Euclidean TSP in Narrow Strips

Henk Alkema
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
h.y.alkema@tue.nl

Mark de Berg
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
m.t.d.berg@tue.nl

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

Abstract

We investigate how the complexity of Euclidean TSP for point sets P inside the strip $(-\infty, +\infty) \times [0, \delta]$ depends on the strip width δ. We obtain two main results.

- For the case where the points have distinct integer x-coordinates, we prove that a shortest bitonic tour (which can be computed in $O(n \log^2 n)$ time using an existing algorithm) is guaranteed to be a shortest tour overall when $\delta \leqslant 2\sqrt{2}$, a bound which is best possible.
- We present an algorithm that is fixed-parameter tractable with respect to δ. More precisely, our algorithm has running time $2^{O(\sqrt{\delta})} n^2$ for sparse point sets, where each $1 \times \delta$ rectangle inside the strip contains $O(1)$ points. For random point sets, where the points are chosen uniformly at random from the rectangle $[0, n] \times [0, \delta]$, it has an expected running time of $2^{O(\sqrt{\delta})} n^2 + O(n^3)$.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms

Keywords and phrases Computational geometry, Euclidean TSP, bitonic TSP, fixed-parameter tractable algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.4

Related Version A full version of the paper is available at `https://arxiv.org/abs/2003.09948`.

Funding The work in this paper is supported by the Netherlands Organisation for Scientific Research (NWO) through Gravitation-grant NETWORKS-024.002.003.

Acknowledgements We thank Remco van der Hofstad for discussions about the probabilistic analysis of an earlier version of the algorithm.

1 Introduction

In the Traveling Salesman Problem one is given an edge-weighted complete graph and the goal is to compute a tour – a simple cycle visiting all nodes – of minimum total weight. Due to its practical as well as theoretical importance, the Traveling Salesman Problem and its many variants are among the most famous problems in computer science and combinatorial optimization. In this paper we study the Euclidean version of the problem. In Euclidean TSP the input is a set P of n points in $\mathbb{R}^d$, and the goal is to compute a minimum-length tour visiting each point. Euclidean TSP in the plane was proven to be NP-hard in the 1970s [16, 21]. Around the same time, Christofides [4] gave an elegant (3/2)-approximation algorithm, which works in any metric space. For a long time it was unknown if Euclidean TSP is APX-hard, until Arora [2], and independently Mitchell [20], presented a PTAS. Mitchell's algorithm works for the planar case, while Arora's algorithm also works in higher dimensions. Rao and Smith [22] later improved the running time of Arora's PTAS, obtaining a running time of $2^{(1/\varepsilon)^{O(d)}} n + (1/\varepsilon)^{O(d)} n \log n$ in $\mathbb{R}^d$.

36th International Symposium on Computational Geometry (SoCG 2020).
Editors: Sergio Cabello and Danny Z. Chen; Article No. 4; pp. 4:1–4:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We are interested in exact algorithms for Euclidean TSP. As mentioned, the problem is already NP-hard in the plane. Unlike the general (metric) version, however, it can be solved in *subexponential* time, that is, in time $2^{o(n)}$. In particular, Kann [19] and Hwang et al. [17] presented algorithms with $n^{O(\sqrt{n})}$ running time. Smith and Wormald [26] gave a subexponential algorithm that works in any (fixed) dimension; its running time in $\mathbb{R}^d$ is $n^{O(n^{1-1/d})}$. Very recently De Berg et al. [8] improved this to $2^{O(n^{1-1/d})}$, which is tight up to constant factors in the exponent, under the Exponential-Time Hypothesis (ETH) [18].

There has also been considerable research on special cases of Euclidean TSP that are polynomial-time solvable. One example is Bitonic TSP, where the goal is to find a shortest *bitonic* tour. (A tour is bitonic if any vertical line crosses it at most twice; here the points from the input set P are assumed to have distinct x-coordinates.) It is a classic exercise [5] to prove that Bitonic TSP can be solved in $O(n^2)$ time by dynamic programming. De Berg et al. [9] showed how to speed up the algorithm to $O(n \log^2 n)$. When P is in convex position, then the convex hull of P is a shortest tour and so one can solve Euclidean TSP in $O(n \log n)$ time [10]. Deineko et al. [12] studied the case where the points need not all be on the convex hull; the points inside the convex hull, however, are required to be collinear. Their algorithm runs in $O(n^2)$ time. Deineko and Woeginger [13] extended this to the case where the points in the interior of the convex hull lie on k parallel lines, obtaining an $O(n^{k+2})$ algorithm. These results generalize earlier work by Cutler [6] and Rote [24] who consider point sets lying on three, respectively k, parallel lines. Deineko et al. [11] gave a fixed-parameter tractable algorithm for Euclidean TSP where the parameter k is the number of points inside the convex hull, with running time $O(2^k k^2 n)$. Finally, Reinhold [23] and Sanders [25] proved that when there exists a collection of disks centered at the points in P whose intersection graph is a single cycle – this is called the necklace condition – then the tour following the cycle is optimal. Edelsbrunner et al. [15] gave an $O(n^2 \log n)$ algorithm to verify if such a collection of disks exists (and, if so, find one).

