda_{ij}^j\Big) && && \text{(14a)} \\ \text{s.t. } & \|\alpha_{ij}^i\| \leq \lambda_{ij}^i, && \forall (i,j) \in A, && \text{(14b)} \\ & \|\alpha_{ij}^j\| \leq \lambda_{ij}^j, && \forall (i,j) \in A, && \text{(14c)} \\ & \|\gamma_{ij}\| \leq 1, && \forall (i,j) \in A, && \text{(14d)} \end{aligned}$$

$$\alpha^i_{ij} = \eta_i - \gamma_{ij}, \qquad \forall (i,j) \in A, \tag{14e}$$
$$\alpha^j_{ij} = \gamma_{ij} - \eta_j, \qquad \forall (i,j) \in A, \tag{14f}$$

where $(\alpha^i_{ij}, \lambda^i_{ij})$ and $(\alpha^j_{ij}, \lambda^j_{ij})$ are the dual variables associated with (13b) and (13c), γ_{ij} with (13d), and $\eta_i \in \mathbb{R}^2$ with (13e). As $\hat{x}$ appears only in the dual objective, which is linear in it, the dual feasible region does not depend on $\hat{x}$. Hence, by weak duality, every dual feasible point gives a valid cut of the form

$$\theta \geq - \sum_{(i,j)\in A} (m_{ij} + C_{ij})\, x_{ij}, \tag{15}$$

with $C_{ij} := c_i^\top \alpha^i_{ij} + r_i \lambda^i_{ij} + c_j^\top \alpha^j_{ij} + r_j \lambda^j_{ij}$, which is tight at $\hat{x}$ if the dual point is optimal. In what follows, we show how to reduce the dual (14) to improve the cut generation procedure.

Note that off-support arcs have $\hat{x}_{ij} = 0$, so their variables do not appear in the dual objective and only need to be feasible. Moreover, for any values of η, they can always be completed feasibly: for any γ_{ij} with $\|\gamma_{ij}\| \leq 1$, setting α^i_{ij} and α^j_{ij} through (14e) and (14f), and $\lambda^i_{ij} = \|\alpha^i_{ij}\|$, $\lambda^j_{ij} = \|\alpha^j_{ij}\|$, yields a dual feasible point. Hence, the dual can be solved over $S(\hat{x})$ alone, and any such completion of the off-support variables gives an optimal dual solution.

Over $S(\hat{x})$, the dual can be further reduced. Let $\mathrm{nxt}(i)$ and $\mathrm{prv}(i)$ denote the successor and predecessor of location i in the tour given by $\hat{x}$, so that each location has exactly one outgoing arc, $(i, \mathrm{nxt}(i))$, and one incoming arc, $(\mathrm{prv}(i), i)$, in $S(\hat{x})$. First, as the objective is nonincreasing in λ, we can set $\lambda^i_{ij} = \|\alpha^i_{ij}\|$ and $\lambda^j_{ij} = \|\alpha^j_{ij}\|$. Then, substituting (14e)–(14f), the term of each arc $(i,j) \in S(\hat{x})$ becomes

$$-m_{ij} - c_i^\top (\eta_i - \gamma_{ij}) - r_i \|\eta_i - \gamma_{ij}\| - c_j^\top (\gamma_{ij} - \eta_j) - r_j \|\gamma_{ij} - \eta_j\|.$$

When summing over $S(\hat{x})$, the linear terms in η cancel, as each η_i appears with $-c_i^\top \eta_i$ in its outgoing arc and with $+c_i^\top \eta_i$ in its incoming arc. Grouping the norm terms by location, the dual objective becomes

$$\sum_{(i,j)\in S(\hat{x})} \left((c_i - c_j)^\top \gamma_{ij} - m_{ij} \right) - \sum_{i \in N} r_i \left(\|\eta_i - \gamma_{i\,\mathrm{nxt}(i)}\| + \|\gamma_{\mathrm{prv}(i)\,i} - \eta_i\| \right).$$

Each η_i now appears in a single term, so it can be optimized independently. By the triangle inequality,

$$\|\gamma_{i\,\mathrm{nxt}(i)} - \gamma_{\mathrm{prv}(i)\,i}\| \leq \|\gamma_{i\,\mathrm{nxt}(i)} - \eta_i\| + \|\eta_i - \gamma_{\mathrm{prv}(i)\,i}\|,$$

with equality whenever η_i lies on the segment between $\gamma_{\mathrm{prv}(i)\,i}$ and $\gamma_{i\,\mathrm{nxt}(i)}$, in particular at its midpoint. Since $r_i \geq 0$, the optimal value of each term is therefore $-r_i \|\gamma_{i\,\mathrm{nxt}(i)} - \gamma_{\mathrm{prv}(i)\,i}\|$. Introducing auxiliary variables τ_i for these norms, the dual

becomes

$$Q^P(\hat{x}) = \max \sum_{(i,j)\in S(\hat{x})} \left((c_i - c_j)^\top \gamma_{ij} - m_{ij}\right) - \sum_{i\in N} r_i \tau_i \tag{16a}$$

$$\text{s.t. } \left\|\gamma_{i\,\mathrm{nxt}(i)} - \gamma_{\mathrm{prv}(i)\,i}\right\| \le \tau_i, \qquad \forall i \in N, \tag{16b}$$

$$\|\gamma_{ij}\| \le 1, \qquad \forall (i,j) \in S(\hat{x}), \tag{16c}$$

$$\gamma_{ij} \in \mathbb{R}^2, \qquad \forall (i,j) \in S(\hat{x}), \tag{16d}$$

$$\tau_i \in \mathbb{R}, \qquad \forall i \in N. \tag{16e}$$

The dual problem (16) has $3n$ variables and $2n$ conic constraints, the same number as the subproblem of ABF-D. Given an optimal $\hat{\gamma}$, we recover $\hat{\eta}_i = \frac{1}{2}\left(\hat{\gamma}_{i\,\mathrm{nxt}(i)} + \hat{\gamma}_{\mathrm{prv}(i)\,i}\right)$, and $\hat{\alpha}$ and $\hat{\lambda}$ from the expressions above.

With these values, we can compute the coefficients of (15). Substituting (14e)–(14f), with λ set to the corresponding norms, the coefficient of each arc $(i,j) \in A$ becomes

$$C_{ij} = c_i^\top \hat{\eta}_i - c_j^\top \hat{\eta}_j + h_{ij}(\tilde{\gamma}_{ij}), \quad h_{ij}(\gamma) := (c_j - c_i)^\top \gamma + r_i \|\hat{\eta}_i - \gamma\| + r_j \|\hat{\eta}_j - \gamma\|, \tag{17}$$

where $\tilde{\gamma}_{ij} = \hat{\gamma}_{ij}$ for $(i,j) \in S(\hat{x})$, while, for $(i,j) \notin S(\hat{x})$, $\tilde{\gamma}_{ij}$ can be any point with $\|\tilde{\gamma}_{ij}\| \le 1$, as shown above. In particular, $\tilde{\gamma}_{ij} = 0$ is always a valid choice. Moreover, as the coefficients C_{ij} enter (15) with a negative sign, smaller values yield stronger cuts, so the strongest choice off the support is any $\tilde{\gamma}_{ij} \in \arg\min_{\|\gamma\|\le 1} h_{ij}(\gamma)$. We compute $\tilde{\gamma}_{ij}$ with a projected variant of Weiszfeld's algorithm [33–36], which decreases h_{ij} monotonically (see Proposition 5 in Appendix B). As every point of the unit ball yields a valid cut, we stop after a fixed number of iteration (see details in Appendix B).

These cuts are particularly useful when neighborhoods overlap. In that case, a group of consecutive locations of the tour given by $\hat{x}$ can often be visited in a different order at the same total length. Enumerative cuts (11) strengthen the master only at $\hat{x}$, so such reorderings may require additional cuts. In contrast, when the tour passes straight through these locations, cut (15) is tight for every reordering that yields a tour of the same length, provided the dual solution is constant along the corresponding stretch and the off-support coefficients satisfy $h_{ij}(\tilde{\gamma}_{ij}) \le \min\{h_{ij}(\hat{\eta}_i), h_{ij}(\hat{\eta}_j)\}$. The precise statement and proof appear in Appendix C.

5.2.2 Decomposition with MILP approximation

As in Section 5.1, the underestimators m_{ij} can be very loose, so we strengthen the master by using the approximated formulation PBF-A, thus obtaining PBF-A-D, whose master problem is

$$\min \sum_{(i,j)\in A} d_{ij} + \theta \tag{PBF-A-D}$$

$$\text{s.t. } (3a), (3e), (3f), (3g), (9a), (9b), (9c),$$

$$\theta \ge 0,$$

$$\theta \geq g(x,d), \qquad \forall g \in \mathcal{C},$$

while, given a master solution $(\hat{x},\hat{d})$ with $\hat{x}$ integer, the subproblem is

$$Q^P_\nu(\hat{x},\hat{d}) = \min\left\{\sum_{(i,j)\in A} d_{ij} : (13b)\text{–}(13f)\right\} - \sum_{(i,j)\in A} \hat{d}_{ij},$$

where hats denote values fixed at the current master solution. When necessary, cuts (11) are added, with $\hat{Q} = Q^P_\nu(\hat{x},\hat{d})$ and $\kappa = 3$.

Alternatively, dual cuts can be derived as in PBF-D. Since $Q^P_\nu(\hat{x},\hat{d})$ differs from $Q^P(\hat{x})$ only by a constant, (16) has the same optimal solutions in both cases, so the coefficients C_{ij} in (17) are the same, and the resulting cut is

$$\theta \geq -\sum_{(i,j)\in A} (C_{ij}x_{ij} + d_{ij}). \tag{18}$$

This cut is valid because (15) implies that $-\sum_{(i,j)\in A} C_{ij}x_{ij}$ never exceeds the length of the tour given by x, and it is tight at $(\hat{x},\hat{d})$. Moreover, because its dependence on x matches that of (15), the results on reorderings carry over.

5.3 Sequence-based decompositions

We now present two decompositions based on SBF: SBF-D, whose master provides no distance estimate beyond θ, and SBF-A-D, whose master uses the MILP approximation SBF-A.

5.3.1 Decomposition without distance estimation

In SBF-D, the master problem is given by

$$\begin{aligned} &\min\ \theta && && \text{(SBF-D)}\\ &\text{s.t. } (4a)\text{–}(4c), (4f), && &&\\ &\quad \theta \geq 0, && &&\\ &\quad \theta \geq g(z), && \forall g \in \mathcal{C}. && \end{aligned}$$

Since the master provides no estimate of the tour length, θ represents the full tour length. Given an integer solution $\hat{z}$, the subproblem is

$$Q^S(\hat{z}) = \min \sum_{k\in N} t_k \tag{19a}$$

$$\text{s.t. } \|q_k - q_{k+1}\| \leq t_k, \qquad \forall k \in N, \tag{19b}$$

$$\left\|\sum_{i\in N} \hat{z}_{ik}c_i - q_k\right\| \leq \sum_{i\in N} \hat{z}_{ik}r_i, \qquad \forall k \in N, \tag{19c}$$

$$q_k \in \mathbb{R}^2,\ t_k \geq 0, \qquad \forall k \in N, \tag{19d}$$

which has $2n$ conic constraints.

Cuts (11) can be used with $\kappa = 2$, but as all information about the tour length comes from the cuts, they would effectively require enumerating sequences. Thus, as in Section 5.2, we use dual cuts. Constraints (19c) of stages with zero radius reduce to linear equalities, while the remaining conic constraints admit a strictly feasible point, so strong duality holds [37]. Then, the dual model is given by

$$\begin{aligned}
Q^S(\hat{z}) = \max\ & \sum_{i \in N} \sum_{k \in N} \mu_{ik} \hat{z}_{ik} && && \text{(20a)}\\
\text{s.t. } & 1 - \lambda_k^n - \lambda_k^d = 0, && \forall k \in N, && \text{(20b)}\\
& \mu_{ik} + c_i^\top \rho_k^c + r_i \lambda_k^c = 0, && \forall i, k \in N, && \text{(20c)}\\
& \rho_k^d - \rho_{k-1}^d + \rho_k^c = 0, && \forall k \in N, && \text{(20d)}\\
& \|\rho_k^d\| \leq \lambda_k^d, && \forall k \in N, && \text{(20e)}\\
& \|\rho_k^c\| \leq \lambda_k^c, && \forall k \in N, && \text{(20f)}\\
& \lambda_k^n, \lambda_k^d, \lambda_k^c \geq 0, && \forall k \in N, && \text{(20g)}\\
& \mu_{ik} \in \mathbb{R}, && \forall i, k \in N, && \text{(20h)}\\
& \rho_k^d, \rho_k^c \in \mathbb{R}^2, && \forall k \in N, && \text{(20i)}
\end{aligned}$$

where variables with superscripts d, c and n are associated with constraints (19b), (19c), and the nonnegativity constraints in (19d), respectively. Abusing the notation, we consider $k - 1 = n$ in (20d) for the case when $k = 1$. Variables μ_{ik} are fully determined by (20c), but we keep them as is, given they are the coefficients of the resulting cuts. As $\hat{z}$ appears only in the dual objective, the dual feasible region does not depend on $\hat{z}$, so, by weak duality, every dual feasible point $\hat{\mu}$ gives a valid cut of the form

$$\theta \geq \sum_{i \in N} \sum_{k \in N} \hat{\mu}_{ik} z_{ik}, \tag{21}$$

which is tight at $\hat{z}$ if the dual point is optimal.

5.3.2 Decomposition with MILP approximation

As the master of SBF-D provides no estimation of the tour length, we strengthen it by using the approximated formulation SBF-A, thus obtaining SBF-A-D, whose master problem is

$$\begin{aligned}
\min\ & \sum_{k \in N} t_k + \theta && && \text{(SBF-A-D)}\\
\text{s.t. } & \text{(4a)–(4c)}, \text{(4f)–(4g)}, \text{(10a)}, \text{(10b)}, && &&\\
& \theta \geq 0, && &&\\
& \theta \geq g(z, t), && \forall g \in \mathcal{C}. &&
\end{aligned}$$

Given a master solution $(\hat{z}, \hat{t})$ with $\hat{z}$ integer, the subproblem is $Q^S_\nu(\hat{z}, \hat{t}) = Q^S(\hat{z}) - \sum_{k \in N} \hat{t}_k$. When necessary, cuts (11) are added, with $\hat{Q} = Q^S_\nu(\hat{z}, \hat{t})$ and $\kappa = 2$.

