A Quasi-Polynomial Algorithm for Well-Spaced Hyperbolic TSP

Sándor Kisfaludi-Bak
Max Planck Institute for Informatics, Saarbrücken, Germany
sandor.kisfaludi-bak@mpi-inf.mpg.de

Abstract

We study the traveling salesman problem in the hyperbolic plane of Gaussian curvature -1. Let α denote the minimum distance between any two input points. Using a new separator theorem and a new rerouting argument, we give an $n^{O(\log^2 n)\max(1,1/\alpha)}$ algorithm for Hyperbolic TSP. This is quasi-polynomial time if α is at least some absolute constant, and it grows to $n^{O(\sqrt{n})}$ as α decreases to $\log^2 n/\sqrt{n}$. (For even smaller values of α, we can use a planarity-based algorithm of Hwang et al. (1993), which gives a running time of $n^{O(\sqrt{n})}$.)

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Computational geometry, Hyperbolic geometry, Traveling salesman

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.55

Related Version A full version of the paper is available at [19], `https://arxiv.org/abs/2002.05414`.

1 Introduction

The Traveling Salesman Problem (or TSP for short) is very widely studied in combinatorial optimization and computer science in general, with a long history. In the general formulation, we are given a complete graph G with positive weights on its edges. The task is to find a cycle through all the vertices (i.e., a Hamiltonian cycle) of minimum weight. The first non-trivial algorithm (with running time $O(2^n n^2)$) was given by Held and Karp [11], and independently by Bellman [3]. The problem was among the first problems to be shown NP-hard by Karp [16].

A very important case of TSP concerns metric weight functions, where the edge weights satisfy the triangle inequality. The problem has a (3/2)-approximation due to Christofides [6], which is still unbeaten. On the other hand, it is NP-hard to approximate Metric TSP within a factor of 123/122 [17]. Fortunately, the problem is more tractable in low-dimensional geometric spaces. Arora [1] and independently, Mitchell [21] gave the first polynomial time approximation schemes (PTASes) for the low-dimensional Euclidean TSP problem, where vertices correspond to points in $\mathbb{R}^d$ and the weights are defined by the Euclidean distance between the given points. The PTAS was later improved by Rao and Smith [23], and after two decades, several more general approximation schemes are known. In particular, there is a PTAS in metric spaces of bounded doubling dimension by Bartal *et al.* [2], and in metric spaces of negative curvature by Krauthgamer and Lee [20]. The PTAS of [20] applies in the hyperbolic plane.

Turning to the exact version of the problem in the geometric setting, we can again get significant improvements over the best known $O(2^n \operatorname{poly}(n))$ running time for the general version. In the Euclidean case, the first set of improved algorithms were proposed in the plane by Kann [15] and by Hwang *et al.* [12] with running time $n^{O(\sqrt{n})}$. Later, an algorithm in $\mathbb{R}^d$ with running time $n^{O(n^{1-1/d})}$ was given by Smith and Wormald [24]. The latest improvement to $2^{O(n^{1-1/d})}$ by De Berg *et al.* [7] came with a matching lower bound under the Exponential Time Hypothesis (ETH) [13]. To our knowledge, the exact version of the problem in hyperbolic space has not been studied yet.

Given the history of the problem, the PTAS results and the Euclidean exact algorithm, one might expect that the hyperbolic case is very similar to the Euclidean, and a good hyperbolic TSP algorithm will have a running time of $n^{O(n^\delta)}$ for some constant δ. In this paper, we show that we can often get significantly faster algorithms. Let $\mathbb{H}^2$ denote the hyperbolic plane of Gaussian curvature -1. The first hopeful sign is that $\mathbb{H}^2$ exhibits special properties when it comes to intersection graphs. Recently, the present author has given quasi-polynomial algorithms for several classic graph problems in certain hyperbolic intersection graphs of ball-like objects [18]. The studied problems include Independent Set, Dominating Set, Steiner Tree, Hamiltonian Cycle and several other problems that are NP-complete in general graphs. Interestingly, a polynomial time algorithm was given for the Hamiltonian Cycle problem in hyperbolic unit disk graphs. The question arises whether a quasi-polynomial algorithm is available for TSP in $\mathbb{H}^2$? Given that the best running times for Hamiltonian Cycle in unit disk graphs in $\mathbb{R}^2$ and for Euclidean TSP are identical, perhaps even polynomial time is achievable for Hyperbolic TSP?

Unfortunately, a quasi-polynomial algorithm is unlikely to exist for the general Hyperbolic TSP problem: the lower bound of [8] in grids can be carried over to $\mathbb{H}^2$, which rules out a $2^{o(\sqrt{n})}$ algorithm under the Exponential Time Hypothesis (ETH) [13]. This however relies on embedding a grid-like structure in $\mathbb{H}^2$ efficiently, which seems to be possible only if the points are densely placed. Since $\mathbb{H}^2$ is locally Euclidean, it comes as no surprise that we cannot beat the Euclidean running time for dense point sets.

For this reason, we use a parameter measuring the density of the input point set. We say that the input point set P is *α-spaced* if for any pair of distinct points $p, p' \in P$, we have that $\mathrm{dist}(p, p') \geqslant \alpha$. Our main contribution is the following theorem.

▶ **Theorem 1.** *Let $P \subset \mathbb{H}^2$ be an α-spaced set of points. Then the shortest traveling salesman tour of P can be computed in $n^{O(\log^2 n)\cdot \max(1,1/\alpha))}$ time.*

Note that for $\alpha \geqslant 1$, this is a quasi-polynomial algorithm. In the full version [19] we show that for very dense inputs, it is unlikely that our running time can be improved significantly: we prove that there is no $2^{o(\sqrt{n})}$ algorithm for point sets of spacing $\Theta(1/\sqrt{n})$, unless the Exponential Time Hypothesis (ETH) fails.

Adapting algorithms from the Euclidean plane

Most algorithms for Euclidean TSP are difficult to adapt to the hyperbolic setting. The majority of known subexponential algorithms for Euclidean TSP (see [15, 24, 7]) are based on some version of the so-called *Packing Property* [7], which roughly states that for any disk δ of radius r, the number of segments in an optimal tour of length at least r that intersect δ is at most some absolute constant. This starting point is not available to us, since a direct adaptation of the Packing Property as stated above is false in $\mathbb{H}^2$. For example, we can create a regular n-gon where the length of each side is $c \log n$ for some constant c, and the inscribed circle has radius $r < c \log n$. The boundary of the n-gon is an optimal tour of its vertices, and the inscribed disk is intersected more than a constant times with tour segments of length at least r.