Our contribution. The computational complexity of Euclidean TSP in $\mathbb{R}^d$ is $2^{\Theta(n^{1-1/d})}$ (for $d \geqslant 2$), assuming ETH. Thus the complexity depends heavily on the dimension d. This is most pronounced when we compare the complexity for $d = 2$ with the trivial case $d = 1$: in the plane Euclidean TSP takes $2^{\Theta(\sqrt{n})}$ time in the worst case, while the 1-dimensional case is trivially solved in $O(n \log n)$ time by just sorting the points. We study the complexity of Euclidean TSP for planar point sets that are "almost 1-dimensional". In particular, we assume that the point set P is contained in the strip $(-\infty, \infty) \times [0, \delta]$ for some relatively small δ and investigate how the complexity of Euclidean TSP depends on the parameter δ. As any instance of Euclidean TSP can be scaled to fit inside a strip, we need to make some additional restriction on the input. We consider three scenarios.

- *Integer x-coordinates.* Bitonic TSP can be solved in $O(n \log^2 n)$ time [9]. It is natural to conjecture that for points with distinct integer x-coordinates inside a sufficiently narrow strip, an optimal bitonic tour is a shortest tour overall. We give a (partially computer-assisted) proof that this is indeed the case: we prove that when $\delta \leqslant 2\sqrt{2}$ an optimal bitonic tour is optimal overall, and we show that the bound $2\sqrt{2}$ is best possible.
- *Sparse point sets.* We generalize the case of integer x-coordinate to the case where each rectangle $[x, x+1] \times [0, \delta]$ contains $O(1)$ points, and we investigate how the complexity of Euclidean TSP grows with δ. We show in the full version [1] that for sparse point sets an optimal tour must be k-tonic – a tour is k-tonic if it intersects any vertical line at most k-times – for $k = O(\sqrt{\delta})$. This suggests that one might be able to use a dynamic-programming algorithm similar to the ones for for points on k parallel lines [13, 24].

The latter algorithms run in $O(n^k)$ time, suggesting that a running time of $n^{O(\sqrt{\delta})}$ is achievable in our case. We give a much more efficient algorithm, which is fixed-parameter tractable (and subexponential) with respect to the parameter δ. Its running time is $2^{O(\sqrt{\delta})}n^2$.

- *Random point sets.* In the third scenario the points in P are drawn independently and uniformly at random from the rectangle $R := [0, n] \times [0, \delta]$. For this case we prove that the same algorithm as for sparse point sets has a (now expected) running time of $2^{O(\sqrt{\delta})}n^2 + O(n^3)$.

Notation and terminology. Let $P := \{p_1, \ldots, p_n\}$ be a set of points in a horizontal strip of width δ – we call such a strip a *δ-strip* – which we assume without loss of generality to be the strip $(-\infty, \infty) \times [0, \delta]$. We denote the x-coordinate of a point $p \in \mathbb{R}^2$ by $x(p)$, and its y-coordinate by $y(p)$. To simplify the notation, we also write x_i for $x(p_i)$, and y_i for $y(p_i)$. We sort the points in P such that $0 \leqslant x_i \leqslant x_{i+1}$ for all $1 \leqslant i < n$.

For two points $p, q \in \mathbb{R}^2$, we write pq to denote the *directed* edge from p to q. Paths are written as lists of points, so $(q_1, q_2, ..., q_m)$ denotes the path consisting of the edges $q_1q_2, \ldots, q_{m-1}q_m$. All points in a path must be distinct, except possibly $q_1 = q_m$ in which case the path is a tour. The length of an edge pq is denoted by $|pq|$, and the total length of a set E of edges is denoted by $\|E\|$.

A *separator* is a vertical line not containing any of the points in P that separates P into two non-empty subsets.

2 Bitonicity for points with integer x-coordinates

In this section we consider the case where the points in P have distinct integer x-coordinates. For our purposes, two separators s, s' that induce the same partitioning of P are equivalent. Therefore, we can define $\mathcal{S} := \{s_1, \ldots, s_{n-1}\}$ as the set of all combinatorially distinct separators, obtained by taking one separator between any two points p_i, p_{i+1}. Let E be a set of edges with endpoints in P. The *tonicity of E at a separator s*, written as $\text{ton}(E, s)$, is the number of edges in E crossing s. We say that a set E has *lower tonicity* than a set F of edges, denoted by $E \preccurlyeq F$, if $\text{ton}(E, s_i) \leqslant \text{ton}(F, s_i)$ for all $s_i \in \mathcal{S}$. The set E has *strictly lower tonicity*, denoted by $E \prec F$, if there also exists at least one i for which $\text{ton}(E, s_i) < \text{ton}(F, s_i)$. Finally, we call a set E of edges *k-tonic* – or *monotonic* when $k = 1$, and *bitonic* when $k = 2$ – if $\text{ton}(E, s_i) \leqslant k$ for all $s_i \in \mathcal{S}$.

The goal of this section is to prove the following theorem.

▶ **Theorem 1.** *Let P be a set of points with distinct and integer x-coordinates in a δ-strip. When $\delta \leqslant 2\sqrt{2}$, a shortest bitonic tour on P is a shortest tour overall. Moreover, for any $\delta > 2\sqrt{2}$ there is a point set P in a δ-strip such that a shortest bitonic tour on P is not a shortest tour overall.*

The construction for the case $\delta > 2\sqrt{2}$ is shown in Fig. 1. It is easily verified that, up to symmetrical solutions, the tours T_1 and T_2 are the only candidates for the shortest tour. Observe that $\|T_2\| - \|T_1\| = |p_1p_4| - |p_4p_5| = 3 - \sqrt{1+\delta^2}$. Hence, for $\delta > 2\sqrt{2}$ we have $\|T_2\| < \|T_1\|$, which proves the lower bound of Theorem 1. The remainder of the section is devoted to proving the first statement.