Alternatively, dual cuts can be derived as in SBF-D. Since $Q^S_\nu(\hat{z}, \hat{t})$ differs from $Q^S(\hat{z})$ only by a constant, (20) yields the same $\hat{\mu}$, and the resulting cut is

$$\theta \geq \sum_{i \in N} \sum_{k \in N} \hat{\mu}_{ik} z_{ik} - \sum_{k \in N} t_k, \tag{22}$$

which is valid, as (21) implies that $\sum_{i \in N} \sum_{k \in N} \hat{\mu}_{ik} z_{ik}$ never exceeds the length of the tour given by z, and is tight at $(\hat{z}, \hat{t})$.

6 On the applicability of generalized Benders

The decompositions presented in the previous section utilized cuts that leverage SOCP duality or the binary nature of the master problem variables (i.e., enumerative cuts). An alternative is to consider Generalized Benders Decomposition (GBD) [10], which was proposed to solve the PDTSPN problem [9], a variant of the CETSP. In what follows, we explain some of the basic concepts regarding GBD and show that the cuts originally proposed by Gao et al. [9] for the PDTSPN are not valid for the CETSP (and the PDTSPN), and thus the resulting decomposition does not guarantee optimality.

As its name suggests, GBD was proposed as an extension of the classic Benders decomposition [38] to handle non-linear subproblems, allowing the decomposition of problems in which the subproblem is a convex optimization problem, with applications to some non-convex problems as well [39]. The approach considers problems of the following form:

$$\min_{x,y} \{f(x,y) : \ G(x,y) \leq 0, x \in X, y \in Y\}$$

where X and Y are real-valued domains, $f : X \times Y \to \mathbb{R}$ is the objective function, and $G : X \times Y \to \mathbb{R}^m$ is a vector of constraint functions. This problem can be re-stated as $\min_x \{v(x) : \ x \in X \cap V\}$, where $v(x) = \inf_{y \in Y} \{f(x,y) \text{ s.t. } G(x,y) \leq 0\}$ is the optimal value of the problem for a fixed x, and $V = \{x \in X : G(x,y) \leq 0 \text{ for some } y \in Y\}$ is the projection of the feasible region of the original problem onto the domain of variables x, i.e., the set of all $x \in X$ such that there exists a feasible value assignment for y. Under some fairly weak assumptions on f, G, and Y [10], it holds that

$$v(x) = \sup_{u \geq 0} \left[\inf_{y \in Y} \left\{ f(x,y) + u^\top G(x,y) \right\} \right].$$

With this result, the original problem can be reformulated as

$$\min_{x \in X} \quad \theta \tag{23a}$$

$$\text{s.t.} \quad \theta \geq \inf_{y \in Y} \left\{ f(x,y) + u^\top G(x,y) \right\}, \quad \forall u \geq 0, \tag{23b}$$

$$x \in V. \tag{23c}$$

The logic of the decomposition is to relax constraints (23b) and (23c) to obtain a master problem in variable x. Given a solution $\hat{x}$ of the master problem, the approach iteratively solves the subproblem $v(\hat{x})$ and adds optimality cuts (from (23b)) or feasibility cuts (from (23c)) as needed until convergence. In the case of the CETSP, the subproblem is always feasible, so we focus solely on optimality cuts, which take the form

$$L^*(x, \hat{u}) \leq \theta, \tag{24}$$

where

$$L^*(x, \hat{u}) = \inf_{y \in Y} \left\{ f(x, y) + \hat{u}^\top G(x, y) \right\},$$

and $\hat{u}$ is the optimal dual vector associated with the subproblem $v(\hat{x})$.

An important remark regarding the viability of GBD is that, in order for the cuts (24) to be practical, it is necessary that the function $L^*(\cdot, \hat{u})$ on X can be obtained explicitly in a way that requires little or no more effort than that which is required to evaluate it at a single value of x, which the author of [10] refers to as Property (P). Otherwise, the functional form of the cuts would be too complex to be of any practical use. An example where Property (P) holds is when both f and G are linearly separable in x and y.

In the context of the PDTSPN, Gao et al. [9] propose a formulation similar to ABF, for which they implement a decomposition scheme based on GBD. The resulting master problem in their decomposition is the same as ABF-D (with additional constraints corresponding to the pickup-and-delivery aspect of the problem), and the subproblem is as follows:

$$\begin{aligned}
Q(x) = \min \quad & \sum_{(i,j) \in A} x_{ij}(d_{ij} - m_{ij}) && && \text{(25a)} \\
\text{s.t.} \quad & f_i(p_i) \leq 0, && \forall i \in N, && \text{(25b)} \\
& \|p_i - p_j\| \leq d_{ij}, && \forall (i,j) \in A, && \text{(25c)} \\
& d_{ij} \geq 0, && \forall (i,j) \in A, && \text{(25d)} \\
& p_i \in \mathbb{R}^2, && \forall i \in N, && \text{(25e)}
\end{aligned}$$

where functions f_i represent the neighborhoods, which are assumed to be convex and compact. Note that circular neighborhoods are a special case of these functions. This subproblem differs from the one we present in ABF-D in that it is instantiated over all arcs in A, instead of only over the arcs in the support of x, $S(x)$.

For simplicity, we assume that the distance estimations m_{ij} are all equal to 0. In this context, the function defining cuts (24) is given by

$$L^*(x, \hat{\pi}, \hat{\mu}) = \inf_{\substack{p \in \mathbb{R}^{2n} \\ d \geq 0}} \left\{ \sum_{(i,j) \in A} x_{ij} d_{ij} + \sum_i \hat{\pi}_i f_i(p_i) + \sum_{(i,j) \in A} \hat{\mu}_{ij} \big(\|p_i - p_j\| - d_{ij} \big) \right\},$$

where $\hat{\pi} \geq 0$ and $\hat{\mu} \geq 0$ are the optimal dual vectors associated with the neighborhood constraints (25b) and the distance constraints (25c), respectively.

Based on this formulation, Gao et al. [9] propose that, given a master problem solution $\hat{x}$, and the corresponding primal and dual optimal solutions of the subproblem $(\hat{p}, \hat{d})$ and $(\hat{\pi}, \hat{\mu})$, respectively, the following cut can be added to the master problem if needed:

$$\sum_{(i,j)\in A} \hat{d}_{ij} x_{ij} + \sum_{i\in N} \hat{\pi}_i f_i(\hat{p}_i) + \sum_{(i,j)\in A} \hat{\mu}_{ij}(\|\hat{p}_i - \hat{p}_j\| - \hat{d}_{ij}) \leq \theta.$$

Since x does not appear in the second two summations, by complementary slackness, these terms are equal to zero, thus simplifying the cut to

$$\sum_{i,j\in N, i\neq j} \hat{d}_{ij} x_{ij} \leq \theta.$$

However, this substitution does not guarantee a valid cut, since

$$L^*(x, \hat{\pi}, \hat{\mu}) \leq \sum_{(i,j)\in A} \hat{d}_{ij} x_{ij} + \sum_{i\in N} \hat{\pi}_i f_i(\hat{p}_i) + \sum_{(i,j)\in A} \hat{\mu}_{ij}(\|\hat{p}_i - \hat{p}_j\| - \hat{d}_{ij}),$$

as $(\hat{p}, \hat{d})$ is a feasible solution to the problem defining $L^*(x, \hat{\pi}, \hat{\mu})$, but not necessarily optimal. Thus, the proposed cuts may overestimate $L^*(x, \hat{\pi}, \hat{\mu})$ and are therefore not necessarily valid. According to Bagajewicz and Manousiouthakis [40], using these cuts can lead to a procedure that converges to non-optimal solutions.

We now show that such cuts can indeed be invalid by analyzing their structure and using a numerical example. First, let us consider the sequence $\tilde{S}$ associated with some first-stage solution $\tilde{x}$ (i.e. $\tilde{S} = \{(i,j) \in A \mid \tilde{x}_{ij} = 1\}$). Given that x is binary, the objective function of the subproblem can be rewritten as

$$\sum_{(i,j)\in A} d_{ij} \tilde{x}_{ij} = \sum_{(i,j)\in \tilde{S}} d_{ij}.$$

Let $\tilde{d}$ denote the corresponding vector of optimal distances obtained by solving the subproblem. Note that, since only entries $(i,j) \in \tilde{S}$ appear in the objective function, and there is no upper bound on any of the distance variables, the optimal distance $\tilde{d}_{ij}$ for $(i,j) \notin \tilde{S}$ can take any non-negative value greater than or equal to $\|p_i - p_j\|$, without affecting the optimality of the solution. Thus, the entries $\tilde{d}_{ij}$ for $(i,j) \notin \tilde{S}$ are not necessarily defined as the distance between the corresponding visit points, and can in fact be arbitrarily large, potentially leading to cuts that incorrectly eliminate feasible solutions.

We now present a numerical example that illustrates how this interpretation of (24) can cut off feasible solutions. Consider an instance with four locations, whose coordinates and radii are given in Table 1, and plotted in Figure 3.

The optimal solution to this instance has a total distance of approximately 13.24, and the optimal visit order is $1 \to 4 \to 2 \to 3 \to 1$ (or the reverse). If we solve the subproblem for the visit order $1 \to 2 \to 3 \to 4 \to 1$, we can compute the distances

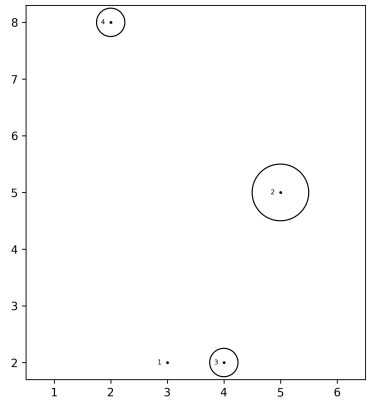


Fig. 3: Plot of the example instance

Location	c^x	c^y	r
1	3	2	0
2	5	5	0.5
3	4	2	0.25
4	2	8	0.25

Table 1: Coordinates and radii

	1	2	3	4
1	0.00	3.11	1.03	5.83
2	3.11	0.00	2.43	4.20
3	1.03	2.43	0.00	5.84
4	5.83	4.20	5.84	0.00

Table 2: Distance matrix for $1 \to 2 \to 3 \to 4 \to 1$

manually given the optimal visit points to avoid the problem mentioned before, thus obtaining the distance matrix presented in Table 2.

Let S^* denote the optimal sequence, $\tilde{S}$ the second sequence, and $\tilde{d}$ its corresponding distance matrix. Suppose we generated the cut associated with $\tilde{S}$:

$$\sum_{(i,j)\in A} x_{ij}\tilde{d}_{ij} \leq \theta,$$

which, given the distance matrix in Table 2, can be rewritten approximately as

$$\begin{aligned} &3.11(x_{12} + x_{21}) + 1.03(x_{13} + x_{31}) + 5.83(x_{14} + x_{41}) \\ &+ 2.43(x_{23} + x_{32}) + 4.20(x_{24} + x_{42}) + 5.84(x_{34} + x_{43}) \leq \theta. \end{aligned}$$

Now, if we evaluate the left-hand side of this cut for the sequence S^*, we obtain

$$\begin{aligned} \sum_{(i,j)\in S^*} \tilde{d}_{ij} &= \tilde{d}_{14} + \tilde{d}_{42} + \tilde{d}_{23} + \tilde{d}_{31} \\ &\approx 5.83 + 4.20 + 2.43 + 1.03 = 13.49, \end{aligned}$$

which is greater than the optimal distance of approximately 13.24.

Thus, the cut obtained by solving the subproblem for sequence $\tilde{S}$ leads to an overestimation of the distance of the optimal sequence S^*, and is therefore incorrect. This example illustrates that the cuts proposed by Gao et al. [9] are invalid, and, as such, the decomposition scheme originally proposed in [9] does not guarantee convergence to the optimal solution of the CETSP, or its extension, the PDTSPN.

Remark. *After an earlier version of this work was made available, Gao et al. [11] released an erratum acknowledging that the cuts originally proposed in [9] may be invalid, and presenting a corrected decomposition scheme based on generalized Benders decomposition that yields valid cuts.*

7 Computational experiments

In this section, we conduct computational experiments to test the different approaches presented in this work. Section 7.1 describes the instances and the experimental setup. Section 7.2 compares the monolithic formulations ABF, PBF, and SBF with their decompositions ABF-D, PBF-D, and SBF-D, to assess the value of decomposition. Then, Section 7.3 compares the monolithic formulations with their MILP approximations ABF-A, PBF-A, and SBF-A, measuring the quality of the distance estimations they provide, and Section 7.4 compares the decompositions with their approximated counterparts ABF-A-D, PBF-A-D, and SBF-A-D, along with the two types of cuts. Next, Section 7.5 analyzes how the number of locations and the radii of the neighborhoods affect the performance of the best-performing decompositions. Finally, Section 7.6 compares our best-performing approach with the discretization scheme of Behdani and Smith [2] (BS).

7.1 Experimental setup

We generate several CETSP instances, where $n \in \{15, 20, 25, 30\}$ is the number of locations, $r \in \{0.25, 0.5, 1\}$ the nominal mean non-depot radius, and $\sigma \in \{0, 0.2, 0.5\}$ the variation among radii. The coordinates of the n center locations of the neighborhoods are randomly chosen from a uniform distribution over a square with side length 10. Then, for each point, a radius is sampled uniformly from $[(1-\sigma)r, (1+\sigma)r]$, except for the depot, whose radius is fixed at 0. For each parameter combination, we generate 5 instances, yielding a total of 180 distinct instances. This instance-generation procedure is based on the one used by Behdani and Smith [2], adding the variability parameter σ to generate more diverse instances. Before solving each instance, we consider a pre-processing step, where, if there is a pair $i, j \in N$ such that the neighborhood of i is wholly contained in the neighborhood of j, we eliminate node j from the problem, since it is trivially visited by any tour that visits the neighborhood of i; n refers to the number of locations before this step.