The only exact Euclidean TSP algorithm that directly carries over to $\mathbb{H}^2$ is the algorithm of Hwang, Chang and Lee [12], as it only relies on the fact that any optimal tour in the plane is crossing-free. Unfortunately, this algorithm has a running time of $n^{O(\sqrt{n})}$, which is far from our goal. Nonetheless, we can use this algorithm for the case when the point set P has close point pairs, that is, when $\alpha \leqslant \log^2 n/\sqrt{n}$. This is discussed further in Section 2.

Our techniques

To get a quasi-polynomial algorithm for $\alpha = \Omega(1)$, we need to prove our own separator theorem. The separator itself is fairly simple: it is a line segment of length $O(\log n)$. Due to the special properties of $\mathbb{H}^2$, optimal tours cannot go "around" this segment. The difficulty is to show that the line segment is crossed only $O(\log n)$ times by an optimal tour. We show that having a pair of "nearby"[1] tour edges crossing a certain neighborhood R of the segment can be ruled out with a rerouting argument that is reminiscent of the proof of the Packing Property in $\mathbb{R}^2$. This limits the number of segments crossing both R and the segment to $O(\log n)$. All other tour edges crossing the segment must have an endpoint in R. Since R is "narrow", it can contain at most $O(\log n)$ points from P, as P is α-spaced. These bounds together limit the number of tour edges crossing our segment to $O(\log n)$. With the separator at hand, we use a standard divide-and-conquer algorithm to prove Theorem 1. For values $\alpha \leqslant \log^2 n/\sqrt{n}$, we suggest using the algorithm of Hwang *et al.* [12].

Computational model

As our input, we get a list of points P with rational coordinates in the Poincaré disk model (which we briefly introduce in Section 2) and a rational number x. The goal is to decide if there is a tour of length at most x.

It is a common issue in computational geometry that one needs to be able to compare sums of distances. In geometric variants of TSP, this directly impacts the output, and unfortunately no method is known to tackle this in a satisfactory manner on a word-RAM machine. For this reason, most work in the area assumes that the computation is done on a real-RAM machine that can compute square roots exactly. Perhaps even less is known about comparing sums of distances in hyperbolic space. For this article, we work in a real-RAM that, in addition to taking square roots, is also capable of computing the natural logarithm $\ln(.)$.

2 Preliminaries

The hyperbolic plane and the Poincaré disk model

Introducing the hyperbolic plane properly is well beyond the scope of this section, but we list some important properties that we will be using. A detailed exposition can be found in several textbooks [4, 26, 10, 22].

The hyperbolic plane $\mathbb{H}^2$ is a homogeneous metric space with the key property that the area and circumference of disks grows exponentially with the radius, that is, a disk of radius r has area $4\pi\sinh^2(r/2)$ and circumference $2\pi\sinh(r)$. For $r > 1$, both the area and circumference are $\Theta(e^r)$. On the other hand, a small neighborhood of any point in the hyperbolic plane is very similar to a small neighborhood of a point in the Euclidean plane. More precisely, the disk of radius ε around a point in $\mathbb{H}^2$ and $\mathbb{R}^2$ have a smooth bijective mapping that preserve distances up to a multiplicative factor of $1 + f(\varepsilon)$, where $\lim_{\varepsilon\to 0} f(\varepsilon) = 0$.

The hyperbolic plane itself can be defined in many ways, but it is most convenient to take some region of $\mathbb{R}^2$, and equip it with a custom metric. Such definitions are also called *models* of the hyperbolic plane. In this article, we use the Poincaré disk model for all of the figures, however, most of the claims and proofs are model-independent.

[1] The absolute distance of crossing edges cannot be bounded; we use a special definition of "nearby".

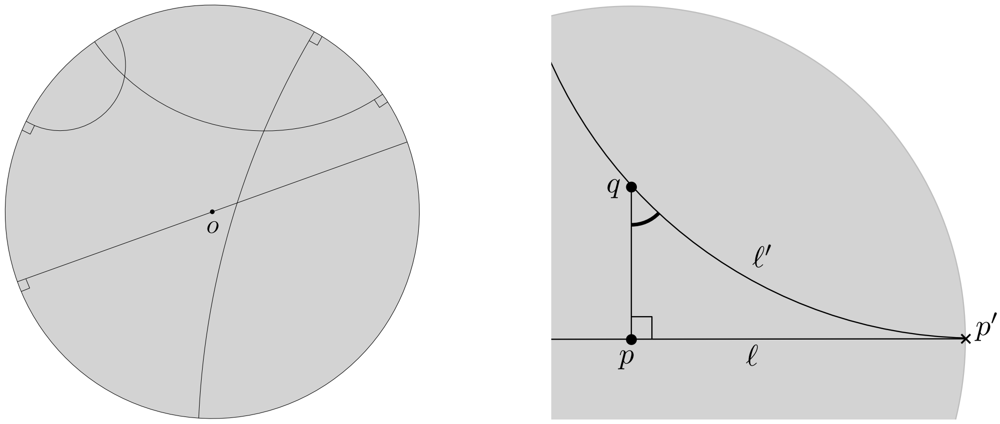

Figure 1 Left: lines in the Poincaré model. Right: the angle of parallelism for the length $|pq|$.

The Poincaré disk model is the open unit disk of $\mathbb{R}^2$ equipped with the distance function

$$\operatorname{dist}(u,v) = \cosh^{-1}\left(1 + 2\frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)}\right),$$

where $\|.\|$ is the Euclidean norm.[2] The precise function here is irrelevant; we present the formula just as an example of defining a custom metric space.[3] We list some further properties of $\mathbb{H}^2$ used in the article.

- *Lines, angles, and ideal points.*
  In the Poincaré disk model hyperbolic lines appear as Euclidean circular arcs that are perpendicular to the unit circle, as illustrated on the left of Figure 1. In particular, hyperbolic lines through the center of the disk are diametrical segments of the unit disk. The model is *conformal*, that is, the angle of a pair of lines in $\mathbb{H}^2$ is the same as the angle of the corresponding arcs in $\mathbb{R}^2$. The points on the boundary of the disk are called *ideal points*.
- *Angle of parallelism.*
  Let $p, q \in \mathbb{H}^2$ and let ℓ be a line through p that is perpendicular to pq, and let p' be an ideal point of ℓ, see the right hand side of Figure 1. Let ℓ' be the line through q and p'. Note that ℓ and ℓ' are disjoint lines in the open disk; they are called *limiting parallels*. The angle $\sphericalangle pqp'$ is called the *angle of parallelism*, which only depends on the length of the segment pq the following way [22].

  $$\tan(\sphericalangle pqp') = \frac{1}{\sinh(|pq|)} \tag{1}$$

- *Hypercycles or equidistant curves.*
  The set of points at a given distance ϱ from a line ℓ is not a line, but it forms a *hypercycle* in $\mathbb{H}^2$. A hypercycle has two arcs, one on each side of ℓ. In the Poincaré model, a hypercycle for a line ℓ consists of two circular arcs, ending at the same ideal points as ℓ.