Let P be a point set in a δ-strip for $\delta = 2\sqrt{2}$, where all points in P have distinct integer x-coordinates. Among all shortest tours on P, let T_{opt} be one that is minimal with respect to the $\preccurlyeq$-relation; T_{opt} exists since the number of different tours on P is finite. We claim that T_{opt} is bitonic, proving the upper bound of Theorem 1.

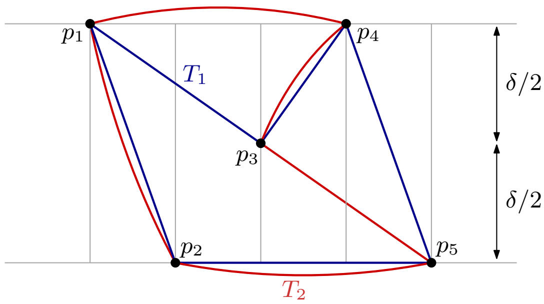


Figure 1 Construction for $\delta > 2\sqrt{2}$ for Theorem 1. The grey vertical segments are at distance 1 from each other. If $\delta > 2\sqrt{2}$ then T_1, the shortest bitonic tour (in blue), is longer than T_2, the shortest non-bitonic tour (in red).

Suppose for a contradiction that T_{opt} is not bitonic. Let $s^* \in \mathcal{S}$ be the rightmost separator for which $\text{ton}(T_{\text{opt}}, s^*) > 2$. We must have $\text{ton}(T_{\text{opt}}, s^*) = 4$ because otherwise $\text{ton}(T_{\text{opt}}, s) > 2$ for the separator $s \in \mathcal{S}$ immediately to the right of s^*, since there is only one point from P between s^* and s. Let F be the four edges of T_{opt} crossing s^*, and let E be the remaining set of edges of T_{opt}. Let Q be the set of endpoints of the edges in F. We will argue that there exists a set F' of edges with endpoints in Q such that $E \cup F'$ is a tour and (i) $\|F'\| < \|F\|$, or (ii) $\|F'\| = \|F\|$ and $F' \prec F$. We will call such an F' *superior to* F. Option (i) contradicts that T_{opt} is a shortest tour, and (ii) contradicts that T_{opt} is a shortest tour that is minimal with respect to $\preccurlyeq$ (since $E \cup F' \prec E \cup F$ if and only if $F' \prec F$). Hence, proving that such a set F' exists finishes the proof.

The remainder of the proof proceeds in two steps. In the first step we move the points in Q to obtain a set $\overline{Q}$ with consecutive integer coordinates, in such a way that there exists an edge set $\overline{F}$ on $\overline{Q}$ such that if an $\overline{F}'$ superior to $\overline{F}$ exists, then there also exists an F' superior to F. In the second step we then give a computer-assisted proof that the desired set $\overline{F}'$ exists.

Step 1: Finding a suitable $\overline{Q}$ with consecutive x-coordinates. Let T_{opt}, s^*, E, F and Q be defined as above. We assume without loss of generality that the x-coordinate of s^* is equal to $x^* + \frac{1}{2}$, where x^* is the largest integer such that the line $x = x^* + \frac{1}{2}$ intersects all four edges in F. Since the actual edges in E are not important for our arguments, we replace them by abstract "connections" specifying which pairs of endpoints of the edges in F are connected by paths of edges in E. It will be convenient to duplicate the points in Q that are shared endpoints of two edges in F, and add a connection between the two copies; see Fig. 2. We denote the set of connections obtained in this way by $\widetilde{E}$, and we call $\widetilde{E}$ the *connectivity pattern* of F (in $E \cup F$).

Next we show how to move the points in Q such that the modified set $\overline{Q}$ uses consecutive x-coordinates. Recall that $s^* : x = x^* + \frac{1}{2}$ is a separator that intersects all edges in F. Let Q_{left} and Q_{right} be the subsets of points from Q lying to the left and right of s^*, respectively. We will move the points in Q_{left} such that they will get consecutive x-coordinates with the largest one being equal to x^*, while the points in Q_{right} will get consecutive x-coordinates with the smallest one being $x^* + 1$.

We move the points in Q_{left} as follows. Let $z \leqslant x^*$ be the largest x-coordinate currently not in use by any of the points in Q_{left}. If Q_{left} lies completely to the right of the line $\ell(z) : x = z$, then we are done: the set of x-coordinates used by points in Q_{left} is $\{z+1, \ldots, x^*\}$. Otherwise we take an arbitrary edge $e \in F$ that crosses $\ell(z)$, and we move its left endpoint to the point $e \cap \ell(z)$; see Fig. 3(i). This process is repeated until Q_{left} uses consecutive x-coordinates.

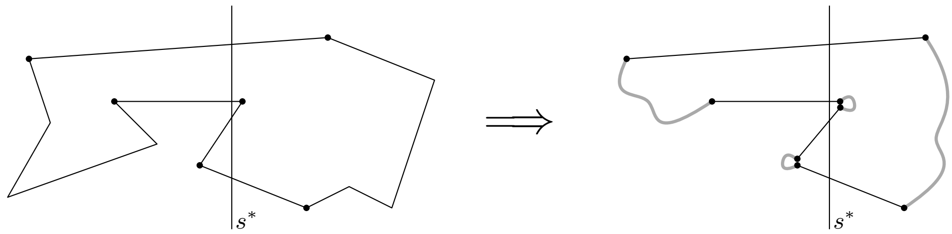


Figure 2 Replacing the paths connecting endpoints of edges in F by abstract connections. The copies of duplicated shared endpoints are slightly displaced in the figure to be able to distinguish them, but they are actually coinciding.