The approximations of Section 4 are implemented with $\nu = 3$, which yields $\epsilon(\nu) \approx 0.020$. Preliminary experiments show that $\nu = 2$ ($\epsilon(\nu) \approx 0.082$) yielded approximations too loose to provide accurate distance estimations, while $\nu = 4$ ($\epsilon(\nu) \approx 0.005$) yielded only marginal improvements in the quality of the estimations at the cost of larger models. Thus, $\nu = 3$ shows the best trade-off between quality and computational time.

In all formulations based on arc variables, $\mathcal{X}$ is modeled with the single-commodity flow formulation [41]. We strengthen the formulation by separating fractional solutions at the root node by introducing subtour elimination constraints [22], which yielded the best performance in preliminary experiments. All formulations and master problems include the symmetry-breaking inequalities (5) and (6).

ABF-D and ABF-A-D use cuts (11), while PBF-D and SBF-D use the dual cuts (15) and (21), respectively (recall that PBF-D with enumerative cuts is identical to ABF-D). For PBF-A-D and SBF-A-D, we report both types of cuts as separate procedures. We do not combine both types of cuts in a single procedure, as, in preliminary experiments, doing so did not improve performance with respect to using either type alone. The off-support coefficients of the dual cuts of PBF-D and PBF-A-D are computed with the procedure of Appendix B, applying 20 iterations of update (B1). This number was chosen in preliminary experiments, as it yields near-optimal coefficients in most cases without a significant increase in solution time.

To verify the consistency of our results, we checked that, for every instance, the lower bound reported by each procedure does not exceed the upper bound of any procedure, up to a relative tolerance. This check only fails for ABF on instances that are numerically difficult for it, which we discuss in Section 7.2.

All procedures are implemented in Python, using Gurobi 13.0.3 as the solver, with a thirty-minute time limit. Experiments are conducted on an AWS m8a.metal-48xl instance, with 192 cores and 768 GiB of RAM, running 48 problems in parallel, each limited to 4 threads.

7.2 Monolithic vs. decomposed formulations

We first compare the three monolithic formulations, ABF, PBF, and SBF, with their decompositions ABF-D, PBF-D, and SBF-D, across the 180 instances. Figure 4 presents a comparison of the performance of the six procedures as a function of execution time and the optimality gap. On the X-axis, values to the left of the second vertical line represent time, while values to the right correspond to the optimality gap. The Y-axis indicates the number of instances solved. Each colored line represents the performance of a different procedure. The first vertical dotted line marks the 1800-second time limit; thus, the left part of the plot shows the number of instances solved to optimality and the time needed to do so, while the right part presents the gap left on instances not solved to optimality.

Arc-based procedures (i.e., ABF and ABF-D) are plotted in red, perspective-based procedures (i.e., PBF and PBF-D) in purple, and sequence-based procedures (i.e., SBF and SBF-D) in blue. In all cases, the lighter shade corresponds to the monolithic formulation, and the darker shade to its decomposition. On three instances (one with $n = 20$ and two with $n = 25$), Gurobi terminated with an error when solving SBF, which is why its line ends below the total number of instances; these instances are excluded from the results reported for SBF. Table 3 summarizes these results, where column "Procedure" states the approach used, "#Opt" the number of instances solved to optimality within the time limit, "TimeOpt" the average time on instances solved to optimality, "Gap" the average gap across all instances, "Gap (NoOpt)" the average gap on instances not solved to optimality, and "Nodes" the average number of branch-and-bound nodes explored.

We observe that decomposition increases the number of instances solved to optimality for both ABF and PBF, but not for SBF. For ABF, the decomposition solves 31 more instances to optimality than the monolithic formulation (37 vs. 6), although it leaves larger gaps overall (27.82% vs. 24.91%). In contrast, PBF-D improves on PBF

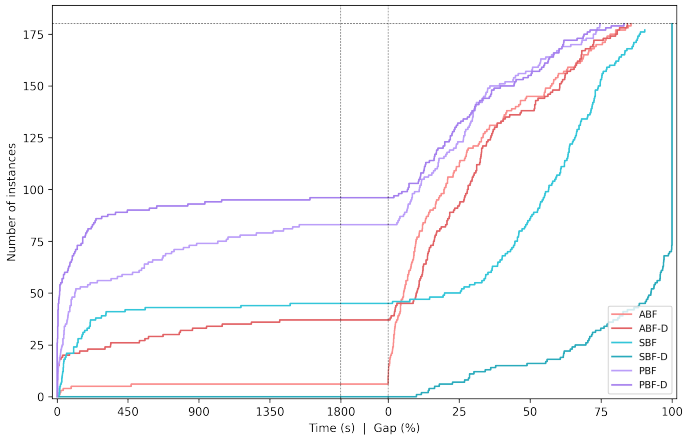


Fig. 4: Performance profile of the monolithic formulations and their decompositions.

Procedure	#Opt	TimeOpt	Gap (%)	Gap (NoOpt) (%)	Nodes
ABF	6	106.95	24.91	25.77	197,738
ABF-D	37	296.01	27.82	35.01	336,128
PBF	83	306.38	17.19	31.89	43,710
PBF-D	96	118.02	16.12	34.54	360,391
SBF	45	178.05	43.78	58.71	484,881
SBF-D	0	-	88.29	88.29	238,701

Table 3: Summary of results for the monolithic formulations and their decompositions. Results for SBF exclude the three instances on which the solver terminated with an error.

overall, solving 13 more instances to optimality (96 vs. 83), in considerably less average time (118.02 seconds vs. 306.38), and with a slightly smaller average gap (16.12% vs. 17.19%). Lastly, SBF-D fails to solve any instance to optimality, with an average gap of 88.29%. This poor performance stems from the absence of distance estimation in its master problem, so all information about tour length comes from cuts generated in the subproblem.

We note that, on some instances, the solver terminates ABF with a suboptimal status, likely due to numerical issues, which are known to arise with big-M constraints (see Section 3). In such cases, the lower bound reported by the solver can exceed the best known upper bound for the instance. This occurs on 11 instances, by at most 1.8%, all but one of which terminate with a suboptimal status. As the gap reported for ABF is computed from this lower bound, it is underestimated on these instances. Thus, the results reported for ABF in this section and the following one are slightly optimistic, which does not affect our conclusions: ABF remains one of the weakest formulations, and its numerical issues are an additional drawback of relying on big-M constraints.

Regarding the comparison between ABF and PBF, we observe that the dominance of the continuous relaxation of PBF established in Propositions 1 and 2 is consistent with its computational performance: PBF solves 83 instances to optimality, against only 6 for ABF, with a considerably smaller average gap. The comparison between ABF-D and PBF-D is particularly interesting, as both decompositions share the same master problem and subproblem value and differ only in the cuts used (see Section 5.2). Thus, the performance difference, with PBF-D solving 59 more instances to optimality in less than half the average time, can be attributed to the dual cuts (15). In fact, both perspective-based procedures dominate ABF-D, including the monolithic formulation PBF, which solves more than twice as many instances to optimality. Moreover, Figure 4 shows that the lines of PBF and PBF-D lie above those of the remaining procedures in both the time and gap sections of the plot.

We also note that SBF solves more instances to optimality than both ABF and ABF-D, but with substantially larger gaps on the remaining ones (58.71% on average, against 25.77% and 35.01%, respectively). Moreover, its performance stagnates early: it solves about 40 instances within the first 300 seconds and only a few more afterward, suggesting that, given the time limit, this formulation either finds an optimal solution quickly or fails to find one altogether. A similar behavior, although less pronounced, can be observed for the remaining procedures, which find most of their optimal solutions early in the execution; for instance, PBF-D solves close to 90 instances within the first 300 seconds, and only slightly more than 5 additional ones during the remaining time.

Overall, these results illustrate the value of the distance estimation provided by the master problem: decomposition benefits formulations whose masters provide such an estimation, while SBF-D, whose master provides none, performs considerably worse than its monolithic counterpart. This motivates using stronger estimates, such as those from the approximations in Section 4, which we study in the following sections. Moreover, the best-performing procedure in this comparison is the one based on dual cuts, which motivates comparing both types of cuts in the approximated decompositions.

7.3 Exact vs. approximated monolithic formulations

We now compare the exact monolithic formulations (ABF, PBF, and SBF) with their MILP approximations (ABF-A, PBF-A, and SBF-A). Since the approximations provide lower bounds for the original problem, we compute their gap using the MILP lower bound and an upper bound obtained by fixing the visit sequence of the best incumbent solution and solving the resulting SOCP subproblem. For the approximations, an instance solved to optimality means the MILP is solved to optimality, and not necessarily the CETSP; thus, the number of instances solved to optimality is not directly comparable between exact and approximated formulations. Lastly, the exact MISOCP models are solved by Gurobi using its default algorithms, so this comparison measures the value of explicitly constructing the polyhedral approximation, in the spirit of the lifted LP-based algorithms studied by Vielma et al. [30], rather than relying on the solver's internal handling of the conic constraints.

Figure 5 presents scatter plots comparing, for each instance, the runtime (left column) and the gap (right column). Each row corresponds to one formulation, where

Procedure	#Opt	TimeOpt	Gap (%)	Gap (NoOpt) (%)	Root gap (%)	Nodes
ABF	6	106.95	24.91	25.77	100.00	197,738
ABF-A	80	306.10	17.86	31.44	100.00	276,901
PBF	83	306.38	17.19	31.89	51.05	43,710
PBF-A	102	254.23	14.28	31.74	51.71	21,459
SBF	45	178.05	43.78	58.71	81.91	484,881
SBF-A	50	99.31	29.88	41.03	82.06	1,477,916

Table 4: Summary of results for the exact monolithic formulations and their approximations

each (x, y) point in the grid represents an instance: the monolithic formulation (x-axis) and its linear approximation (y-axis). Different marker shapes and colors distinguish the values of n and r for each instance, and the diagonal line is the identity; thus, points below this line represent instances where the approximation is smaller than the exact formulation. Because we cannot compare the three instances on which the solver terminated with an error when solving SBF, we do not show them in the corresponding plots and exclude them from the results of SBF in the following tables. Table 4 summarizes the results, with the same columns as Table 3, plus column "Root gap", which shows the average gap between the optimal value of the continuous relaxation of each model and the best upper bound obtained by any procedure in this work for the same instance.

Regarding runtimes, the approximations are faster than their exact counterparts on almost every instance. This effect is particularly notable for ABF: on many instances, ABF hits the time limit, while ABF-A finishes in a fraction of that time, often within a few seconds. For PBF, the points follow the diagonal more closely, with PBF-A being faster, but by a smaller margin. Lastly, SBF-A is faster than SBF on most instances, with SBF either solving instances quickly or reaching the time limit, as observed in the previous section.

In terms of gap, all three approximations obtain smaller average gaps than their exact counterparts (17.86% against 24.91% for ABF, 14.28% against 17.19% for PBF, and 29.88% against 43.78% for SBF), with most points in Figure 5 lying below the diagonal. Although, as noted earlier, the number of instances solved to optimality is not directly comparable, the approximations achieve smaller gaps in less time, even though their gaps include approximation error, indicating they are considerably easier to solve. This improvement is smallest for PBF, whose gaps and runtimes closely follow those of PBF-A, suggesting that, as its continuous relaxation is already strong, it benefits less from replacing its conic constraints with polyhedral approximations. In contrast, ABF benefits most, while SBF yields the most dispersed results, with a few instances, mainly at $n = 30$, where SBF-A obtains larger gaps than SBF.

The root gaps are consistent with the relationships between the continuous relaxations established in Section 3. The relaxations of ABF and ABF-A are very weak, with average root gaps of 100.00%, likely due to the big-M constraints, while that of PBF is considerably stronger, with an average root gap of 51.05%, in line with Propositions 1 and 2. The relaxation of SBF lies in between, with an average root gap of 81.91%; thus, although the relaxations of SBF and PBF are incomparable in

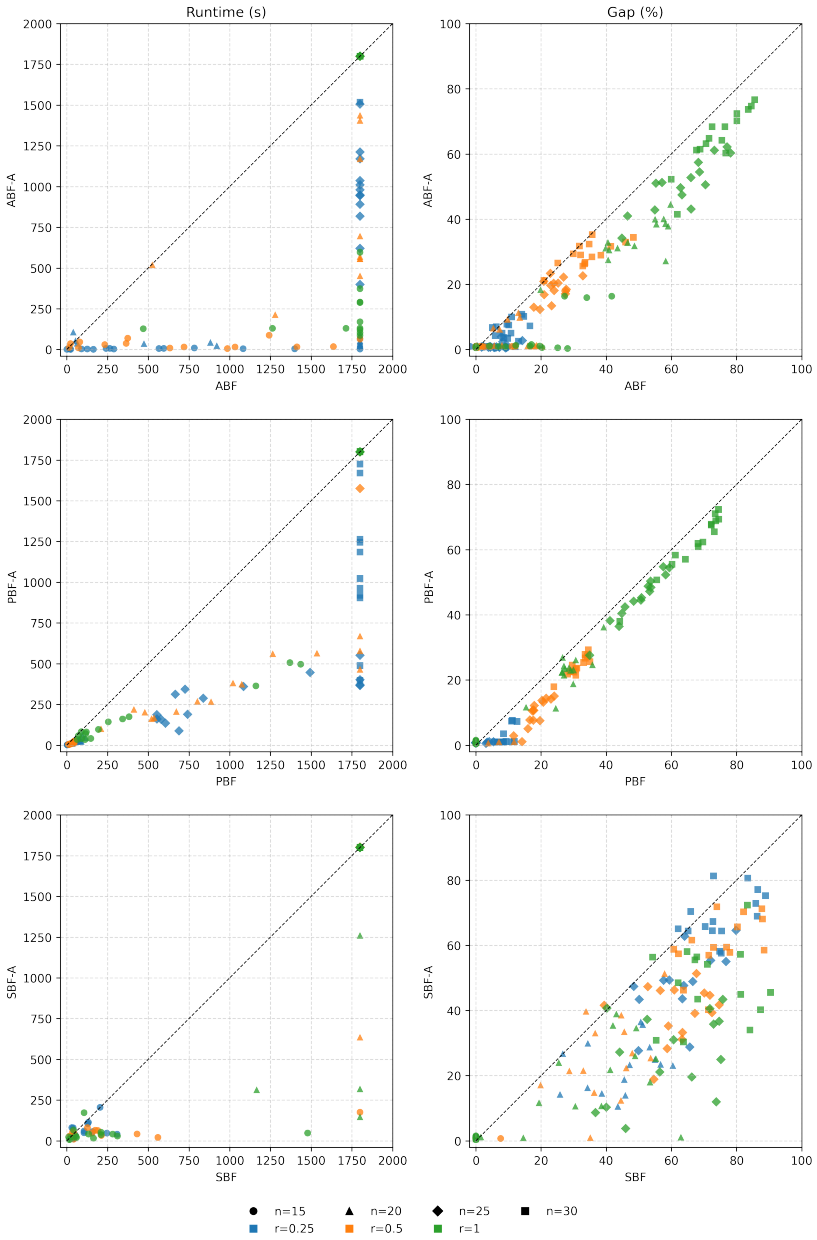


Fig. 5: Runtime and gap comparisons between the exact monolithic formulations and their approximations

general (Proposition 4), that of PBF is considerably stronger on average. Lastly, the approximations slightly weaken the relaxations (from 51.05% to 51.71% for PBF, and from 81.91% to 82.06% for SBF), as expected from outer approximations of the conic constraints, although the difference is small, given the accuracy of the approximation.