---

[2] As $\cosh^{-1}(x) = \ln(x + \sqrt{x^2-1})$, the distance of two points in the Poincaré disk model with given Euclidean coordinates can be computed on a real-RAM machine which is capable of taking square roots and computing $\ln(.)$.

[3] If we need to calculate angles, curve length, and area, we should define the metric tensor instead: $ds^2 = 4\frac{\|dx\|^2}{(1-\|x\|^2)^2}$ [5].

- *Optimal tours in $\mathbb{H}^2$ and crossings.*
  An optimal traveling salesman tour will consist of geodesics between pairs of input points, i.e., hyperbolic segments, just as in $\mathbb{R}^2$. Moreover, the triangle inequality implies that any self-crossing tour (where two segments pp' and qq' cross) can be shortened. Thus, optimal tours in $\mathbb{H}^2$ are non-crossing.

Getting a subexponential algorithm for all values of α

We can give the following more general formulation of the result of [12].

▶ **Theorem 2** (Hwang, Chang and Lee [12], stated generally). *Let P be a set of n points in $\mathbb{R}^2$, and let $w : \binom{P}{2} \to \mathbb{R}$ be a weight function on the (straight) segments defined by the point pairs. Suppose that the optimal TSP tour of P with respect to w is crossing-free. Then there is an algorithm to compute this optimal tour in $n^{O(\sqrt{n})}$ time.*

We convert our initial point set P in the Poincaré model to the Beltrami-Klein model of $\mathbb{H}^2$ to get a point set P_{BK}. In the Beltrami-Klein model, Euclidean segments inside the open unit disk are (geodesic) segments of $\mathbb{H}^2$. Since the optimal hyperbolic TSP tour is crossing-free, the tour in the Beltrami-Klein model is a polygon with vertex set P_{BK}. The hyperbolic distances can be used as weights on all segments with endpoints from P_{BK}, and we can apply Theorem 2 to get an $n^{O(\sqrt{n})}$ algorithm regardless of the value of α.

3 A separator for Hyperbolic TSP

Centerpoint and a separating line

It has already been observed in [18] that for any set $P \subset \mathbb{H}^2$ of n points there exists a point $q \in \mathbb{H}^2$ such that for any line ℓ through q the two open half-planes with boundary ℓ both contain at most $\frac{2}{3}n$ points from P, that is, the line ℓ is a $2/3$-*balanced separator* of P. Such a point q is called a *centerpoint* of P. It has been observed in [18] that given P, a centerpoint of P can be computed using a Euclidean centerpoint algorithm, which takes linear time [14].

It is now easy to prove that we can find a balanced line separator that has a small neighborhood empty of input points.

▶ **Lemma 3.** *Given a point set $P \subset \mathbb{H}^2$, there exists a point $q \in \mathbb{H}^2$ and there exists a line ℓ through q such that P is disjoint from the open double cone with center q, axis ℓ and half-angle $\frac{\pi}{2n}$. Any such line ℓ is a $2/3$-balanced separator of P, and given P, a suitable point q and line ℓ can be found in linear time.*

Proof. Let q be a centerpoint of P. For each point $p \in P$, let ℓ_p be the line through q and p. Since we have defined n lines through q, there is a pair of consecutive lines $\ell_p, \ell_{p'}$ whose acute angle is at least π/n. Let ℓ be the angle bisector of ℓ_p and $\ell_{p'}$. Then ℓ clearly has the desired properties, and the centerpoint q, the lines $\ell_p, \ell_{p'}$ and the line ℓ can all be computed in linear time. ◀

We can extend Lemma 3 to get balance with respect to a subset $B \subseteq P$, that is, both half-planes bounded by ℓ would contain at most $\frac{2}{3}|B|$ points of B. One only needs to set q to be the centerpoint of B instead of P.

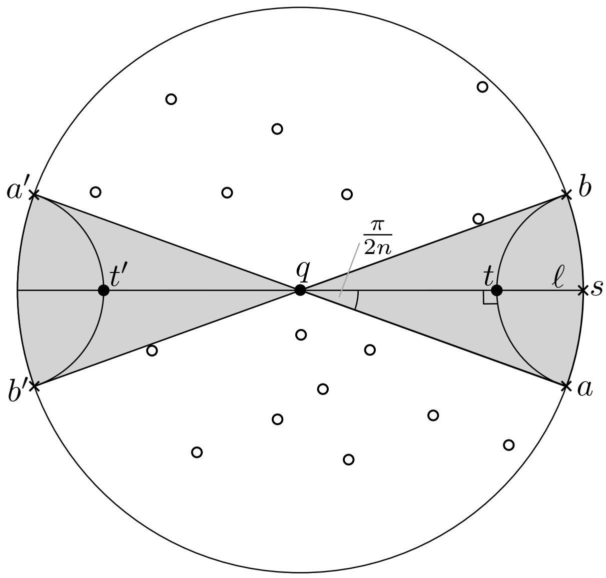

Figure 2 The empty double cone with axis ℓ.

Defining a region around the separator

From this point onwards, q denotes a centerpoint of P, and ℓ is a line through q with the properties from Lemma 3. Let C denote the double cone of center q, axis ℓ and half-angle $\frac{\pi}{2n}$, see Figure 2. Note that by Lemma 3, we have that $C \cap P = \emptyset$. Let s be an ideal point of ℓ, and let a, b be ideal points on the boundary of C, such that $\sphericalangle aqs = \sphericalangle sqb = \frac{\pi}{2n}$. Let $t = ab \cap \ell$. Notice that qta is a right-angle triangle with ideal point a, and it has angle $\frac{\pi}{2n}$ at q. Therefore, $\frac{\pi}{2n}$ is the angle of parallelism for the distance $|qt|$, and it satisfies

$$\sinh(|qt|) = \frac{1}{\tan(\frac{\pi}{2n})}. \tag{2}$$

The line ab splits $\mathbb{H}^2$ into two open half-planes: the side H_q containing q, and the side H_s that has s on its boundary. Note that $H_s \subset C$, therefore $P \subset H_q$. Consequently, all segments of the tour are contained in H_q. We mirror a, b and t to the point q; let a', b' and t' denote the resulting points respectively. By our earlier observation, the entire tour is contained in the geodesically convex region between the lines ab and $a'b'$, and any tour segment intersecting ℓ will intersect it somewhere on the segment tt'.