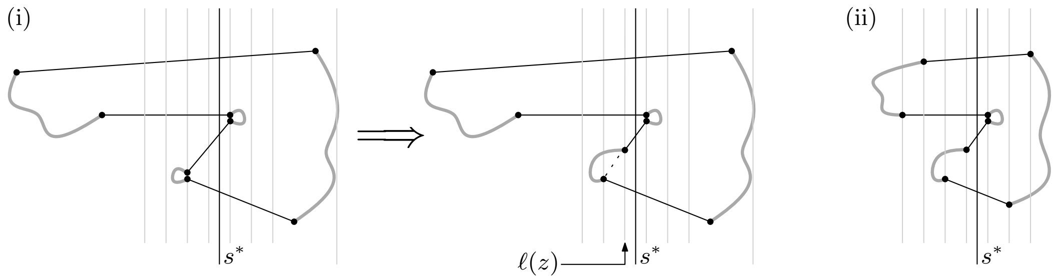


Figure 3 The process of moving the points in Q. Grey vertical lines have integer x-coordinates. (i) Moving a point in Q_{left} so that it gets x-coordinate z. (ii) A possible configuration after Q_{left} and Q_{right} have been treated.

After moving the points in Q_{left} we treat Q_{right} in a similar manner; the only difference is that now we define $z > x^*$ to be the smallest x-coordinate currently not in use by any of the points in Q_{right}. Fig. 3(ii) shows the final result for the example in part (i) of the figure.

Before we prove that this procedure preserves the desired properties, two remarks are in order about the process described above. First, in each iteration we may have different choices for the edge e crossing $\ell(z)$, and the final result depends on these choices. Second, when we move a point in Q to a new location, then the new x-coordinate is not used by Q but it may already be used by points in $P \setminus Q$. Neither of these facts cause any problems for the coming arguments.

Let $\overline{Q}$ be the set of points from Q after they have been moved to their new locations, and let $\overline{F}$ be the set of edges from F after the move. With a slight abuse of notation we still use $\widetilde{E}$ to specify the connectivity pattern on $\overline{F}$, which is simply carried over from F. The following lemma shows that we can use $\overline{F}$, $\overline{Q}$ and $\widetilde{E}$ in Step 2 of the proof.

▶ **Lemma 2.** *Let $E, F, Q, \widetilde{E}, \overline{F}, \overline{Q}$ be defined as above. Let $\overline{F}'$ be any set of edges (with endpoints in $\overline{Q}$) superior to $\overline{F}$, such that $\widetilde{E} \cup \overline{F}'$ is a tour. Then there is a set of edges F' (with endpoints in Q) superior to F such that $E \cup F'$ is a tour.*

Proof. We define F' in the obvious way, by simply taking $\overline{F}'$ and replacing each endpoint (which is a point in $\overline{Q}$) by the corresponding point in Q. Clearly $E \cup F'$ forms a tour if $\widetilde{E} \cup \overline{F}'$ forms a tour.

Suppose $\overline{F}'$ is superior to $\overline{F}$. We will now show that F' is superior to F. We will show this by first proving that $\|F\| - \|F'\| \geqslant \|\overline{F}\| - \|\overline{F}'\|$, and then proving that for any separator we have $s \in \mathcal{S}$, $\text{ton}(F, s) - \text{ton}(F', s) = \text{ton}(\overline{F}, s) - \text{ton}(\overline{F}', s)$.

Recall that each edge $\overline{e} \in \overline{F}$ is obtained from the corresponding edge $e \in F$ by moving one or both endpoints along the edge e itself. Also recall that we duplicated shared endpoints of edges in F, so if we move a point in Q we move the endpoint of a single edge in F. Hence,

$$\|F\| - \|\overline{F}\| = \text{total distance over which the points in } Q \text{ are moved.}$$

Since we added a connection to $\widetilde{E}$ between the two copies of a shared endpoint, each point in Q is incident to exactly one connection in $\widetilde{E}$ and, hence, to exactly one edge in F'. This means that if we move a point in Q we move the endpoint of a single edge in F', so

$$\|F'\| - \|\overline{F}'\| \leqslant \text{total distance over which the points in } Q \text{ are moved.}$$

We conclude that $\|F\| - \|F'\| \geqslant \|\overline{F}\| - \|\overline{F}'\|$.

Next, consider any separator $s \in \mathcal{S}$ to the left of s^*. Since points in Q_{left} are only moved towards s^* we know that

$$\text{ton}(F, s) - \text{ton}(\overline{F}, s) = (\text{number of points in } Q_{\text{left}} \text{ moved across } s) = \text{ton}(F', s) - \text{ton}(\overline{F}', s).$$

A similar argument shows that $\text{ton}(F, s) - \text{ton}(F', s) = \text{ton}(\overline{F}, s) - \text{ton}(\overline{F}', s)$ for any separator s to the right of s^*.

Now, since we have proven that $\|F\| - \|F'\| \geqslant \|\overline{F}\| - \|\overline{F}'\|$ and that for any separator we have $s \in \mathcal{S}$, $\text{ton}(F, s) - \text{ton}(F', s) = \text{ton}(\overline{F}, s) - \text{ton}(\overline{F}', s)$, we can conclude that if $\overline{F}'$ is superior to $\overline{F}$, then F' is superior to F. ◀

Step 2: Finding the set F'. The goal of Step 2 of the proof is the following: given the tour $T_{\text{opt}} = E \cup F$ inside a δ-strip of width $\delta = 2\sqrt{2}$, show that there exists a set F' of edges such that $E \cup F'$ is a tour and F' is superior to F. Lemma 2 implies that we may work with $\widetilde{E}$ and $\overline{F}$ instead of E and F (and then find $\overline{F}'$ instead of F').