For instances where the approximations are solved to optimality, corresponding to points near the horizontal axis in the gap plots of Figure 5, the gap averages below 1% and never exceeds 1.53%. Thus, the approximations provide accurate distance estimations, and their larger gaps on the remaining instances are mostly due to the MILP not being solved to optimality, rather than to the approximation itself. Lastly, for all formulations, the largest gaps are obtained on instances with a large number of locations and large radii, a result which we analyze further in Section 7.5.

These results show that the approximations provide accurate distance estimations at a considerably lower computational cost than the exact formulations, although solving the MILP itself becomes the bottleneck on larger instances. This motivates using them as master problems in the decompositions ABF-A-D, PBF-A-D, and SBF-A-D, which, in addition, removes the error introduced by the approximation, and which we study in the following section.

7.4 Decompositions with and without approximation

This section compares the decompositions of Section 5 with their approximated counterparts to better understand the role of the approximation and the different cut alternatives for each decomposition approach. For PBF-A-D and SBF-A-D, we report both the variant with enumerative cuts (11) and dual cuts ((18) and (22), respectively), denoted by the suffixes "enum" and "dual". The remaining decompositions use a single type of cut, for the reasons discussed in Section 5: dual cuts are not used in the arc-based decompositions, as they would involve the big-M constants (see the remark in Section 5.1); PBF-D with enumerative cuts is identical to ABF-D (see Section 5.2); and enumerative cuts in SBF-D would effectively require enumerating sequences, as its master provides no distance estimation (see Section 5.3). Figure 6 presents the performance profile of the eight procedures, following the same conventions as Figure 4, with the darkest shades corresponding to the approximated decompositions. Table 5 summarizes the results, with the same columns as Table 3, plus column "#Cuts", which shows the average number of cuts generated.

Procedure	#Opt	TimeOpt	Gap (%)	Gap (NoOpt) (%)	#Cuts	Nodes
ABF-D	37	296.01	27.82	35.01	41630.68	336,128
ABF-A-D	61	198.33	22.87	34.59	137.28	277,307
PBF-D	96	118.02	16.12	34.54	1463.11	360,391
PBF-A-D-enum	95	330.53	15.44	32.70	172.41	19,123
PBF-A-D-dual	90	324.73	16.77	33.55	117.04	17,484
SBF-D	0	-	88.29	88.29	5231.48	238,701
SBF-A-D-enum	46	235.01	40.93	54.98	543.13	1,301,516
SBF-A-D-dual	43	344.51	43.90	57.68	427.53	776,271

Table 5: Summary of results for the decompositions and their approximated counterparts

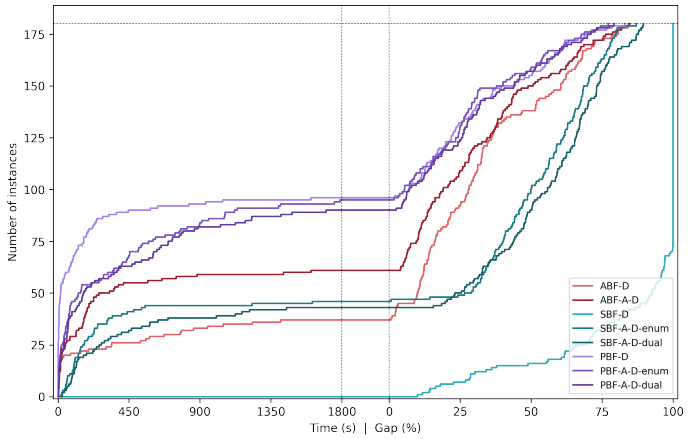


Fig. 6: Performance profile of the decompositions and their approximated counterparts on all instances

We observe that the three decompositions based on PBF are the best-performing procedures, with their lines in the gap section of the profile nearly coinciding, followed by ABF-A-D and then by both variants of SBF-A-D. These procedures also differ considerably in how quickly they find optimal solutions: PBF-D solves close to 90 instances within the first 300 seconds, while the approximated decompositions accumulate optimal solutions more steadily throughout the time limit.

The approximation considerably strengthens the decompositions based on ABF and SBF. ABF-A-D solves 24 more instances to optimality than ABF-D (61 against 37), in less average time and with a smaller average gap (22.87% against 27.82%). For SBF, the effect is even more pronounced: while SBF-D fails to solve any instance to optimality, with an average gap of 88.29%, its approximated counterparts solve 46 and 43 instances to optimality, with average gaps of 40.93% and 43.90%, respectively.

In contrast, the results for PBF are mixed. PBF-A-D with enumerative cuts obtains the smallest average gap of all procedures (15.44% against 16.12% for PBF-D), while solving nearly the same number of instances to optimality (95 against 96). However, PBF-D is considerably faster, with an average time on optimal instances of 118.02 seconds against 330.53, making it preferable overall. This aligns with the results in Section 7.3, where PBF benefited least from the approximation. Recall that the master problem of PBF-D is the same as that of ABF-D, so its performance comes from the dual cuts, which already compensate for its master's weak distance estimation. Thus, replacing it with the considerably larger master given by PBF-A adds computational cost with little benefit. This is also reflected in the number of nodes explored: with similar results, PBF-A-D with enumerative cuts explores around 19 times fewer nodes than PBF-D (19,123 against 360,391 on average), consistent with each node of its master problem being considerably more expensive to solve.

Regarding the number of cuts, we observe that the approximated decompositions generate fewer cuts than their counterparts: from over 41,000 to 137 on average for ABF, from over 5,000 to around 430–540 for SBF, and from close to 1,500 to around

120–170 for PBF. This is expected, since the approximated master problems are much closer to the original problem, and thus fewer tours need to be examined. This also explains why, in both approximated decompositions, enumerative cuts outperform dual cuts, solving more instances to optimality (95 against 90 for PBF-A-D, and 46 against 43 for SBF-A-D) with smaller average gaps. Because only a few tours need to be examined, the greater generality of dual cuts offers limited benefit, while their density (i.e., involving every arc or location–stage pair) is greater than that of enumerative cuts, which involve only the current solution's support. Moreover, in PBF-A-D, computing the off-support coefficients adds further overhead to generate each cut. In both approximated decompositions, the variants with dual cuts explore fewer nodes than those with enumerative cuts (776,271 against 1,301,516 on average for SBF-A-D, and 17,484 against 19,123 for PBF-A-D), which suggests that dense cuts make the master problem harder to solve at each node.

Lastly, when the master problem provides a weak distance estimation, the choice of cuts makes a considerable difference: with identical master problems, PBF-D generates around 28 times fewer cuts than ABF-D (1,463 against 41,631 on average), while solving 59 more instances to optimality. Thus, dual cuts are most effective when the distance estimation of the master is weak, while enumerative cuts perform better when the master problem is already close to the original problem.

7.5 Analysis by instance features

We now analyze how the number of locations and the radii of the neighborhoods affect performance. We do not analyze the results based on the variation among radii σ, as the performance of the procedures does not vary considerably across its values. We consider the best-performing decomposition for each formulation, namely ABF-A-D, SBF-A-D, and PBF-A-D with enumerative cuts, along with PBF-D, since both PBF-based decompositions performed similarly overall. Figures 7 and 8 present the performance profiles of these procedures for each value of n and r, respectively, and Tables 6 and 7 summarize the corresponding results. Each value of n corresponds to 45 instances, each value of r to 60, and each combination of both to 15.

7.5.1 Analysis based on the number of locations

All four procedures perform well on instances with $n = 15$, with PBF-A-D and SBF-A-D solving all 45 instances to optimality, and ABF-A-D and PBF-D solving 42 and 40, respectively. However, ABF-A-D solves no instance to optimality for $n \geq 25$, and SBF-A-D solves only one for $n \geq 20$, with their average gaps reaching 42.43% and 71.84% for $n = 30$. In contrast, both decompositions based on PBF solve the same number of instances to optimality for $n = 20$ and $n = 25$ (30 and 15, respectively), although PBF-D does so considerably faster (e.g., 97.20 seconds on average for $n = 25$, against 525.12), and, for $n = 30$, PBF-D solves more than twice as many instances as PBF-A-D (11 against 5), with similar gaps for every value of n. Thus, PBF-D scales best with the number of locations among the four procedures. As it differs from the remaining ones both in its master problem (i.e., does not include the approximated constraints) and in its cuts, this comparison does not isolate the effect of either. However, one possible

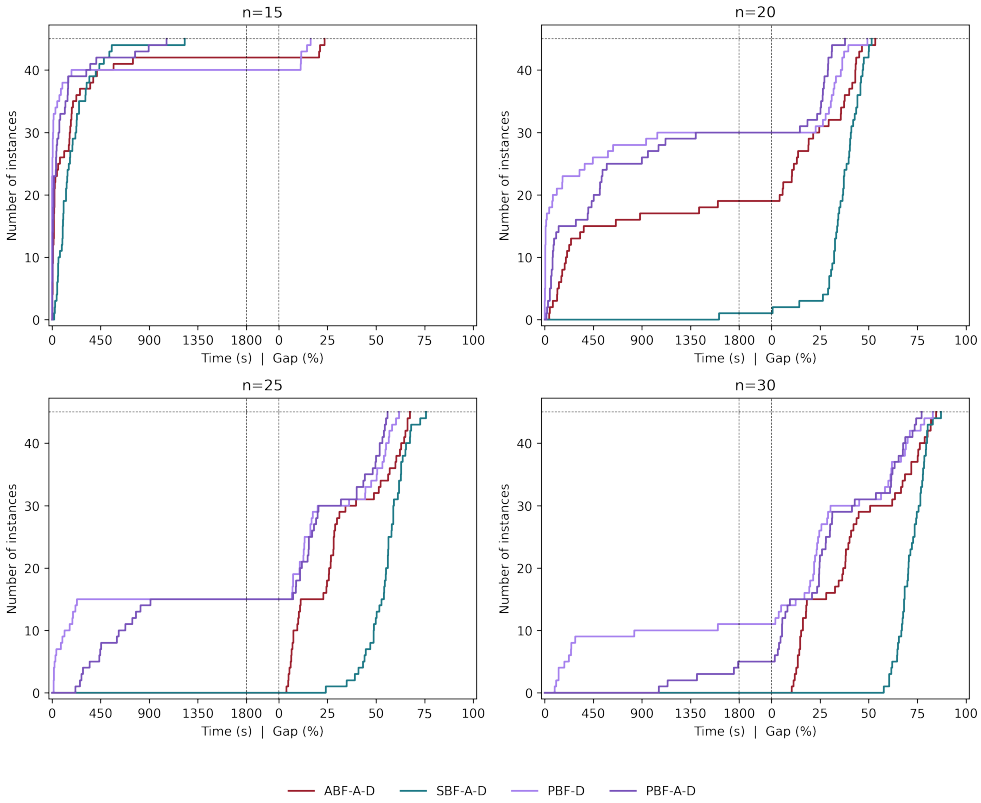


Fig. 7: Performance profiles of the selected decompositions grouped by number of locations

explanation for the poorer scaling of the approximated decompositions is that, beyond their larger master problems, they rely on enumerative cuts that provide information about a single tour, while the number of tours that may need to be examined grows very quickly with n. The number of nodes explored supports the first part of this explanation: while PBF-D explores increasingly more nodes as n increases (from 72,979 on average for $n = 15$ to 557,539 for $n = 30$), the approximated decompositions explore fewer nodes beyond $n = 20$ (e.g., from 29,718 to 11,822 for PBF-A-D), indicating that their master problems become considerably more expensive to solve at each node. Lastly, the average number of cuts generated by ABF-A-D and SBF-A-D decreases as n increases, likely because their master problems become harder to solve, so that fewer integer solutions are found within the time limit.

7.5.2 Analysis based on the radii

In general, the problem becomes more difficult as the radii increase, which is consistent with the use of the overlap ratio as a measure of instance difficulty in the literature [4, 12]. This is particularly evident for the arc-based and perspective-based procedures:

n	Procedure	#Opt	TimeOpt	Gap (%)	Gap (NoOpt) (%)	#Cuts	Nodes
15	ABF-A-D	42	117.59	1.45	21.82	171.93	203,143
15	PBF-D	40	18.64	1.44	12.99	800.58	72,979
15	PBF-A-D	45	119.00	0.00	-	166.56	16,321
15	SBF-A-D	45	204.39	0.00	-	946.16	444,759
20	ABF-A-D	19	376.80	16.32	28.24	137.13	417,329
20	PBF-D	30	168.66	11.10	33.31	1470.93	340,889
20	PBF-A-D	30	367.45	8.91	26.73	154.11	29,718
20	SBF-A-D	1	1612.95	36.43	37.26	690.96	2,195,628
25	ABF-A-D	0	-	31.28	31.28	128.93	306,096
25	PBF-D	15	97.20	21.62	32.42	1984.71	470,155
25	PBF-A-D	15	525.12	20.86	31.30	194.69	18,630
25	SBF-A-D	0	-	55.44	55.44	275.29	1,534,943
30	ABF-A-D	0	-	42.43	42.43	111.11	182,660
30	PBF-D	11	369.65	30.30	40.11	1596.20	557,539
30	PBF-A-D	5	1428.95	31.99	35.99	174.27	11,822
30	SBF-A-D	0	-	71.84	71.84	260.11	1,030,735

Table 6: Summary of results for the selected decompositions grouped by number of locations

r	Procedure	#Opt	TimeOpt	Gap (%)	Gap (NoOpt) (%)	#Cuts	Nodes
0.25	ABF-A-D	30	90.65	5.50	11.01	84.50	143,358
0.25	PBF-D	56	99.96	0.38	5.75	376.70	100,898
0.25	PBF-A-D	50	321.79	0.91	5.43	106.25	7,992
0.25	SBF-A-D	15	236.14	41.77	55.69	355.23	1,358,840
0.50	ABF-A-D	19	295.52	19.08	27.93	102.95	277,988
0.50	PBF-D	30	167.85	9.02	18.03	1101.15	414,247
0.50	PBF-A-D	30	348.66	10.46	20.91	135.78	16,133
0.50	SBF-A-D	15	210.62	42.30	56.40	424.63	1,256,492
1.00	ABF-A-D	12	313.65	44.02	55.03	224.38	410,576
1.00	PBF-D	10	69.62	38.95	46.74	2911.47	566,027
1.00	PBF-A-D	15	323.39	34.97	46.62	275.18	33,244
1.00	SBF-A-D	16	256.81	38.71	52.79	849.52	1,289,217

Table 7: Summary of results for the selected decompositions grouped by radius values

ABF-A-D solves 30, 19, and 12 instances to optimality for $r = 0.25$, 0.5, and 1, respectively, while PBF-D solves 56, 30, and 10, and PBF-A-D 50, 30, and 15, with average gaps increasing accordingly. In fact, for $r = 1$, neither decomposition based on PBF solves any instance with $n > 15$ to optimality.