Let a_t and b_t be the points on $a'b'$ at distance ϱ from t, where $\varrho \in (0, \alpha/2)$ is a suitable number that will be defined later. Let a'_t and b'_t denote the analogous points on $a'b'$, see Figure 3. Let R denote the region of the hyperbolic plane consisting of all points between ab and $a'b'$ whose distance from tt' is at most ϱ. The resulting shape R is geodesically convex; its boundary consists of two segments (a_tb_t and $a'_tb'_t$), and two hypercycle arcs, denoted by $\widehat{a_tb'_t}$ and $\widehat{b_ta'_t}$. In general, for two points u, v on one of these hypercycle arcs, let $\widehat{uv}$ denote the arc between them, and let $|\widehat{uv}|$ be the length of this arc.

Note that any tour segment that connects points on two different sides of ℓ also intersects R. A tour segment that intersects R can have $0, 1$ or 2 endpoints in R. A segment with exactly 1 endpoint in R is called *entering*. As R is geodesically convex, segments with both endpoints in R are entirely contained in R. All other tour segments crossing ℓ must intersect at least one of $\widehat{a_tb'_t}$ and $\widehat{b_ta'_t}$. We say that a segment *crosses* R if it intersects both $\widehat{a_tb'_t}$ and $\widehat{b_ta'_t}$. (It is possible that a segment whose endpoints lie outside R on the same side of ℓ intersect one of these arcs twice. These segments are not relevant for our algorithm.)

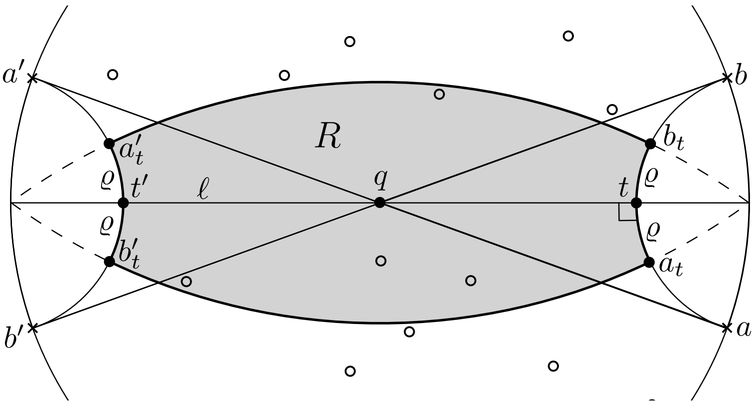

Figure 3 The construction of the region R.

The rest of this section focuses on the following main lemma.

▶ **Lemma 4.** *The region R has the following properties:*
(i) $|R \cap P| < n_{in} \stackrel{\text{def}}{=} 1 + \frac{2(\ln n+1)}{\alpha - 2\varrho}$
(ii) *There are less than $s_{cr} \stackrel{\text{def}}{=} 2 + \frac{2(\ln n+1)\cosh\varrho}{\varrho}$ tour segments that cross R.*

The proof requires that we explore the geometry of R more thoroughly.

▶ **Lemma 5.** *We have $|qt| < \ln n + 1$, and $|\widehat{a_t b'_t}| = |\widehat{b_t a'_t}| < 2(\ln n + 1)\cosh\varrho$.*

Proof. We first prove our bound on $|qt|$. Note that $\sinh(.)$ is monotone increasing and $\sinh(|qt|) = \frac{1}{\tan(\frac{\pi}{2n})}$ by (2), so it suffices to show that $\sinh(\ln n + 1) > \frac{1}{\tan(\frac{\pi}{2n})}$. Indeed,

$$\sinh(\ln n + 1) = \frac{en - \frac{1}{en}}{2} > n \quad \text{and} \quad \frac{1}{\tan(\frac{\pi}{2n})} < \frac{1}{\frac{3}{2n}} < n.$$

The arc length of the equidistant hypercycle of base b and distance ϱ is $b\cosh\varrho$ according to [25], therefore $|\widehat{a_t b'_t}| = |tt'|\cosh(\varrho) = 2|qt|\cosh(\varrho) < 2(\ln n + 1)\cosh\varrho$. ◀

Ruling out dense crossings

Our next ingredient for the proof is to show that if two segments cross R very close to each other, then they cannot both be in an optimal tour.

▶ **Lemma 6.** *Let $p_1 p_2 \dots p_i p_{i+1} \dots p_{n-1} p_n$ be an optimal tour on P where both p_1p_2 and p_ip_{i+1} cross R, and where p_1, p_i and a_t lie on the same side of ℓ. Let $p'_1 = p_1p_2 \cap \widehat{a_t b'_t}$, and define p'_2, p'_i and p'_{i+1} analogously. Then $|\widehat{p'_1 p'_i}| + |\widehat{p'_2 p'_{i+1}}| \geqslant 4\varrho$.*

Proof. We can create a new tour by removing the segments $p'_1p'_2$ and $p'_ip'_{i+1}$, and replacing them with $p'_1p'_i$ and $p'_2p'_{i+1}$, see Figure 4. The resulting tour is

$$p_1 p'_1 p'_i p_i p_{i-1} p_{i-2} \dots p_2 p'_2 p'_{i+1} p_{i+1} p_{i+2} \dots p_n.$$