In Step 1 we duplicated shared endpoints of edges in E. We now merge these two copies again if they are still at the same location. This will always be the case for the shared endpoint immediately to the right of the separator s^*, since we picked $s^* : x = x^* + \frac{1}{2}$ such that there is a shared endpoint at $x = x^* + 1$ and the copies of this endpoint will not be moved. So if n_{left} and n_{right} denote the number of distinct endpoints to the right and left of s^*, respectively, then $n_{\text{right}} \in \{2, 3\}$ and $n_{\text{left}} \in \{2, 3, 4\}$. We thus have six cases in total for the pair $(n_{\text{left}}, n_{\text{right}})$, as depicted in Fig. 4. Each of the six cases has several subcases, depending on the left-to-right order of the vertices inside the gray rectangles in the figure. Once we fixed the ordering, we can still vary the y-coordinates in the range $[0, \delta]$, which may lead to scenarios where different sets $\overline{F}'$ are required. We handle this potentially huge amount of cases in a computer-assisted manner, using an automated prover *FindShorterTour*$(n_{\text{left}}, n_{\text{right}}, \overline{F}, \widetilde{E}, X, \delta, \varepsilon)$. The input parameter X is an array where $X[i]$ specifies the set from which the x-coordinate of the i-th point in the given scenario may be chosen, where we assume w.l.o.g. that $x(s^*) = -1/2$; see Fig. 4. The role of the parameter ε will be explained below.

The output of *FindShorterTour* is a list of *scenarios* and an *outcome* for each scenario. A scenario contains for each point q an x-coordinate $x(q)$ from the set of allowed x-coordinates for q, and a range $y\text{-range}(q) \subseteq [0, 2\sqrt{2}]$ for its y-coordinate, where the y-range is an interval of length at most ε. The outcome is either Success or Fail. Success means that a set $\overline{F}'$ has been found with the desired properties: $\widetilde{E} \cup \overline{F}'$ is a tour, and for all possible instantiations of the scenario – that is, all choices of y-coordinates from the y-ranges in the scenario – we have $\|\overline{F}'\| < \|\overline{F}\|$. Fail means that such an $\overline{F'}$ has not been found, but it does not guarantee that such an $\overline{F'}$ does not exist for this scenario. The list of scenarios is complete in the sense that for any instantiation of the input case there is a scenario that covers it.

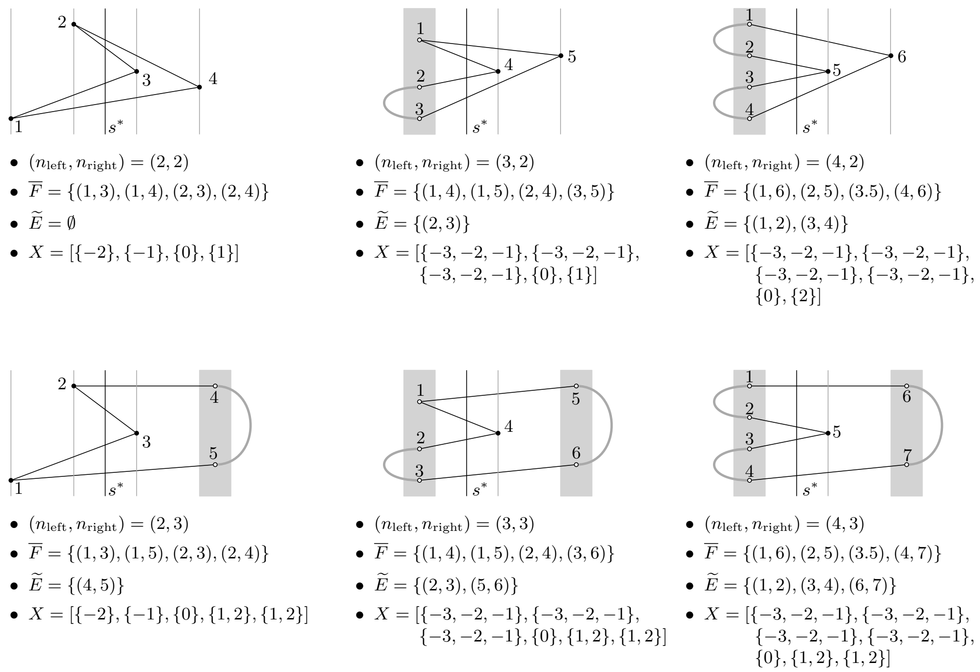


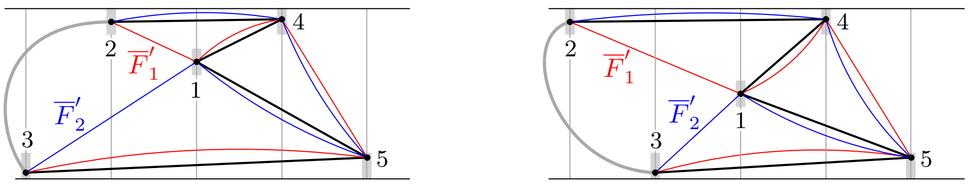
Figure 4 The six different cases that result after applying Step 1 of the proof. Points indicated by filled disks have a fixed x-coordinate. The left-to-right order of points drawn inside a grey rectangle, on the other hand, is not known yet. The vertical order of the edges is also not fixed, as the points can have any y-coordinate in the range $[0, 2\sqrt{2}]$.