Notably, the relative performance of PBF-D and PBF-A-D is inverted as the radii increase. For $r = 0.25$, PBF-D solves more instances to optimality (56 vs. 50), in less than a third of the average time; for $r = 0.5$, both solve the same number; and, for $r = 1$, PBF-A-D solves more (15 vs. 10), with a smaller average gap (34.97% against 38.95%). In fact, the five instances with $n = 15$ that PBF-D fails to solve are precisely those with $r = 1$. This behavior can be explained by the underestimators m_{ij} used

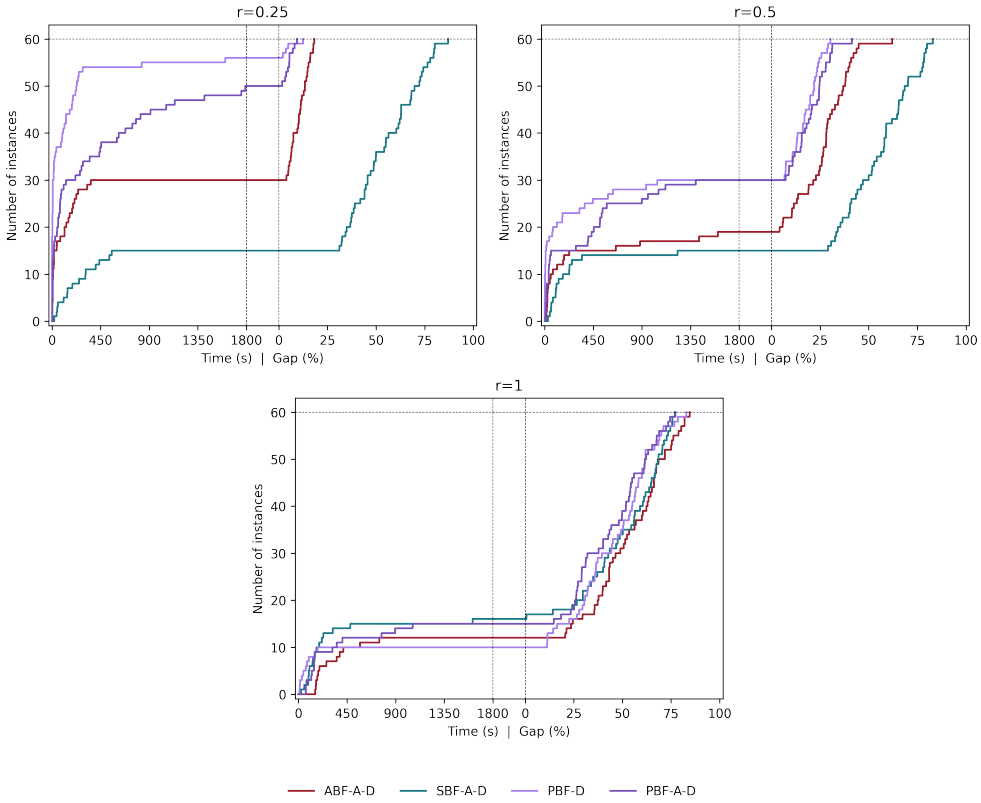


Fig. 8: Performance profiles of the selected decompositions grouped by radius values

in the master problem of PBF-D, which are reasonable when the neighborhoods are small and far apart, but are zero for overlapping neighborhoods. As the radii increase, the master problem thus provides an increasingly poor estimation, and the cuts must provide most of the information, as reflected by the average number of cuts generated by PBF-D (i.e., from 376.70 for $r = 0.25$ to 2,911.47 for $r = 1$) and branch-and-bound nodes (i.e., from 100,898 to 566,027). In contrast, the master problem of PBF-A-D retains an accurate estimation regardless of the radii. Nevertheless, the number of cuts also grows with the radii for the decompositions with enumerative cuts, roughly doubling from $r = 0.5$ to $r = 1$. One possible explanation is that, as neighborhoods overlap, more tours share the same optimal length, and because enumerative cuts improve the master only at the tour that generates them, each of these tours may require its own cuts without improving the lower bound.

In contrast, the number of instances solved to optimality by SBF-A-D does not decrease with the radii (15, 15, and 16, respectively). However, this is due to SBF-A-D solving all instances with $n = 15$, for every value of r, but almost none of the larger ones, with average gaps between 38.71% and 42.30% across all radii. The number of nodes is also nearly constant across radii, averaging 1.26 to 1.36 million. We conjecture

that the absence of degradation is related to the structure of its master problem: in its continuous relaxation, each stage can be fractionally assigned to several locations, so that the neighborhood of each stage becomes an average of several neighborhoods, similarly to the averaging that occurs in the relaxation of SBF (see Proposition 3). Thus, its relaxation is loose regardless of the radii, while the relaxations of arc-based formulations rely on pairwise distances between neighborhoods, which are informative for small radii but lose information as the neighborhoods overlap.

Lastly, PBF-A-D solves more instances to optimality than ABF-A-D for every value of r, mirroring the relationship observed between ABF and PBF in Section 7.2. Also, the differences between procedures decrease considerably as the radii increase: for $r = 0.25$, the average gaps range from below 1% for both decompositions based on PBF to 41.77% for SBF-A-D, while, for $r = 1$, they range only between 34.97% and 44.02%.

Overall, these results show that no single procedure performs best across all instances. PBF-D scales best with the number of locations, solving the most instances to optimality on the largest instances in considerably less time, while PBF-A-D is more robust to large radii on small instances. Lastly, ABF-A-D and SBF-A-D are only competitive on the smallest instances.

7.6 Comparison with Behdani and Smith

We now compare our best-performing procedure, PBF-D, with our reimplementation of the iterative procedure of Behdani and Smith [2] (BS). Among the three formulations proposed by the authors, we consider the one that performed best in their experiments (LB3 in their notation), solved via the Benders decomposition they propose. Following their experiments, we initially divide the boundary of each neighborhood within the convex hull of the locations into four arcs, and at each iteration we bisect each arc visited by an optimal solution, for a total of five iterations as in [2]. Both procedures are run under the same conditions and time limit. Note that BS provides lower and upper bounds on the optimal tour length, but completing its iterations does not certify optimality.

Table 8 summarizes the results by number of locations, and Figure 9 shows scatter plots that compare both procedures instance by instance, in terms of runtime (left) and gap (right), following the same conventions as Figure 5. Because completing BS iterations and solving an instance to optimality are different events, Table 8 reports them in separate columns, "#Completed" and "#Opt", respectively. Accordingly, column "Time" reports the average time on instances where BS completes its iterations and the average time on instances solved to optimality by PBF-D, while "Gap" reports the average gap across all instances.

We observe that PBF-D dominates BS across all instances. Its final gap is never larger than that of BS on any of the 180 instances, and it is faster on every instance where both procedures terminate, that is, on all 30 such instances with $n = 15$ and all 15 with $n = 20$. Its average times are also smaller (18.64 seconds vs. 90.86 for $n = 15$, and 168.66 vs. 225.80 for $n = 20$), even though they are computed over more instances, including those that BS does not complete. Furthermore, for $n \geq 25$, BS does not complete its iterations on any instance, whereas PBF-D solves 15 and 11

	BS			PBF-D		
n	#Completed	Time	Gap (%)	#Opt	Time	Gap (%)
15	30	90.86	10.75	40	18.64	1.44
20	15	225.80	24.10	30	168.66	11.10
25	0	-	35.31	15	97.20	21.62
30	0	-	41.00	11	369.65	30.30

Table 8: Summary of results for BS and PBF-D by n

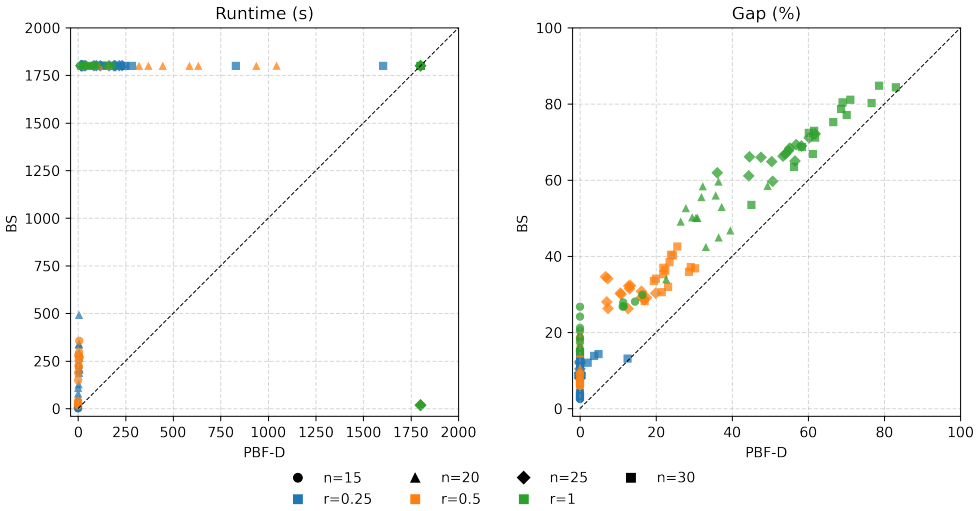


Fig. 9: Runtime and gap comparisons between PBF-D and BS

instances to optimality. Moreover, even on the 45 instances where BS completes its five iterations, it leaves an average gap of 4.99%, ranging from 2.53% to 9.20%, due to the discretization. More generally, on instances that PBF-D solves to optimality, corresponding to points with a PBF-D gap of zero in the gap plot of Figure 9, BS leaves an average gap of 9.89%, up to 26.73%.

Both procedures obtain their largest gaps on instances with $r = 1$, which is consistent with the analysis of Section 7.5, and with the observation of Behdani and Smith [2] that a larger overlap ratio impairs the performance of their procedure.

Lastly, we note that the solver reported a suboptimal status for the BS master problem in five instances with $n = 25$, terminating in roughly 18 seconds. These instances appear in the bottom-right corner of the runtime plot of Figure 9, reflecting early termination of the refinement procedure rather than a solved instance.

8 Conclusions

This work addresses the Close-Enough Traveling Salesman Problem (CETSP). We present three exact MISOCP formulations for this problem and study their continuous relaxations to uncover dominance among them. Specifically, we consider an arc-based formulation using big-M constraints, a perspective-based formulation that avoids them, and a sequence-based formulation. We also present approximated mixed-integer linear formulations derived from polyhedral approximations of the second-order cone, and decomposition schemes for both exact and approximated formulations, using enumerative and dual cuts to iteratively refine the master problem. Furthermore, we show that the cuts originally proposed by Gao et al. [9] are not valid.

Our computational experiments show that the decompositions based on PBF achieve the best overall performance, and that no single procedure dominates across all instances. Specifically, PBF-D scales best with the number of locations, while PBF-A-D is more robust on instances with large neighborhoods, where the distance underestimators of PBF-D become uninformative. The experiments also illustrate the complementary roles of the two types of cuts: dual cuts are most effective when the master problem provides a weak distance estimation, whereas enumerative cuts perform better when the master is already close to the original problem. Finally, our best-performing decomposition outperforms the discretization scheme of Behdani and Smith [2] on every instance tested.

Future research could explore the development of stronger cuts for the decomposition schemes, particularly cuts that combine the accurate distance estimation of the approximated master problems with the generality of dual cuts, to improve the scalability of the approximated decompositions. Additionally, the formulations and decomposition schemes presented could be extended to address related problems, such as the Prize-Collecting TSP with neighborhoods or the Vehicle Routing Problem with neighborhoods. Finally, the combination of polyhedral approximations of the second-order cone and a decomposition scheme could be adapted to solve other optimization problems involving conic constraints and binary variables, potentially leading to new insights and solution methods in this area.

Acknowledgements. We sincerely thank the referees and editorial team for their insightful comments and suggestions, which helped us considerably improve our manuscript and better highlight our contributions. Specifically, we thank Referee 1 for suggesting that we consider a perspective formulation for this problem, which ultimately became the foundation of our best-performing methodology.