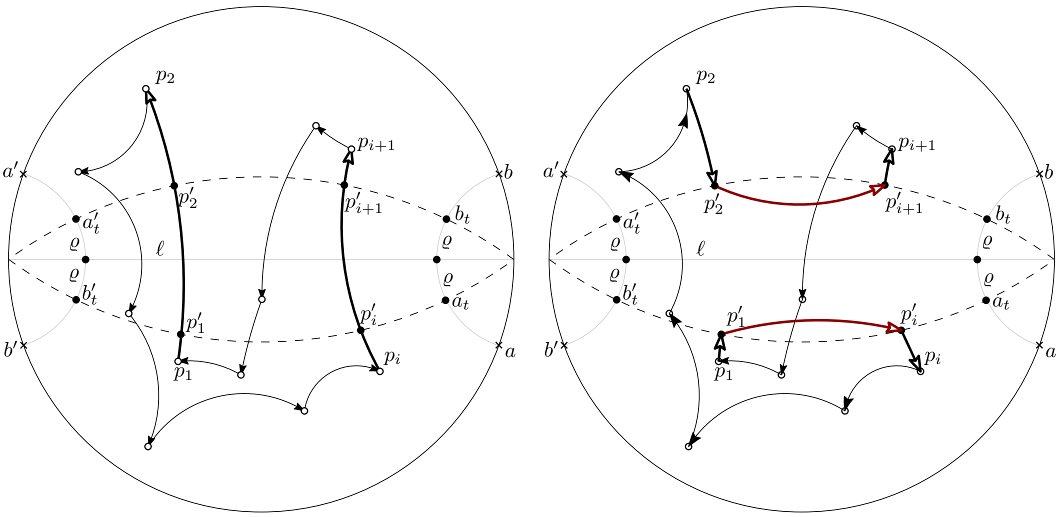

Figure 4 Rerouting two crossing edges (p_1p_2 and p_ip_{i+1}) into a different tour.

Note that this tour contains all the input points.[4] Since the only difference between the tours is that $p'_1p'_2$ and $p'_ip'_{i+1}$ are only present in the optimal tour and $p'_1p'_i$ and $p'_2p'_{i+1}$ are only present in the new tour, by the optimality of $p_1 \dots p_n$ we have that

$$0 \geqslant |p'_1p'_2| + |p'_ip'_{i+1}| - |p'_1p'_i| - |p'_2p'_{i+1}|.$$

Note that $|p'_1p'_2| \geqslant 2\varrho$ by the definition of R, and analogously $|p'_ip'_{i+1}| \geqslant 2\varrho$. Therefore we have

$$0 \geqslant |p'_1p'_2| + |p'_ip'_{i+1}| - |p'_1p'_i| - |p'_2p'_{i+1}| \geqslant 4\varrho - |\widehat{p'_1p'_i}| - |\widehat{p'_2p'_{i+1}}|,$$

which concludes the proof. ◀

We can now prove Lemma 4.

Proof of Lemma 4.

(i) For a point $p \in P \cap R$, let p_ℓ denote the point on ℓ for which pp_ℓ is perpendicular to ℓ. Let $p, p' \in P \cap R$ be points such that p_ℓ, p'_ℓ are consecutive on ℓ (i.e., there is no $p'' \in P \cap R$ such that $p''_\ell \in p_\ell p'_\ell$). By the triangle inequality, $|pp_\ell| + |p_\ell p'_\ell| + |p'_\ell p'| \geqslant |pp'|$, and $|pp'| \geqslant \alpha$ since P is α-spaced. By the definition of R and ϱ, we also have that $|pp_\ell| \leqslant \varrho$ and $|p'_\ell p'| \leqslant \varrho$. Consequently,

$$|p_\ell p'_\ell| \geqslant \alpha - 2\varrho. \tag{3}$$

We can apply this inequality to all consecutive pairs $p_\ell p'_\ell$. Since all the points p_ℓ lie on the segment tt', the total length of the segments $p_\ell p'_\ell$ cannot exceed $|tt'|$. It follows that

$$|P \cap R| \leqslant 1 + \left\lfloor \frac{|tt'|}{\alpha - 2\varrho} \right\rfloor < 1 + \frac{2(\ln n + 1)}{\alpha - 2\varrho},$$

where the second inequality uses our bound from Lemma 5.

[4] This is generally not an optimal tour as it can be further shortened into $p_1p_ip_{i-1}p_{i-2} \dots p_2p_{i+1}p_{i+2} \dots p_n$.

(ii) Let $p_1 \dots, p_n$ be an optimal tour, and let $p_i p_{i+1}$ be an edge crossing R. (Indices are defined modulo n.) Note that $p_i p_{i+1}$ can cross R in two directions: either from the side of a to the side of b or the other way around. By Lemma 6, consecutive crossings $p_i p_{i+1}$ and $p_j p_{j+1}$ in the same direction use at least a total arc length of 4ϱ on the arcs $\widehat{a_t b'_t}$ and $\widehat{b_t a'_t}$. Since the total length of these arcs is less than $4(\ln n + 1)\cosh\varrho$ by Lemma 5, the number of crossings in one direction is less than

$$1 + \left\lfloor \frac{4(\ln n + 1)\cosh\varrho}{4\varrho} \right\rfloor \leqslant 1 + \frac{(\ln n + 1)\cosh\varrho}{\varrho}.$$

Consequently, the total number of crossings (in both directions) is less than

$$2 + \frac{2(\ln n + 1)\cosh\varrho}{\varrho}.$$

This concludes the proof. ◀

4 A divide-and-conquer algorithm

In order for a divide-and-conquer approach to work for Euclidean TSP, one should be able to solve subproblems with partial tours. We follow the terminology and definitions of De Berg *et al.* [7] here. Let M be a perfect matching on a set $B \subseteq P$ of so-called *boundary points*. We say that a collection $\mathcal{P} = \{\pi_1, \dots, \pi_{|B|/2}\}$ of paths *realizes* M *on* P if (i) for each pair $(p, q) \in M$ there is a path $\pi_i \in \mathcal{P}$ with p and q as endpoints, and (ii) the paths together visit each point $p \in P$ exactly once. We define the length of a path π_i to be the sum of the lengths of its edges, and we define the total length of $\mathcal{P}$ to be the sum of the lengths of the paths $\pi_i \in \mathcal{P}$. The subproblems that arise in our divide-and-conquer algorithm can be defined as follows.

Hyperbolic Path Cover
Input: A point set $P \subset \mathbb{H}^2$, a set of boundary points $B \subseteq P$, and a perfect matching M on B.
Task: Find a collection of paths of minimum total length that realizes M on P.

Let $PathTSP(P, B, M)$ be the optimal tour length for the instance (P, B, M). Note that we can solve Hyperbolic TSP on a point set P by solving Hyperbolic Path Cover $n-1$ times on P with $B := \{p, q\}$ and $M := \{(p, q)\}$ for each $q \in P \setminus \{p\}$, and answering

$$\min_{q \in P \setminus \{p\}} \Big(PathTSP\big(P, \{p, q\}, \{(p, q)\}\big) + |pq| \Big).$$

4.1 Algorithm

Our algorithm is a standard divide and conquer algorithm that is very similar to [24] and [7]. The algorithm requires knowledge of the initial value of α; we can compute this before the first call in $O(n^2)$ time. We give a pseudocode and also explain the steps below. In the explanation, we sometimes regard sets of segments with endpoints in P as subgraphs of the complete graph with vertex set P.