FindShorterTour works brute-force, by checking all possible combinations of x-coordinates and subdividing the y-coordinate ranges until a suitable $\overline{F}'$ can be found or until the y-ranges have length at most ε. The implementation details of the procedure are the full version [1].

Note that case $(n_{\text{left}}, n_{\text{right}}) = (2, 3)$ in Fig. 4 is a subcase of case $(n_{\text{left}}, n_{\text{right}}) = (3, 2)$, if we exchange the roles of the points lying to the left and to the right of s^*. Hence, we ignore this subcase and run our automated prover on the remaining five cases, where we set $\varepsilon := 0.001$. It successfully proves the existence of a suitable set $\overline{F}'$ in four cases; the case where the prover fails is the case $(n_{\text{left}}, n_{\text{right}}) = (3, 2)$. For this case it fails for the two scenarios depicted in Fig. 5; all other scenarios for these cases are handled successfully (up to symmetries). For both scenarios we consider two alternatives for the set $\overline{F}'$: the set $\overline{F}'_1$ shown in red in Fig. 5, and the set $\overline{F}'_2$ shown in blue in Fig. 5. We will show that in any instantiation of both scenarios, either $\overline{F}'_1$ or $\overline{F}'_2$ is at least as short as $\overline{F}$; since both alternatives are bitonic this finishes the proof.

For $1 \leqslant i \leqslant 5$, let q_i be the point labeled i in Fig. 5. We first argue that (for both scenarios) we can assume without loss of generality that $y(q_2) = y(q_4) = 2\sqrt{2}$ and $y(q_3) = y(q_5) = 0$. To this end, consider arbitrary instantiations of these scenarios, and imagine moving q_2 and q_4 up to the line $y = 2\sqrt{2}$, and moving q_3 and q_5 down to the line $y = 0$. It suffices to show, for $i \in \{1, 2\}$, that if we have $\|\overline{F}'_i\| \leqslant \|\overline{F}\|$ after the move, then we also have $\|\overline{F}_i\| \leqslant \|\overline{F}\|$ before the move. This can easily be proven by repeatedly applying the following observation.

▶ **Observation 3.** *Let a, b, c be three points. Let ℓ be the vertical line through c, and let us move c downwards along ℓ. Let α be the smaller angle between ac and ℓ if $y(c) < y(a)$, and the larger angle otherwise, and let β be the smaller angle between bc and ℓ if $y(c) < y(b)$, and the larger angle otherwise, and suppose $\alpha < \beta$ throughout the move. Then the move increases $|ac|$ more than it increases $|bc|$.*



- $X = [\{-1\}, \{-2\}, \{-3, \}, \{0\}, \{1\}]$
- $Y = [[1.61, 1.62], [2.82, 2\sqrt{2}],$
  $[0, 0.01], [2.82, 2\sqrt{2}], [0, 0.01]]$

- $X = [\{-1\}, \{-3\}, \{-2, \}, \{0\}, \{1\}]$
- $Y = [[1.41, 1.42], [2.82, 2\sqrt{2}],$
  $[0, 0.01], [2.82, 2\sqrt{2}], [0, 0.01]]$

Figure 5 Two scenarios covering all subscenarios where the automated prover fails. Each point has a fixed x-coordinate and a y-range specified by the array Y; the resulting possible locations are shown as small grey rectangles (drawn larger than they actually are for visibility). For all subscenarios, at least one of $\overline{F}'_1$ (in red) and $\overline{F}'_2$ (in blue) is at most as long as $\overline{F}$ (in black).

So now assume $y(q_2) = y(q_4) = 2\sqrt{2}$ and $y(q_3) = y(q_5) = 0$. Consider the left scenario in Fig. 5, and let $y := y(q_3)$. If $y \geqslant (8\sqrt{2})/7$ then

$$|q_2q_1| + |q_4q_5| = \sqrt{1 + (2\sqrt{2} - y)^2} + 3 \leqslant 2 + \sqrt{4 + y^2} = |q_2q_4| + |q_1q_5|,$$

so $\|\overline{F}'_1\| \leqslant \|\overline{F}\|$. On the other hand, If $y \leqslant (8\sqrt{2})/7$ then

$$|q_3q_1| + |q_4q_5| = \sqrt{4 + y^2} + 3 \leqslant \sqrt{1 + (2\sqrt{2} - y)^2} + 4 = |q_1q_4| + |q_3q_5|,$$

so $\|\overline{F}'_2\| \leqslant \|\overline{F}\|$. So either $\overline{F}'_1$ or $\overline{F}'_2$ is at least as short as $\overline{F}$, finishing the proof for the left scenario in Fig. 5. The proof for the right scenario in Fig. 5 is analogous, with cases $y \geqslant \sqrt{2}$ and $y \leqslant \sqrt{2}$. This finishes the proof for the right scenario and, hence, for Theorem 1.

3 An algorithm for narrow strips

In this section we investigate how the complexity of Euclidean TSP depends on the width δ of the strip containing the point set P. Recall that a point set P inside a δ-strip is *sparse* if for every $x \in \mathbb{R}$ the rectangle $[x, x+1] \times [0, \delta]$ contains $O(1)$ points.