References

[1] Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton (2007). https://doi.org/10.1515/9781400841103

[2] Behdani, B., Smith, J.C.: An integer-programming-based approach to the close-enough traveling salesman problem. INFORMS Journal on Computing **26**(3), 415–432 (2014) https://doi.org/10.1287/ijoc.2013.0574

[3] Coutinho, W.P., Nascimento, R.Q.d., Pessoa, A.A., Subramanian, A.: A branch-and-bound algorithm for the close-enough traveling salesman problem. INFORMS Journal on Computing **28**(4), 752–765 (2016) https://doi.org/10.1287/ijoc.2016.0711

[4] Di Placido, A., Archetti, C., Cerrone, C.: A genetic algorithm for the close-enough traveling salesman problem with application to solar panels diagnostic reconnaissance. Computers & Operations Research **145**, 105831 (2022) https://doi.org/10.1016/j.cor.2022.105831

[5] Yuan, B., Orlowska, M., Sadiq, S.: On the optimal robot routing problem in wireless sensor networks. IEEE Trans. Knowl. Data Eng. **19**, 1252–1261 (2007) https://doi.org/10.1109/TKDE.2007.1062

[6] Di Placido, A., Archetti, C., Cerrone, C., Golden, B.: The generalized close enough traveling salesman problem. European Journal of Operational Research **310**(3), 974–991 (2023) https://doi.org/10.1016/j.ejor.2023.04.010

[7] Carrabs, F., Cerulli, R., D'Ambrosio, C., Murano, G.: Upper bound computation for the multiple close-enough traveling salesman problem, pp. 186–195 (2025). https://doi.org/10.5220/0013377900003893

[8] Cui, W., Zhu, J., Peng, L.: On the service time in the close-enough traveling salesman problem. European Journal of Operational Research (2026)

[9] Gao, C., Wei, N., Walteros, J.L.: An exact approach for solving pickup-and-delivery traveling salesman problems with neighborhoods. Transportation Science **57**(6), 1560–1580 (2023) https://doi.org/10.1287/trsc.2022.0138

[10] Geoffrion, A.M.: Generalized Benders decomposition. Journal of optimization theory and applications **10**, 237–260 (1972)

[11] Gao, C., Wei, N., Walteros, J.L.: Erratum to "An Exact Approach for Solving Pickup-and-Delivery Traveling Salesman Problems with Neighborhoods". Optimization Online, updated version of the preprint. Available at: https://optimization-online.org/2022/04/8891/. Last visited: October 2026 (2026)

[12] Mennell, W.K.: Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem. PhD thesis, The Robert H. Smith School of Business, University of Maryland (2009)

[13] Lei, Z., Hao, J.-K.: An effective memetic algorithm for the close-enough traveling salesman problem. Applied Soft Computing **153**, 111266 (2024) https://doi.org/10.1016/j.asoc.2024.111266

[14] Canturk, D., Dökeroğlu, T.: Scalable parallel memetic algorithms for large close-enough traveling salesman problems. Neural Computing and Applications **38**(2) (2026) https://doi.org/10.1007/s00521-025-11706-4

[15] Duong, K.: A Fast Quasi-Linear Heuristic for the Close-Enough Traveling Salesman Problem (2026). https://arxiv.org/abs/2603.21401

[16] Gentilini, I., Margot, F., Shimada, K.: The travelling salesman problem with neighbourhoods: MINLP solution. Optimization Methods and Software **28**(2), 364–378 (2013)

[17] Carrabs, F., Cerrone, C., Cerulli, R., Gaudioso, M.: A novel discretization scheme for the close enough traveling salesman problem. Computers & Operations Research **78**, 163–171 (2017) https://doi.org/10.1016/j.cor.2016.09.003

[18] Zhang, W., Sauppe, J., Jacobson, S.: Results for the close-enough traveling salesman problem with a branch-and-bound algorithm. Computational Optimization and Applications **85**, 1–39 (2023) https://doi.org/10.1007/s10589-023-00474-3

[19] Gutow, G., Choset, H.: Efficient Second-Order Cone Programming for the Close Enough Traveling Salesman Problem. (2026)

[20] Deckerová, J., Váňa, P., Faigl, J.: Combinatorial lower bounds for the generalized traveling salesman problem with neighborhoods. Expert Systems with Applications **258**, 125185 (2024) https://doi.org/10.1016/j.eswa.2024.125185

[21] Fekete, S.P., Kniep, R., Krupke, D., Perk, M.: A Branch-And-Bound Algorithm for the Traveling Salesman Problem with Difficult Neighborhoods. In: Ahn, H.-K., Hoffmann, M., Nayyeri, A. (eds.) 42nd International Symposium on Computational Geometry (SoCG 2026). Leibniz International Proceedings in Informatics (LIPIcs), vol. 367, pp. 46–14620. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2026). https://doi.org/10.4230/LIPIcs.SoCG.2026.46

[22] 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)

[23] Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. Journal of the ACM **7**(4), 326–329 (1960)

[24] Camm, J.D., Raturi, A.S., Tsubakitani, S.: Cutting Big M down to size. Interfaces **20**(5), 61–66 (1990) https://doi.org/10.1287/inte.20.5.61

[25] Günlük, O., Linderoth, J.: Perspective reformulation and applications. In: Lee, J., Leyffer, S. (eds.) Mixed Integer Nonlinear Programming, pp. 61–89. Springer, New York, NY (2012)

[26] Marcucci, T., Umenberger, J., Parrilo, P., Tedrake, R.: Shortest paths in graphs of convex sets. SIAM Journal on Optimization **34**(1), 507–532 (2024) https://doi.org/10.1137/22M1523790

[27] Marcucci, T.: A Unified and Scalable Method for Optimization over Graphs of Convex Sets (2025). https://arxiv.org/abs/2510.20184

[28] Margot, F.: Symmetry in integer linear programming. In: Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A. (eds.) 50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art, pp. 647–686. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-540-68279-0_17

[29] Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. Mathematics of Operations Research **26**(2), 193–205 (2001)

[30] Vielma, J.P., Dunning, I., Huchette, J., Lubin, M.: Extended formulations in mixed integer conic quadratic programming. Mathematical Programming Computation **9**(3), 369–418 (2017) https://doi.org/10.1007/s12532-016-0113-y

[31] Vielma, J.P., Ahmed, S., Nemhauser, G.L.: A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. INFORMS Journal on Computing **20**(3), 438–450 (2008) https://doi.org/10.1287/ijoc.1070.0256

[32] Laporte, G., Louveaux, F.V.: The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters **13**(3), 133–142 (1993) https://doi.org/10.1016/0167-6377(93)90002-X

[33] Weiszfeld, E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Mathematical Journal, First Series **43**, 355–386 (1937)

[34] Torres, G.A.: A Weiszfeld-like algorithm for a Weber location problem constrained to a closed and convex set (2013). https://arxiv.org/abs/1204.1087

[35] Beck, A., Sabach, S.: Weiszfeld's method: Old and new results. Journal of Optimization Theory and Applications **164**(1), 1–40 (2015)

[36] Neumayer, S., Nimmer, M., Steidl, G., Stephani, H.: On a projected Weiszfeld algorithm. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 486–497 (2017). Springer

[37] Borwein, J.J., Lewis, A.: Convex Analysis and Nonlinear Optimization: Theory and Examples, (2005). https://doi.org/10.1007/978-0-387-31256-9

[38] Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4**(1), 238–252 (1962) https://doi.org/10.1007/BF01386316

[39] Costa, A.M.: A survey on Benders decomposition applied to fixed-charge network design problems. Computers & Operations Research **32**(6), 1429–1450 (2005) https://doi.org/10.1016/j.cor.2003.11.012

[40] Bagajewicz, M.J., Manousiouthakis, V.: On the Generalized Benders Decomposition. Computers & Chemical Engineering **15**(10), 691–700 (1991) https://doi.org/10.1016/0098-1354(91)85015-M

[41] Gavish, B., Graves, S.C.: The Travelling Salesman Problem and Related Problems (1978). http://hdl.handle.net/1721.1/5363

Appendix A Proofs

Proposition 1. *Suppose ABF and PBF use the same constraint set $\mathcal{X}$. Then the projection of the continuous relaxation of PBF onto the (x, d)-space is contained in that of ABF. In particular, the optimal value of the continuous relaxation of PBF is at least that of ABF.*

Proof. Let (x, p, d) be a feasible solution to the continuous relaxation of PBF, where p collects both the p_i^{ij} copies and the p_j^{ij} ones. We will build a feasible solution for the continuous relaxation of ABF keeping x and d unchanged (along with auxiliary variables of $\mathcal{X}$ if there are any). To do so, we first define for all $i \in N$

$$\hat{p}_i := \sum_{j:(i,j)\in A} p_i^{ij} = \sum_{k:(k,i)\in A} p_i^{ki},$$

which is well defined by (3e). We will now prove the constructed $\hat{p}$ satisfies (2b) and (2c) of ABF. Note that, by the degree constraints, we have that

$$\hat{p}_i - c_i = \sum_{j:(i,j)\in A} \left(p_i^{ij} - x_{ij} c_i\right) \qquad \forall i \in N.$$

Then, we have

$$\|\hat{p}_i - c_i\| = \left\| \sum_{j:(i,j)\in A} \left(p_i^{ij} - x_{ij} c_i\right) \right\| \leq \sum_{j:(i,j)\in A} \left\| p_i^{ij} - x_{ij} c_i \right\| \leq \sum_{j:(i,j)\in A} x_{ij} r_i = r_i,$$

where the first inequality comes from the triangle inequality, and the second one from constraints (3b). Thus, $\hat{p}$ satisfies neighborhood constraints (2b). To prove that $\hat{p}$ satisfies the distance constraints (2c), note that we can write

$$\begin{aligned}\hat{p}_i - \hat{p}_j &= \sum_{l:(i,l)\in A} p_i^{il} - \sum_{k:(k,j)\in A} p_j^{kj} \\ &= \sum_{l:(i,l)\in A} \left(p_i^{il} - x_{il} c_i\right) - \sum_{k:(k,j)\in A} \left(p_j^{kj} - x_{kj} c_j\right) + (c_i - c_j),\end{aligned}$$

where the first equality follows from how we defined $\hat{p}$, using continuity constraints (3e), and the second equality follows from the degree constraints in $\mathcal{X}$. We can then pull out from the sums the terms belonging to arc (i, j), namely $l = j$ and $k = i$, which together contribute $(p_i^{ij} - p_j^{ij}) - x_{ij}(c_i - c_j)$. Then, we have

$$\hat{p}_i - \hat{p}_j = (p_i^{ij} - p_j^{ij}) + (1 - x_{ij})(c_i - c_j) + \sum_{\substack{l:(i,l)\in A \\ l\neq j}} \left(p_i^{il} - x_{il} c_i\right) - \sum_{\substack{k:(k,j)\in A \\ k\neq i}} \left(p_j^{kj} - x_{kj} c_j\right).$$

Then, using the triangle inequality, we can bound $\|\hat{p}_i - \hat{p}_j\|$ by the sum of the norms of the terms. Additionally, the first term is bounded by (3d), while the terms in the sums are bounded by (3b) and (3c). Thus, we obtain

$$\|\hat{p}_i - \hat{p}_j\| \leq d_{ij} + (1 - x_{ij})\|c_i - c_j\| + \sum_{\substack{l:(i,l)\in A \\ l\neq j}} x_{il} r_i + \sum_{\substack{k:(k,j)\in A \\ k\neq i}} x_{kj} r_j.$$

From the degree constraints, we know that

$$\sum_{\substack{l:(i,l)\in A \\ l\neq j}} x_{il} = \sum_{\substack{k:(k,j)\in A \\ k\neq i}} x_{kj} = 1 - x_{ij}.$$

Substituting in the previous inequality, we obtain

$$\|\hat{p}_i - \hat{p}_j\| \leq d_{ij} + (1 - x_{ij})\left(r_i + \|c_i - c_j\| + r_j\right) = d_{ij} + (1 - x_{ij})M_{ij},$$

so we conclude that the constructed solution is feasible for the continuous relaxation of ABF. Since x and d are unchanged, (x, d) belongs to the projection of the continuous relaxation of ABF, thus proving the containment. The bound on the optimal values follows directly.

□

Proposition 2. *There exist instances for which, for the same $\mathcal{X}$, the optimal value of the continuous relaxation of PBF is strictly larger than that of ABF.*

Proof. Consider an instance with depot $c_1 = (0, 0)$ and $r_1 = 0$, and three unit disks ($r_j = 1$ for $j \in \{2, 3, 4\}$) centered at $c_2 = (1.5, 0)$, $c_3 = (-1.5, 0)$, and $c_4 = (0, 1.5)$. We will show that the optimal continuous value of ABF is strictly smaller than that of PBF.

Consider a continuous ABF solution with $x_{ij} = 1/3$ for all arcs. Since this is the average of the incidence vectors of the six tours existing for a four-location instance, it belongs to the continuous relaxation of any valid formulation of $\mathcal{X}$. Take $p_i = c_i$ for all i, and $d = 0$. The distance/big-M constraints (2c) then read

$$\|c_i - c_j\| \leq \frac{2}{3} M_{ij} = \frac{2}{3}\left(r_i + \|c_i - c_j\| + r_j\right),$$

which is equivalent to $\|c_i - c_j\| \leq 2(r_i + r_j)$. This inequality holds for every arc. Specifically, the inequality is equal to $1.5 \leq 2$ for arcs incident to the depot, $3 \leq 4$ for arcs between 2 and 3, and $1.5\sqrt{2} \leq 4$ for the remaining ones. Thus, the point is feasible with objective value 0, and, since $d \geq 0$, it must be optimal.

Then, consider any feasible point for the relaxation of PBF. Since $r_1 = 0$, constraint (3b) forces $p_1^{1j} = 0$. Then, the distance constraint for arc $(1, j)$ becomes $\|p_j^{1j}\| \leq d_{1j}$. From the reverse triangle inequality, we have

$$\|p_j^{1j}\| \geq x_{1j}\|c_j\| - \|p_j^{1j} - x_{1j} c_j\|,$$

which, when combined with the previous constraint and with constraint (3c), gives

$$d_{1j} \geq x_{1j}(\|c_j\| - r_j) = 0.5x_{1j},$$

since $\|c_j\| = 1.5$ and $r_j = 1$ for $j \in \{2, 3, 4\}$. Summing over j with the depot's degree constraint gives an objective value of at least 0.5.

Strict containment of the projections follows directly.

□

Proposition 3. *The optimal value of the continuous relaxation of SBF is*

$$V^{SBF} = 2\max\{0, \|c_1 - \bar{c}\| - \bar{r}\}.$$

Proof. For each $k \in N$, let $c_k(z) = \sum_{i \in N} z_{ik} c_i$ and $r_k(z) = \sum_{i \in N} z_{ik} r_i$ denote the center and radius of the disk associated with stage k. Since $r_1 = 0$, by the assignment constraints and $z_{11} = 1$, we have $q_1 = c_1$ and

$$\frac{1}{n-1}\sum_{k=2}^{n} c_k(z) = \bar{c}, \qquad \frac{1}{n-1}\sum_{k=2}^{n} r_k(z) = \bar{r}.$$

For the upper bound, consider $z_{ik} = 1/(n-1)$ for all $i, k \neq 1$. Every stage $k \geq 2$ then has center $\bar{c}$ and radius $\bar{r}$. Taking $q_2 = \cdots = q_n$ as the point in this disk nearest to c_1 yields a feasible solution of value $2\max\{0, \|c_1 - \bar{c}\| - \bar{r}\}$.