As a first step, we run a brute-force algorithm (comparing all path covers of P) if the input points set P has size at most the threshold t, where t will be a large constant. On line 2, we check the size of the boundary. If it is less than $\max\left(\frac{40 \ln |P|}{\alpha}, 8 \ln |P|\right)$, then we

Algorithm 1 *HyperbolicTSP*(P, B, M, α).

Input: A set $P \subset \mathbb{R}^d$, a subset $B \subseteq P$, a perfect matching $M \subseteq \binom{B}{2}$, and initial spacing α
Output: The minimum length of a path cover of P realizing the matching M on B

1: **if** $|P| \leqslant t$ **then return** $\mathit{BruteForceTSP}(P, B, M)$
2: **if** $|B| < \max\left(\frac{40 \ln |P|}{\alpha}, 8 \ln |P|\right)$ **then**
3:     Compute a centerpoint q of P, the line ℓ through q and the region R.
4: **else**
5:     Compute a centerpoint q of B, the line ℓ through q and the region R.
6: $\mathsf{Cr} \leftarrow \{pp' \mid p, p' \in P,\ pp' \text{ crosses } R\}$, $\mathsf{End} \leftarrow \{pp' \mid p \in R \cap P,\ p' \in P, pp' \text{ intersects } \ell\}$
7: $\mathit{mincost} \leftarrow \infty$
8: **for all** $S_{cr} \subseteq \mathsf{Cr}$, $|S_{cr}| \leqslant s_{cr}$ **do**
9:     **for all** $S_{end} \subseteq \mathsf{End}$, the maximum degree of S_{end} is at most 2 **do**
10:         $P_1, P_2 \leftarrow$ uncovered vertices on each side of ℓ
11:         $B_1, B_2 \leftarrow$ boundary vertices of $S_{cr} \cup S_{end}$ and points of B in P_1 (resp., P_2).
12:         **for all** perfect matchings M_1 on B_1 and M_2 on B_2 **do**
13:             **if** $M_1 \cup M_2 \cup S_{cr} \cup S_{end}$ realize M **then**
14:                 $c_1 \leftarrow \mathit{HyperbolicTSP}(P_1, B_1, M_1, \alpha)$
15:                 $c_2 \leftarrow \mathit{HyperbolicTSP}(P_2, B_2, M_2, \alpha)$
16:                 **if** $c_1 + c_2 + \mathbf{length}(S_{cr} \cup S_{end}) < \mathit{mincost}$ **then**
17:                     $\mathit{mincost} \leftarrow c_1 + c_2 + \mathbf{length}(S_{cr} \cup S_{end})$
18: **return** $\mathit{mincost}$

compute the centerpoint of P, the line ℓ with the empty cone according to Lemma 3, and the region R. Otherwise (similarly to [7]), we need to shrink the boundary, so we use a line ℓ through the centerpoint of B instead. Next, we define the segment set Cr as the set of segments pp' that cross R, and End as the set of segments intersecting ℓ that have at least one endpoint in R. We initialize the returned value *mincost* to infinity.

On line 8, we iterate over all segment sets $S_{cr} \subseteq \mathsf{Cr}$ with $|S_{cr}| \leqslant s_{cr}$, where s_{cr} is our bound on the number of crossing segments from Lemma 4. The algorithm considers S_{cr} to be the set of segments crossing R. Next, we iterate over all the sets $S_{end} \subseteq \mathsf{End}$ where each point of P has at most two incident segments from S_{end}. The algorithm considers S_{end} to be the set of segments crossing ℓ with at least one endpoint in R.

Each point in B needs to have one adjacent segment in the optimum tour $\mathcal{P}$, and each point in $P \setminus B$ needs two such points. We say that a point $p \in B$ (resp., $p \in P \setminus B$) is *uncovered* if its degree in $S_{cr} \cup S_{end}$ is less than 1 (resp., 2). We denote by P_1 and P_2 the set of uncovered points on each side of ℓ. A point $p \in P_1$ is a boundary point if $p \in B$ and p is not an endpoint of $S_{cr} \cup S_{end}$, or $p \in P \setminus B$ and it has degree 1 in $S_{cr} \cup S_{end}$. We let B_1 denote the boundary points in P_1. Similarly, B_2 is the set of boundary points in P_2.

Line 13 proceeds by iterating over all perfect matchings M_1 on B_1 and M_2 on B_2. If the graph on $B_1 \cup B_2 \cup B$ formed by $M_1 \cup M_2 \cup S_{cr} \cup S_{end}$ is a set of paths such that contracting edges with an endpoint in $(B_1 \cup B_2) \setminus B$ results in M, then we say that $M_1 \cup M_2 \cup S_{cr} \cup S_{end}$ realize M. If this is the case for a particular choice M_1, M_2, then on lines 14 and 15 we recurse on both P_1 and P_2. The resulting path covers together with $S_{cr} \cup S_{end}$ form a path cover realizing M: if their length is shorter than *mincost*, then we update *mincost*. After the loops have ended, we return *mincost*.

We can also compute the optimum tour itself with a small modification of the algorithm.

Correctness

The same algorithmic strategy has been used several times in the literature [24, 7], so we only give a brief justification. Given an optimal path cover $\mathcal{P}$, the set S_{cr} of segments in $\mathcal{P}$ crossing R has size at most s_{cr} by Lemma 4. The set of segments S_{end} with one endpoint in R has degree at most two at each point of $R \cap P$. Consequently, both sets will be considered in Line 8 and Line 9. The segments of $\mathcal{P}$ not in $S_{cr} \cup S_{end}$ form a path cover of P_1 and P_2 with boundary set B_1 and B_2. These path covers realize some perfect matchings M_1 and M_2 on B_1 and B_2 respectively. The matchings M_1 and M_2 together with $S_{cr} \cup S_{end}$ realize M, therefore M_1 and M_2 will be considered in the loop at line 13. These path covers must be optimal by the optimality of $\mathcal{P}$.