▶ **Theorem 4.** *Let P be a set of n points in δ-strip.*
(i) *If for any $i \in \mathbb{Z}$ the square $[(i-1)\delta, i\delta] \times [0, \delta]$ contains at most k points, then we can solve Euclidean TSP on P in $2^{O(\sqrt{k})}n^2$ time.*
(ii) *If P is sparse then we can solve Euclidean TSP in $2^{O(\sqrt{\delta})}n^2$ time.*

Part (ii) of the theorem is a trivial consequence of part (i), so the rest of the section focuses on proving part (i). Our proof uses and modifies some techniques of [8]. For $i \in \mathbb{Z}$, let σ_i be the square $[(i-1)\delta, i\delta] \times [0, \delta]$. Define $n_i := |\sigma_i \cap P|$ – we assume without loss of generality that all points from P lie in the interior of a square σ_i – and let $k := \max_i n_i$. We say that a square σ_i is *empty* if $n_i = 0$.

We will regularly use that any subset E of edges from an optimal tour of a planar point set P has the *Packing Property* [8]: for any $t > 0$ and any square σ of side length t, the number of edges from E of length at least $t/4$ that intersect σ is $O(1)$. The Packing Property is at the heart of several subexponential algorithms [19, 26]. We also need the following lemma, which is essentially a special case of a recent result by De Berg et al. [8, Theorem 5]; see the full version [1] for a proof.

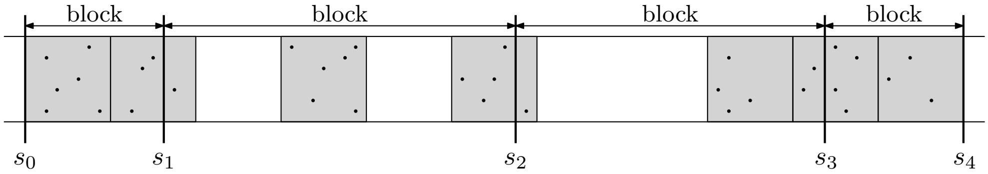


Figure 6 The separators $s_0, \ldots, s_{t+1}$ and the blocks they define.

▶ **Lemma 5.** *Let σ_i be a square as defined above. Then we can compute in $O(k^3)$ time a separator s intersecting σ_i such that the following holds. Let T_{opt} be an optimal TSP tour on P. For a separator s, let $T(s, \sigma_i)$ denote the set of edges from T_{opt} with both endpoints in $\sigma_{i-1} \cup \sigma_i \cup \sigma_{i+1}$ and crossing s. Then $|T(s, \sigma_i)| = O(\sqrt{k})$. Furthermore, there is a family $\mathcal{C}$ of $2^{O(\sqrt{k})}$ sets, which we call* candidate sets*, such that $T(s, \sigma_i) \in \mathcal{C}$, and this family can be computed in $2^{O(\sqrt{k})}$ time.*

Separators and blocks. Consider the sequence of non-empty squares σ_i, ordered from left to right. We use Lemma 5 to place a separator in every second block of this sequence. Let $\mathcal{S} := \{s_1, \ldots, s_t\}$ be the resulting (ordered) set of separators, and let s_0 and s_{t+1} denote separators coinciding with the left side of the leftmost non-empty square and the right side of the rightmost non-empty square, respectively; see Fig. 6. We call the region of the δ-strip between two consecutive separators s_{j-1} and s_j a *block*. Let $P_j \subseteq P$ denote the set of points in this block. Note that $|P_j| \leqslant 3k$.

For an edge set E and a separator s, let $E(s) \subseteq E$ denote the subset of edges intersecting s, and define $P_{\mathrm{right}}(E, s)$ to be the set of endpoints of the edges in $E(s)$ that lie on or to the right of s. We call $P_{\mathrm{right}}(E, s)$ the *endpoint configuration of E at s*. The next two lemmas rule out endpoint configurations with two "distant" points from the separator.

▶ **Lemma 6.** *Let $s_{\mathrm{left}} : x = x_{\mathrm{left}}$ and $s_{\mathrm{right}} : x = x_{\mathrm{right}}$ be two separators such that $x_{\mathrm{right}} - x_{\mathrm{left}} > 3\delta$, and suppose there is a point $z \in P$ with $x_{\mathrm{left}} + 3\delta/2 < x(z) < x_{\mathrm{right}} - 3\delta/2$. Then an optimal tour on P cannot have two edges that both cross s_{left} and s_{right}.*

Proof. Suppose for a contradiction that an optimal tour T has two directed edges, q_1q_2 and r_1r_2, that both cross s_{left} and s_{right}. (The direction of q_1q_2 and r_1r_2 is according to a fixed traversal of the tour.) If both edges cross s_{left} and s_{right} from left to right (or both cross from right to left) then replacing q_1q_2 and r_1r_2 by q_1r_1 and r_2q_2 gives a shorter tour – see Observation 14 in the full version [1] – leading to the desired contradiction.

Now suppose that q_1q_2 and r_1r_2 cross s_{left} and s_{right} in opposite directions. Assume w.l.o.g. that $x(q_1) < x_{\mathrm{left}}$ and $x(r_2) < x_{\mathrm{left}}$, and that z lies on the path from r_2 to q_1. Let $u_1, \ldots, u_k$, $v_1, \ldots, v_l$, and $w_1, \ldots, w_m$ be such that

$$T = (q_1, q_2, u_1, \ldots, u_k, r_1, r_2, v_1, \ldots, v_l, z, z_2, w_1, \ldots, w_m, q_1).$$

We claim that the tour T' defined as

$$T' = (q_1, r_2, v_1, \ldots, v_l, z, r_1, u_k, \ldots, u_1, q_2, z_2, w_1, \ldots, w_m, q_1)$$

is a strictly shorter tour. To show this, we will first change our point set P into a point set P' such that if $\|T'\| < \|T\|$ on P', then $\|T'\| < \|T\|$ also on P. To this end we replace q_1 by $q_1' := q_1q_2 \cap s_{\mathrm{left}}$ and q_2 by $q_2' := q_1q_2 \cap s_{\mathrm{right}}$, and we replace r_1 by $r_1' := r_1r_2 \cap s_{\mathrm{left}}$ and r_2 by $r_2' := r_1r_2 \cap s_{\mathrm{right}}$; see Fig. 7.