For the lower bound, consider any feasible (z, q, t) and define

$$\bar{q} = \frac{1}{n-1}\sum_{k=2}^{n} q_k.$$

Since $\|q_k - c_k(z)\| \leq r_k(z)$, the triangle inequality gives

$$\|\bar{q} - \bar{c}\| \leq \frac{1}{n-1}\sum_{k=2}^{n} \|q_k - c_k(z)\| \leq \bar{r}.$$

Thus, $\bar{q}$ belongs to the disk centered at $\bar{c}$ with radius $\bar{r}$.

By (4d), the objective is at least the perimeter of the closed polygon $q_1, \ldots, q_n$. For every $k \geq 2$, this perimeter contains two paths from $q_1 = c_1$ to q_k, each of length at least $\|q_k - c_1\|$. Therefore,

$$\sum_{k \in N} t_k \geq 2\max_{k \geq 2} \|q_k - c_1\| \geq 2\|\bar{q} - c_1\| \geq 2\max\{0, \|c_1 - \bar{c}\| - \bar{r}\}.$$

This matches the upper bound and proves the result. □

Proposition 4. *For any formulation of $\mathcal{X}$, the continuous relaxation of SBF is incomparable with those of ABF and PBF.*

Proof. We will show two examples: one in which $V^{SBF} > V^{PBF}$ and hence $V^{SBF} > V^{ABF}$, and one in which $V^{ABF} > V^{SBF}$, and hence $V^{PBF} > V^{SBF}$.

For the first case, consider $c_1 = (0, 0)$, $c_2 = (-3, 4)$, $c_3 = (3, 4)$, and $c_4 = (0, 9)$, with $r_2 = r_3 = r_4 = 5$.

For SBF, $\bar{c} = (0, 17/3)$ and $\bar{r} = 5$, so, by Proposition 3, $V^{SBF} = 2(17/3 - 5) = 4/3$.

For PBF, consider the tour $1 \to 2 \to 4 \to 3 \to 1$. Note that this tour only moves between neighborhoods that intersect, as the depot lies on the boundary of disks 2 and 3, since $\|c_2\| = \|c_3\| = 5$, and disk 4 intersects disks 2 and 3, since $\|c_2 - c_4\| = \|c_3 - c_4\| = \sqrt{34} \leq 10$.

Let x^1 be the incidence vector of this tour, and x^2 the one of the reverse tour, and consider $x = (x^1 + x^2)/2$. As x is a convex combination of tours, it is feasible for the relaxation of any formulation of $\mathcal{X}$.

For each edge of the tour, let w_{ij} be a point that lies in both neighborhoods, with $w_{ij} = w_{ji}$. For every arc (i, j) with $x_{ij} = \frac{1}{2}$, we set $p_i^{ij} = p_j^{ij} = \frac{1}{2} w_{ij}$. For every unused arc, we set its point copies to 0. Lastly, set $d = 0$. We will verify that this solution is feasible (and therefore optimal due to the $d \geq 0$ constraint).

On used arcs, constraints (3b) become $\frac{1}{2}\|w_{ij} - c_i\| \leq \frac{1}{2} r_i$, which are satisfied as w_{ij} lies in disk i. The same argument can be made for (3c). The constraints of unused arcs are satisfied trivially.

For distance constraints (3d), both copies on each arc coincide, so the constraint is again trivially satisfied. Lastly, for constraints (3e), note that each location i has two neighbors j and k in the tour. Its incoming copies and its outgoing copies both sum to $\frac{1}{2} w_{ji} + \frac{1}{2} w_{ki}$.

Then, the point is feasible with objective value 0, and thus optimal, so we have $V^{PBF} = 0$, and, by Proposition 1, $V^{ABF} = 0$ as well.

For the second case, consider an instance with $c_1 = (0, 0)$, $c_2 = (-4, 0)$, and $c_3 = (4, 0)$, with unit radii for non-depot locations.

We have that $\bar{c} = c_1$, so by Proposition 3, $V^{SBF} = 0$.

Then, consider any feasible solution to the continuous relaxation of ABF. $r_1 = 0$ forces $p_1 = c_1$, and, for $j = 2, 3$, we have $\|p_1 - p_j\| \geq 3$, since p_j lies in the unit disk around c_j and $\|c_j\| = 4$, and $M_{1j} = 5$, so (2c) gives

$$d_{1j} \geq 3 - 5(1 - x_{1j}) = 5x_{1j} - 2.$$

Summing over j gives

$$d_{12} + d_{13} \geq 5(x_{12} + x_{13}) - 4,$$

and, by the degree constraints, this gives $d_{12} + d_{13} \geq 1$, which, along with $d \geq 0$, guarantees that the objective is at least 1. Since this holds for every feasible point and only degree constraints were used from $\mathcal{X}$, $V^{ABF} \geq 1$ for any formulation of $\mathcal{X}$, and, by Proposition 1, $V^{PBF} \geq 1$. □

Appendix B Coefficient optimization for (15)

For an off-support arc (i, j), we drop the indices and write

$$h(\gamma) = c^\top \gamma + r_1\|\gamma - \eta_1\| + r_2\|\gamma - \eta_2\|,$$

with $c = c_j - c_i$, $\eta_1 = \hat{\eta}_i$, $\eta_2 = \hat{\eta}_j$, $r_1 = r_i$ and $r_2 = r_j$. We seek to decrease h over the unit ball B. We assume $r_1 + r_2 > 0$, as otherwise h is linear and its minimizer over B is $-c/\|c\|$.

For an iterate $\gamma^k \in B$, with $\gamma^k \notin \{\eta_1, \eta_2\}$, consider the following update:

$$\gamma^{k+1} = \Pi_B \left(\frac{r_1 \frac{\eta_1}{\|\gamma^k - \eta_1\|} + r_2 \frac{\eta_2}{\|\gamma^k - \eta_2\|} - c}{r_1 \frac{1}{\|\gamma^k - \eta_1\|} + r_2 \frac{1}{\|\gamma^k - \eta_2\|}} \right), \tag{B1}$$

where Π_B is the projection onto the unit ball.

Proposition 5. *Update (B1) satisfies $h(\gamma^{k+1}) \leq h(\gamma^k)$, with equality if and only if γ^k minimizes h over B.*

Proof. We begin by noting that, for any $a \geq 0, b > 0$, as $(a - b)^2 \geq 0$, it holds that $a \leq \frac{a^2+b^2}{2b}$, with equality if and only if $a = b$. For each $t \in \{1, 2\}$, take $a = \|\gamma - \eta_t\|$ and $b = \|\gamma^k - \eta_t\|$, which is positive by assumption. Additionally, define $w_t = r_t/\|\gamma^k - \eta_t\|$. Then, multiplying by r_t, we have

$$r_t\|\gamma - \eta_t\| \leq \frac{w_t}{2} \left(\|\gamma - \eta_t\|^2 + \|\gamma^k - \eta_t\|^2 \right), \quad \forall \gamma \in \mathbb{R}^2.$$

Summing over $t \in \{1, 2\}$ and adding $c^\top \gamma$ gives

$$h(\gamma) \leq M^k(\gamma) := c^\top \gamma + \sum_{t \in \{1,2\}} \frac{w_t}{2} \left(\|\gamma - \eta_t\|^2 + \|\gamma^k - \eta_t\|^2 \right), \quad \forall \gamma \in \mathbb{R}^2. \tag{B2}$$

At $\gamma = \gamma^k$, each inequality holds with equality, so $h(\gamma^k) = M^k(\gamma^k)$.

Now, we write $W = w_1 + w_2$, $v = w_1\eta_1 + w_2\eta_2 - c$, and $\xi = v/W$, so ξ is the argument of Π_B in (B1). Since $r_1 + r_2 > 0$, we have $W > 0$. Expanding squares in M^k, we get

$$M^k(\gamma) = c^\top \gamma + \sum_{t \in \{1,2\}} \frac{w_t}{2} \left(\|\gamma\|^2 - 2\eta_t^\top \gamma + \|\eta_t\|^2 + \|\gamma^k - \eta_t\|^2 \right).$$

Grouping the terms by their dependence on γ,

$$M^k(\gamma) = \frac{W}{2}\|\gamma\|^2 - v^\top \gamma + K, \qquad K = \sum_{t \in \{1,2\}} \frac{w_t}{2} \left(\|\eta_t\|^2 + \|\gamma^k - \eta_t\|^2 \right),$$

where K does not depend on γ. Since $W\xi = v$, completing the square gives

$$\frac{W}{2}\|\gamma - \xi\|^2 = \frac{W}{2}\|\gamma\|^2 - W\xi^\top\gamma + \frac{W}{2}\|\xi\|^2 = \frac{W}{2}\|\gamma\|^2 - v^\top\gamma + \frac{\|v\|^2}{2W},$$

so

$$\frac{W}{2}\|\gamma\|^2 - v^\top\gamma = \frac{W}{2}\|\gamma - \xi\|^2 - \frac{\|v\|^2}{2W},$$

and, replacing, we obtain

$$M^k(\gamma) = \frac{W}{2}\|\gamma - \xi\|^2 - \frac{\|v\|^2}{2W} + K,$$

which only depends on γ through the first term. As $W > 0$, minimizing M^k over B is equivalent to minimizing $\|\gamma - \xi\|^2$ over B. By definition of the projection, the unique solution is $\Pi_B(\xi) = \gamma^{k+1}$. In particular, this implies $M^k(\gamma^{k+1}) \leq M^k(\gamma^k)$, as $\gamma^k \in B$. Together with (B2), this gives

$$h(\gamma^{k+1}) \leq M^k(\gamma^{k+1}) \leq M^k(\gamma^k) = h(\gamma^k). \tag{B3}$$

It remains to characterize when equality holds. If γ^k minimizes h over B, then $h(\gamma^{k+1}) \geq h(\gamma^k)$, as $\gamma^{k+1} \in B$. Combining this with (B3), equality holds. Conversely, suppose $h(\gamma^{k+1}) = h(\gamma^k)$. Then, every inequality in the chain (B3) holds with equality. In particular, $M^k(\gamma^{k+1}) = M^k(\gamma^k)$. Since γ^{k+1} is the unique minimizer of M^k over B and $\gamma^k \in B$, this forces $\gamma^k = \gamma^{k+1} = \Pi_B(\xi)$. By the characterization of the projection, this means

$$(\xi - \gamma^k)^\top(\gamma - \gamma^k) \leq 0, \quad \forall \gamma \in B.$$

Since $\gamma^k \notin \{\eta_1, \eta_2\}$, h is differentiable at γ^k, with

$$\nabla h(\gamma^k) = c + \sum_{t\in\{1,2\}} r_t \frac{\gamma^k - \eta_t}{\|\gamma^k - \eta_t\|} = c + \sum_{t\in\{1,2\}} w_t(\gamma^k - \eta_t) = W\gamma^k - v = -W(\xi - \gamma^k).$$

As $W > 0$, the previous projection inequality becomes

$$\nabla h(\gamma^k)^\top(\gamma - \gamma^k) \geq 0, \qquad \forall \gamma \in B.$$

Since h is convex and differentiable at γ^k, we have $h(\gamma) \geq h(\gamma^k) + \nabla h(\gamma^k)^\top(\gamma - \gamma^k) \geq h(\gamma^k)$ for all $\gamma \in B$, so γ^k minimizes h over B. □

An important caveat is that update (B1) is not defined at $\gamma = \eta_1$ and $\gamma = \eta_2$. However, their optimality can be checked in closed form.

Lemma 1. *Let $r_1 > 0$.*

- *If $\eta_1 \neq \eta_2$, define $G = c + r_2 \dfrac{\eta_1 - \eta_2}{\|\eta_1 - \eta_2\|}$ and $\rho = r_1$.*
- *If $\eta_1 = \eta_2$, define $G = c$ and $\rho = r_1 + r_2$.*

Then, η_1 minimizes h over B if and only if

- $\|G\| \le \rho$, *when* $\|\eta_1\| < 1$, *or when* $\|\eta_1\| = 1$ *and* $G^\top \eta_1 \ge 0$;
- $\|G - (G^\top \eta_1)\,\eta_1\| \le \rho$, *when* $\|\eta_1\| = 1$ *and* $G^\top \eta_1 < 0$.

Proof. As h is convex and finite on $\mathbb{R}^2$, and B is closed and convex with nonempty interior, η_1 minimizes h over B if and only if $0 \in \partial h(\eta_1) + N_B(\eta_1)$, where $N_B(\eta_1)$ denotes the normal cone of B at η_1.

The norm term centered at η_1, with weight ρ, has subdifferential $\{u : \|u\| \le \rho\}$ at η_1, while the remaining terms of h are differentiable at η_1 with gradient G. Hence, the optimality condition holds if and only if there exists $n \in N_B(\eta_1)$ such that $\|G+n\| \le \rho$.

If $\|\eta_1\| < 1$, then $N_B(\eta_1) = \{0\}$, and the condition reads $\|G\| \le \rho$. If $\|\eta_1\| = 1$, then $N_B(\eta_1) = \{t\eta_1 : t \ge 0\}$, and the condition reads $\min_{t\ge 0} \|G + t\eta_1\| \le \rho$. Since

$$\|G + t\eta_1\| = \sqrt{\|G\|^2 + 2t\, G^\top \eta_1 + t^2},$$

the minimum over $t \ge 0$ is attained at $t = \max\{0, -G^\top \eta_1\}$. If $G^\top \eta_1 \ge 0$, the minimum is at $t = 0$, giving $\|G\|$. Otherwise, the minimum value is $\sqrt{\|G\|^2 - (G^\top \eta_1)^2} = \|G - (G^\top \eta_1)\,\eta_1\|$. □

Given these results, $\tilde{\gamma}$ is computed as follows. First, both η_1 and η_2 are tested with Lemma 1; if one of them is optimal, it is taken as $\tilde{\gamma}$. Otherwise, update (B1) is applied for a fixed number of iterations from an initial point of B, stopping early if an iterate coincides with η_1 or η_2. Finally, the resulting iterate is compared with η_1, η_2, and the trivial completion $\gamma = 0$, and the point with the smallest value of h is taken as $\tilde{\gamma}$. Every point considered lies in B, and hence yields a valid cut; by Proposition 5, h does not increase along the iterations, and the final comparison guarantees $h(\tilde{\gamma}) \le \min\{h(\eta_1), h(\eta_2), h(0)\}$.