4.2 Analyzing the running time

All non-recursive steps can be handled in $O(n^2)$ time. The number of segment sets S_{cr} to be considered in line 8 is at most $\binom{|\mathsf{Cr}|}{s_{cr}} = O(n^{2s_{cr}})$, since $|\mathsf{Cr}| = O(n^2)$. The number of segment sets S_{end} to be considered is at most $O(n^{2|R\cap P|}) \leqslant O(n^{2n_{in}})$. By Lemma 4, the loop in line 8 has at most

$$O\left(n^{2\left(1+\frac{2(\ln n+1)}{\alpha-2\varrho}+2+\frac{2(\ln n+1)\cosh\varrho}{\varrho}\right)}\right) = O\left(n^{4(\ln n+1)\left(\frac{1}{\alpha-2\varrho}+\frac{\cosh\varrho}{\varrho}\right)+O(1)}\right) \tag{4}$$

iterations. Instead of trying to minimize this expression by our choice of ϱ, we settle for something that is easy to handle. Let

$$\varrho \stackrel{\text{def}}{=} \min\left(\frac{3}{10}\alpha, \frac{12}{10}\right).$$

The exponent of (4) can be bounded the following way. If $\alpha < 4$, then $\varrho = \frac{3}{10}\alpha$, and $\cosh(\varrho) < 1.82$, so we get

$$\begin{aligned}
2(s_{cr}+n_{in}) &= 4(\ln n+1)\left(\frac{1}{\alpha-2\varrho}+\frac{\cosh\varrho}{\varrho}\right)+O(1)\\
&< 4(\ln n+1)\left(\frac{1}{\frac{4}{10}\alpha}+\frac{1.82}{\frac{3}{10}\alpha}\right)+O(1)\\
&< 4(\ln n)\left(\frac{8.57}{\alpha}\right)+O(1/\alpha)\\
&< 35\frac{\ln n}{\alpha},
\end{aligned} \tag{5}$$

where the last step uses that n is large enough, which we can ensure by setting the threshold t in Line 1 large enough. If $\alpha \geqslant 4$, then $\varrho = 1.2$:

$$\begin{aligned}
2(s_{cr}+n_{in}) &= 4(\ln n+1)\left(\frac{1}{\alpha-2\varrho}+\frac{\cosh\varrho}{\varrho}\right)+O(1)\\
&< 4(\ln n+1)\left(\frac{1}{\alpha-2.4}+\frac{1.82}{1.2}\right)+O(1)\\
&< 7\ln n.
\end{aligned} \tag{6}$$

Next, we will analyze the loop at line 13, but this will require a bound on the size of the boundary set B.

▶ **Lemma 7.** *The size of the boundary set B is at most $\max(\frac{60\ln|P|}{\alpha}, 12\ln|P|)$ at every recursion level of HyperbolicTSP.*

Proof. The statement holds for the initial call as we have $|B| = 2$ and $|P| = n$ there.

Notice that if $|B| < \max(\frac{40\ln|P|}{\alpha}, 8\ln|P|)$, then we use the branch on line 3. Consequently, the boundary set B_1 (and B_2) in the new recursive call always has size at most $|B|+(s_{cr}+n_{in})$. So by induction and the bounds (5) and (6), we have that

$$\begin{aligned}|B_1| &\leqslant \max\left(\frac{40\ln|P|}{\alpha}, 8\ln|P|\right) + \max\left(\frac{17.5\ln|P|}{\alpha}, 3.5\ln|P|\right)\\ &< \max\left(\frac{57.5\ln|P|}{\alpha}, 11.5\ln|P|\right)\\ &< \max\left(\frac{60\ln|P_1|}{\alpha}, 12\ln|P_1|\right),\end{aligned}$$

where we use $|P_1| \geqslant |P|/3 \Rightarrow \ln(|P|) < \ln(P_1) + 1.1$; therefore, the last inequality holds if we set the threshold t large enough.

In case of $|B| > \max(\frac{40\ln|P|}{\alpha}, 8\ln|P|)$, we use the branch on line 5. We have that $|B_1| \leqslant \frac{2}{3}|B| + (s_{cr} + n_{in})$. By induction, we still have $|B| < \max(\frac{60\ln|P|}{\alpha}, 12\ln|P|)$, so

$$\begin{aligned}|B_1| &\leqslant \frac{2}{3}\max\left(\frac{60\ln|P|}{\alpha}, 12\ln|P|\right) + \max\left(\frac{17.5\ln|P|}{\alpha}, 3.5\ln|P|\right)\\ &< \max\left(\frac{60\ln|P_1|}{\alpha}, 12\ln|P_1|\right).\end{aligned}$$ ◀

The number of perfect matchings on a boundary set B_1 is at most $|B_1|^{O(|B_1|)}$. Let $b \stackrel{\text{def}}{=} \max(\frac{60\ln|P|}{\alpha}, 12\ln|P|)$ be the bound acquired above. The number of iterations of the loop at line 13 is at most $b^{O(b)}$. If $\alpha \geqslant 4$, then this is $(\ln|P|)^{O(\ln|P|)} = |P|^{O(\ln\ln|P|)} < |P|^{\epsilon\ln|P|}$ for any $\epsilon > 0$, as long as $|P|$ is large enough. If $\alpha < 4$, then we get

$$b^{O(b)} = \left(\frac{\ln n}{\alpha}\right)^{O(\frac{\ln n}{\alpha})} = n^{O(\frac{1}{\alpha}(\ln\ln n + \ln(1/\alpha))}.$$

As long as $1/\alpha = n^{o(1)}$, this term is insignificant compared to the iterations of the other loop. Otherwise, we have $\frac{1}{\alpha} \leqslant \sqrt{n}$, and therefore

$$b^{O(b)} = n^{O(\frac{1}{\alpha}(\ln\ln n + \ln(1/\alpha))} = n^{O(\frac{\log n}{\alpha})}.$$

▶ Remark 8. If one wants to optimize the leading coefficient in the exponent of the eventual running time, then it is possible to modify the algorithm to use only $c^{|B_1|}$ matchings for M_1 as all other matchings lead to crossings. See for example the technique in [9]. As a consequence, the leading coefficient will not be influenced by the second loop at all. However, this effort would be in vain if there exists a significantly better algorithm for $\alpha \leqslant 1$, say $n^{O(\log n\cdot(1/\alpha))}$ or even $n^{O(1/\alpha)}$, which we cannot rule out yet.

The following lemma finishes the proof of Theorem 1.

▶ **Lemma 9.** *The running time of HyperbolicTSP on our initial call is $n^{O(\log^2 n)\max(1,1/\alpha)}$.*

Proof. By the analysis above, the running time for an instance (P, B, M, α) with $|P| = n$ satisfies the following recursion.

$$T(n) \leqslant n^{O(\max(\frac{\log n}{\alpha}, \log n))} T\left(\frac{2}{3}n\right)$$

Therefore, there exists a constant c such that the running time is at most

$$\begin{aligned} T(n) &\leqslant n^{\max(1,1/\alpha)\cdot c\log n}\left(\frac{2}{3}n\right)^{\max(1,1/\alpha)\cdot c(\log(\frac{2}{3}n))}\cdot\left(\frac{4}{9}n\right)^{\max(1,1/\alpha)\cdot c(\log(\frac{4}{9}n))}\cdots \\ &= n^{\max(1,1/\alpha)\cdot c(\log n+\log(\frac{2}{3}n)+\log(\frac{4}{9}n)+\dots)} \\ &= n^{\max(1,1/\alpha)\cdot O(\log^2 n)}. \end{aligned}$$ ◀

5 Conclusion

We have devised a separator theorem in $\mathbb{H}^2$ that led to a quasi-polynomial algorithm for Hyperbolic TSP on constant-spaced point sets. For α-spaced point sets with spacing $\alpha \geqslant \log^2 n/\sqrt{n}$ our algorithm runs in $n^{O(\log^2 n)\max(1,1/\alpha)}$ time. When the point set has spacing only $\Theta(\log^2 n/\sqrt{n})$, the algorithm's performance degrades to the point of reaching (roughly) the performance of the Euclidean algorithm. If the point set has even closer point pairs, then the algorithm of Hwang *et al.* [12] can be used to obtain a running time of $n^{O(\sqrt{n})}$. We have shown that our algorithm's dependence on density is necessary and for spacing $1/\sqrt{n}$, it cannot be significantly improved under ETH. There are several intriguing questions that are left open. We list some of these questions below.

- **Improving the running time, lower bounds.** There is a considerable gap between the running time for Hamiltonian Cycle in hyperbolic unit disk graphs (which is polynomial) and our Hyperbolic TSP algorithm, which for constant α runs in $n^{O(\log^2(n))}$ time. Is there an $n^{O(\log n)}$ or a polynomial algorithm for $\alpha \geqslant 1$? Alternatively, can we prove a (conditional) superpolynomial lower bound? Such a lower bound would have to go beyond the quasi-polynomial lower bound for Independent Set seen in [18], as that relies heavily on dense point sets which are not allowed for $\alpha = \Omega(1)$. Another approach would be to use the naïve grid embedding of [18] directly, but that does not lead to a superpolynomial lower bound here.
- **Higher dimensions.** The grid-based lower-bound framework of [8] can be used in $\mathbb{H}^{d+1}$, see [18]. In particular, the ETH-based lower bound of [7] for Euclidean TSP implies that there is no $2^{o(n^{1-1/(d-1)})}$ algorithm for Hyperbolic TSP in $\mathbb{H}^d$ under ETH. Can we extend our algorithmic techniques to constant-spaced point sets in $\mathbb{H}^d$ and gain algorithms with running time $2^{n^{1-1/(d-1)}\,\mathrm{poly}(\log n)}$? What happens for denser point sets? As observed in [7], the techniques of Hwang *et al.* [12] do not even seem to extend to $\mathbb{R}^d$ for $d \geqslant 3$. Is a running time of $2^{n^{1-1/d}\,\mathrm{poly}(\log n)}$ possible for all point sets in $\mathbb{H}^d$?
- **A less forgiving parameter.** Our usage of the spacing parameter α may be too restrictive. Is there a better algorithm that can handle more general inputs that can contain a few close point pairs?

References

1 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. `doi:10.1145/290179.290180`.

2 Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM J. Comput.*, 45(4):1563–1581, 2016. `doi:10.1137/130913328`.

3 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, 1962. `doi:10.1145/321105.321111`.

4 Riccardo Benedetti and Carlo Petronio. *Lectures on hyperbolic geometry*. Springer Science & Business Media, 2012.

5 James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31:59–115, 1997.

6 Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

7 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, and Sudeshna Kolay. An ETH-tight exact algorithm for Euclidean TSP. In *Proceedings of FOCS 2018*, pages 450–461. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00050.

8 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for ETH-tight algorithms and lower bounds in geometric intersection graphs. In *Proceedings of STOC 2018*, pages 574–586, 2018. doi:10.1145/3188745.3188854.

9 Vladimir G. Deineko, Bettina Klinz, and Gerhard J. Woeginger. Exact algorithms for the Hamiltonian cycle problem in planar graphs. *Operations Research Letters*, 34(3):269–274, 2006. doi:10.1016/j.orl.2005.04.013.

10 Marvin J Greenberg. *Euclidean and non-Euclidean geometries: Development and history*. Macmillan, 1993.

11 Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM National Meeting*, ACM '61, pages 71.201–71.204, New York, NY, USA, 1961. ACM.

12 R. Z. Hwang, R. C. Chang, and Richard C. T. Lee. The searching over separators strategy to solve some NP-hard problems in subexponential time. *Algorithmica*, 9(4):398–423, 1993. doi:10.1007/BF01228511.

13 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

14 Shreesh Jadhav and Asish Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete & Computational Geometry*, 12:291–312, 1994. doi:10.1007/BF02574382.

15 Viggo Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.

16 Richard M. Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming*, pages 219–241. Springer, 2010.

17 Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *J. Comput. Syst. Sci.*, 81(8):1665–1677, 2015. doi:10.1016/j.jcss.2015.06.003.

18 Sándor Kisfaludi-Bak. Hyperbolic intersection graphs and (quasi)-polynomial time. In *Proceedings of SODA 2020*, pages 1621–1638. SIAM, 2020. doi:10.1137/1.9781611975994.100.

19 Sándor Kisfaludi-Bak. A quasi-polynomial algorithm for well-spaced hyperbolic TSP. *CoRR*, abs/2002.05414, 2020. arXiv:2002.05414.

20 Robert Krauthgamer and James R. Lee. Algorithms on negatively curved spaces. In *Proceedings of FOCS 2006*, pages 119–132, 2006. doi:10.1109/FOCS.2006.9.

21 Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.

22 Arlan Ramsay, Robert Davis Richtmyer, and Robert D. Richtmyer. *Introduction to hyperbolic geometry*. Universitext. Springer, New York, 1995.

23 Satish Rao and Warren D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 540–550. ACM, 1998. doi:10.1145/276698.276868.

24 Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *Proceedings of FOCS 2018*, pages 232–243. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743449.

25 Aleksandr S. Smogorževskij. *Lobatschewskische Geometrie*. Mathematische Schülerbücherei 96. Teubner, Leipzig, 1. aufl. edition, 1978.

26 William P. Thurston. *Three-Dimensional Geometry and Topology*, volume 1. Princeton University Press, 1997.