Appendix C Tightness of (15) under reorderings

Throughout this appendix, we fix a tour $\hat{x}$, an optimal solution $\hat{\gamma}$ of (16), the corresponding $\hat{\eta}$, and the coefficients (17), as in Section 5.

C.1 The subproblem at a tour

Since $\hat{x}_{ij} = 0$ outside $S(\hat{x})$, constraints (13b) and (13c) force the copies of off-support arcs to vanish, so the corresponding distance variables vanish at optimality, and (13e) reduces to $p_i^{i\,\mathrm{nxt}(i)} = p_i^{\mathrm{prv}(i)\,i}$. Writing p_i for this common value, $Q^P(\hat{x})$ is the optimal value of

$$\min \left\{ \sum_{(i,j)\in S(\hat{x})} (d_{ij} - m_{ij}) : \|p_i - c_i\| \le r_i,\ \forall i \in N,\ \|p_i - p_j\| \le d_{ij},\ \forall (i,j) \in S(\hat{x}) \right\}.$$

For $\gamma = (\gamma_{ij})_{(i,j)\in S(\hat{x})}$, let $\Delta_i(\gamma) := \gamma_{i\,\mathrm{nxt}(i)} - \gamma_{\mathrm{prv}(i)\,i}$ for each $i \in N$, and

$$\Phi(\gamma) := \sum_{(i,j)\in S(\hat{x})} \big((c_i - c_j)^\top \gamma_{ij} - m_{ij}\big) - \sum_{i\in N} r_i \|\Delta_i(\gamma)\|,$$

so that, eliminating τ from (16), $Q^P(\hat{x}) = \max\{\Phi(\gamma) : \|\gamma_{ij}\| \le 1,\ \forall (i,j) \in S(\hat{x})\}$.

Lemma 2. *For any feasible (p, d) and any γ with $\|\gamma_{ij}\| \le 1$ for all $(i,j) \in S(\hat{x})$,*

$$\sum_{(i,j)\in S(\hat{x})} (d_{ij} - m_{ij}) \overset{(1)}{\ge} \sum_{(i,j)\in S(\hat{x})} (\|p_i - p_j\| - m_{ij}) \overset{(2)}{\ge} \sum_{(i,j)\in S(\hat{x})} \big((p_i - p_j)^\top \gamma_{ij} - m_{ij}\big)$$
$$\overset{(3)}{=} \sum_{(i,j)\in S(\hat{x})} \big((c_i - c_j)^\top \gamma_{ij} - m_{ij}\big) + \sum_{i\in N} (p_i - c_i)^\top \Delta_i(\gamma) \overset{(4)}{\ge} \Phi(\gamma).$$

Moreover, the following hold:

(a) If $(\hat{p}, \hat{d})$ and γ are optimal, then $\gamma_{ij} = (\hat{p}_i - \hat{p}_j)/\|\hat{p}_i - \hat{p}_j\|$ for every $(i,j) \in S(\hat{x})$ with $\hat{p}_i \ne \hat{p}_j$.

(b) If $(\hat{p}, \hat{d})$ is optimal, γ is feasible, and every termwise inequality in (1), (2) and (4) holds with equality at $(\hat{p}, \hat{d}, \gamma)$, then γ is optimal.

(c) If $(\hat{p}, \hat{d})$ and γ are optimal and $\|\hat{p}_v - c_v\| < r_v$ for some $v \in N$, then $\Delta_v(\gamma) = 0$.

Proof. Inequality (1) follows from the distance constraints, and (2) from the Cauchy-Schwarz inequality on each arc, as $\|\gamma_{ij}\| \le 1$. Equality (3) follows from writing $p_i - p_j = (c_i - c_j) + (p_i - c_i) - (p_j - c_j)$ and grouping the terms by location: each location i appears as the tail of $(i, \mathrm{nxt}(i))$ and as the head of $(\mathrm{prv}(i), i)$, which yields $(p_i - c_i)^\top \Delta_i(\gamma)$. Inequality (4) follows, for each location, from

$$(p_i - c_i)^\top \Delta_i(\gamma) \ge -\|p_i - c_i\|\,\|\Delta_i(\gamma)\| \ge -r_i \|\Delta_i(\gamma)\|,$$

by the Cauchy-Schwarz inequality and the neighborhood constraints. Each of (1), (2) and (4) is a sum of termwise inequalities, one per arc or per location.

If $(\hat{p}, \hat{d})$ and γ are optimal, both ends of the chain equal $Q^P(\hat{x})$ by strong duality, so every termwise inequality holds with equality. For (a), consider the term of arc (i,j) in (2): if $\hat{p}_i \ne \hat{p}_j$, then $(\hat{p}_i - \hat{p}_j)^\top \gamma_{ij} = \|\hat{p}_i - \hat{p}_j\|$ with $\|\gamma_{ij}\| \le 1$ holds if and only if $\gamma_{ij} = (\hat{p}_i - \hat{p}_j)/\|\hat{p}_i - \hat{p}_j\|$. For (c), if $\Delta_v(\gamma) \ne 0$ and $\|\hat{p}_v - c_v\| < r_v$, then the term of v in (4) satisfies $(\hat{p}_v - c_v)^\top \Delta_v(\gamma) \ge -\|\hat{p}_v - c_v\|\,\|\Delta_v(\gamma)\| > -r_v \|\Delta_v(\gamma)\|$, a contradiction. Finally, for (b), if every termwise inequality holds with equality, then $\Phi(\gamma)$ equals the objective value of $(\hat{p}, \hat{d})$, which is $Q^P(\hat{x})$, so γ is optimal. □

By Lemma 2(a), on arcs of positive length, $-\hat{\gamma}_{ij}$ is the direction of travel from $\hat{p}_i$ to $\hat{p}_j$.

C.2 Reorderings and the cut

A *stretch* of $\hat{x}$ is a sequence $(a, v_1, \dots, v_k, b)$ of distinct consecutive locations visited by $\hat{x}$, and its *cluster* is $V = \{v_1, \dots, v_k\}$. For an ordering $\sigma = (v_{\sigma(1)}, \dots, v_{\sigma(k)})$ of V, let

$$T^\sigma := \big\{(a, v_{\sigma(1)}),\ (v_{\sigma(1)}, v_{\sigma(2)}),\ \dots,\ (v_{\sigma(k)}, b)\big\}$$

be the arcs traversing V in the order σ, write $T := T^{\mathrm{id}}$, and let $\hat{x}^\sigma$ be the tour with support $(S(\hat{x}) \setminus T) \cup T^\sigma$.

For a tour x, let $L(x) := Q^P(x) + \sum_{(i,j)\in A} m_{ij} x_{ij}$ be its length. Since each location has exactly one outgoing and one incoming arc in any tour, the terms $c_i^\top \hat{\eta}_i - c_j^\top \hat{\eta}_j$ of (17) cancel, and (15) reads

$$\sum_{(i,j)\in A} m_{ij} x_{ij} + \theta \geq \mathrm{LB}(x) := -\sum_{(i,j)\in A} h_{ij}(\tilde{\gamma}_{ij}) x_{ij}.$$

By the validity of (15), $\mathrm{LB}(x) \leq L(x)$ for every tour x, and, as the cut is tight at $\hat{x}$, $\mathrm{LB}(\hat{x}) = L(\hat{x})$. As $\hat{x}$ and $\hat{x}^\sigma$ coincide outside the stretch,

$$\mathrm{LB}(\hat{x}^\sigma) - \mathrm{LB}(\hat{x}) = \sum_{(i,j)\in T} h_{ij}(\tilde{\gamma}_{ij}) - \sum_{(i,j)\in T^\sigma} h_{ij}(\tilde{\gamma}_{ij}). \tag{C4}$$

Both T and T^σ are paths from a to b through V. Hence, for any $u \in \mathbb{R}^2$, the linear terms of $h_{ij}(u)$ telescope to $(c_b - c_a)^\top u$ along either path, and each norm term $r_l \|\hat{\eta}_l - u\|$ appears the same number of times in both sums: once for $l \in \{a, b\}$, and twice for $l \in V$. Thus,

$$\sum_{(i,j)\in T} h_{ij}(u) = \sum_{(i,j)\in T^\sigma} h_{ij}(u), \qquad \forall u \in \mathbb{R}^2. \tag{C5}$$

C.3 Passive clusters

We say that V is *passive* if some optimal solution of (16) takes a common value u on all arcs of T. For such a solution, $\hat{\eta}_v = u$ for every $v \in V$.

Corollary 1. *If V is passive, then, for every optimal $\hat{p}$, the points $\hat{p}_a, \hat{p}_{v_1}, \dots, \hat{p}_{v_k}, \hat{p}_b$ lie, in this order, on a segment.*

Proof. Let γ be optimal with value u on T. By Lemma 2(a), every arc $(i, j) \in T$ satisfies $\hat{p}_j - \hat{p}_i = -\|\hat{p}_i - \hat{p}_j\|\, u$. Hence, each consecutive pair of points along the stretch differs by a nonnegative multiple of $-u$, so the points lie, in this order, on the segment from $\hat{p}_a$ to $\hat{p}_b$. □

Lemma 3. *Let $(\hat{p}, \hat{d})$ be optimal, and suppose that $\hat{p}_a, \hat{p}_{v_1}, \dots, \hat{p}_{v_k}, \hat{p}_b$ lie, in this order, on a segment, with $\hat{p}_a \neq \hat{p}_{v_1}$ and $\hat{p}_{v_k} \neq \hat{p}_b$. Then V is passive.*

Proof. Let $u = (\hat{p}_a - \hat{p}_b)/\|\hat{p}_a - \hat{p}_b\|$, and let γ be any optimal solution of (16). By Lemma 2(a), $\gamma_{av_1} = \gamma_{v_k b} = u$. Define γ' by $\gamma'_{ij} = u$ for $(i, j) \in T$, and $\gamma'_{ij} = \gamma_{ij}$

otherwise, which is feasible as $\|u\| = 1$. We verify that every termwise inequality of Lemma 2 holds with equality at $(\hat{p}, \hat{d}, \gamma')$.

The terms in (1) do not depend on γ. In (2), only the arcs of T changed; for those of positive length, $\hat{p}_i - \hat{p}_j$ is a positive multiple of $\hat{p}_a - \hat{p}_b$ by the order of the points, so $(\hat{p}_i - \hat{p}_j)^\top u = \|\hat{p}_i - \hat{p}_j\|$, while the terms of zero-length arcs are zero on both sides. In (4), the arcs of a and b that belong to T already took value u under γ, so $\Delta_a(\gamma') = \Delta_a(\gamma)$ and $\Delta_b(\gamma') = \Delta_b(\gamma)$, and the remaining locations outside V are unaffected; for $v \in V$, $\Delta_v(\gamma') = u - u = 0$, so both sides of the term of v are zero. By Lemma 2(b), γ' is optimal, and, as it is constant on T, V is passive. □

Corollary 2. *If, for each $v \in V$, some optimal solution of the subproblem places $\hat{p}_v$ in the interior of its neighborhood, then every optimal solution of (16) is constant on T. In particular, V is passive.*

Proof. Let γ be optimal. For each $v \in V$, applying Lemma 2(c) to γ and to an optimal solution of the subproblem with $\hat{p}_v$ in the interior of its neighborhood yields $\Delta_v(\gamma) = 0$. Then $\gamma_{av_1} = \gamma_{v_1v_2} = \cdots = \gamma_{v_kb}$. □

C.4 Tightness at reorderings

Proposition 6. *Let V be passive, and let (15) be generated from an optimal $\hat{\gamma}$ with value u on all arcs of T, and with $\|\tilde{\gamma}_{ij}\| \le 1$ and $h_{ij}(\tilde{\gamma}_{ij}) \le \min\{h_{ij}(\hat{\eta}_i), h_{ij}(\hat{\eta}_j)\}$ for every $(i,j) \notin S(\hat{x})$. Then $\mathrm{LB}(\hat{x}^\sigma) \ge \mathrm{LB}(\hat{x})$ for every ordering σ of V. Therefore, $L(\hat{x}^\sigma) \ge L(\hat{x})$, and the cut is tight at every $\hat{x}^\sigma$ with $L(\hat{x}^\sigma) = L(\hat{x})$.*

Proof. Consider the right-hand side of (C4). As $T \subseteq S(\hat{x})$, the first sum equals $\sum_{(i,j)\in T} h_{ij}(u)$. The arcs of T^σ join locations of $\{a, b\} \cup V$, and the only arcs of $S(\hat{x})$ among these locations are those of T and, possibly, (b, a), which never belongs to T^σ. Hence, the arcs of T^σ in $S(\hat{x})$ are those of $T^\sigma \cap T$, which have argument u, while, for the remaining arcs of T^σ, each arc has an endpoint $v \in V$, for which $\hat{\eta}_v = u$. Thus, the assumed bound implies $h_{ij}(\tilde{\gamma}_{ij}) \le h_{ij}(u)$. Then, by (C5), the second sum is at most $\sum_{(i,j)\in T^\sigma} h_{ij}(u) = \sum_{(i,j)\in T} h_{ij}(u)$, so $\mathrm{LB}(\hat{x}^\sigma) \ge \mathrm{LB}(\hat{x})$. The remaining claims follow from $L(\hat{x}^\sigma) \ge \mathrm{LB}(\hat{x}^\sigma)$ and $\mathrm{LB}(\hat{x}) = L(\hat{x})$. □

Remark. *The procedure of Appendix B guarantees the conditions on the off-support coefficients required by Proposition 6. Moreover, by Corollary 2, if every location of V can be visited in the interior of its neighborhood at optimality, the solution of (16) returned by the solver is constant on T, so the proposition applies to the cut actually generated. Otherwise, the solver may return an optimal solution that is not constant on T; the cut remains valid, but its tightness at the reorderings is not guaranteed.*