Finally, we replace z_2 by a point z'_2 coinciding with z (note that if $z_2 = q_1$, we can split it before moving the resulting two points, analogous to the proof of Theorem 1). Using a similar reasoning as in the proof of Lemma 2, one can argue that the point set $P' := (P \setminus \{q_1, q_2, r_1, r_2, z_2\}) \cup \{q'_1, q'_2, r'_1, r'_2, z'_2\}$ has the required property. To get the desired contradiction it thus suffices to show that $\|T\| - \|T'\| > 0$ on P'. This is true because

$$\begin{aligned} \|T\| - \|T'\| &= |q'_1 q'_2| + |r'_1 r'_2| + |z z'_2| - |q'_1 r'_2| - |z r'_1| - |q'_2 z'_2| \\ &\geqslant |x_{\text{left}} - x_{\text{right}}| + |x_{\text{right}} - x_{\text{left}}| + 0 \\ &\quad -\delta - (|x(z) - x_{\text{right}}| + \delta) - (|x_{\text{right}} - x(z)| + \delta) \\ &= 2(x(t) - x_{\text{left}}) - 3\delta > 0, \end{aligned}$$

where the last line uses that $x_{\text{left}} + 3\delta/2 < x(z)$. ◀

▶ **Lemma 7.** *Let $s_j \in \mathcal{S}$ be a separator, and let $\sigma_{j^*} = [(j^* - 1)\delta), j^*\delta] \times [0, \delta]$ denote the square in which it is placed. Let T_{opt} be an optimal tour on P and let $V := P_{\text{right}}(T_{\text{opt}}, s_j)$ be its endpoint configuration at s_j. Let P' denote the set of input points with x-coordinates between $(j^* + 1)\delta$ and $x(s_{j+3})$, and let P'' be the set of input points with x-coordinate larger than $x(s_{j+3})$. Then (i) $|P' \cap V| \leqslant c^*$ for some absolute constant c^*, and (ii) $|P'' \cap V| \leqslant 1$.*

Proof. Let uv be a tour edge crossing s_j. By definition of s_j, the number of edges crossing s_j with both endpoints in $\sigma_{j^*-1} \cup \sigma_{j^*} \cup \sigma_{j^*+1}$ is $O(\sqrt{k})$. Any other edge crossing s_j must fully cross σ_{j^*-1} or σ_{j^*+1} (or both), see Fig. 8. Therefore such edges have length at least δ. By the Packing Property, there can be at most $c^* = O(1)$ such edges. This proves (i).

To prove (ii), note there are five non-empty squares between s_j and s_{j+3}. Hence, there is a non-empty square between s_j and s_{j+3} with distance at least 2δ from s_j and s_{j+3}. Lemma 6 thus implies that T_{opt} has at most one edge crossing both s_j and s_{j+3}, proving (ii). ◀

Putting Lemma 5 and Lemma 7 together, we get the following corollary.

▶ **Corollary 8.** *Let T_{opt} be an optimal tour, let $s_j \in \mathcal{S}$ be a separator, and let $V \subset P$ be the endpoint configuration of T_{opt} at s_j. Then we can enumerate in $2^{O(\sqrt{k})} \cdot n$ time a family $\mathcal{B}_j$ of candidate endpoint sets such that $V \in \mathcal{B}_j$.*

In addition to the sets $\mathcal{B}_j$ $(j = 1, \ldots, t)$, we define $\mathcal{B}_0 = \mathcal{B}_{t+1} := \emptyset$.

Matchings, the rank-based approach, and representative sets. When we cut a tour using a vertical separator line, the tour falls apart into several paths. As in other TSP algorithms, we need to make sure that the paths on each side of the separator can be patched up into a

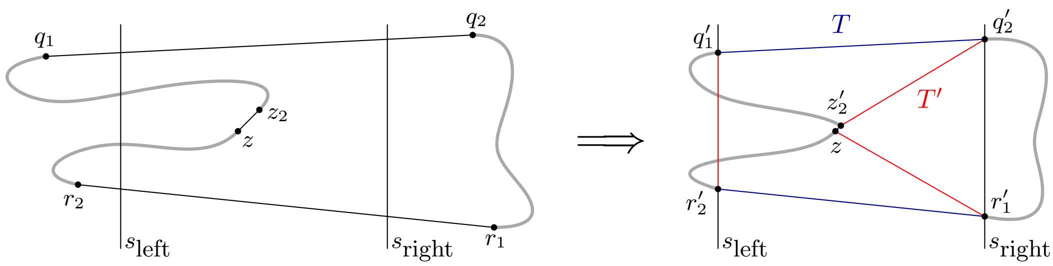


Figure 7 Illustration for the proof of Lemma 6. The point z'_2 coincides with z but is slightly displaced for visibility. The sum of the length of the edges unique to T (displayed in blue) is strictly larger than the sum of the length of the edges unique to T' (displayed in red).

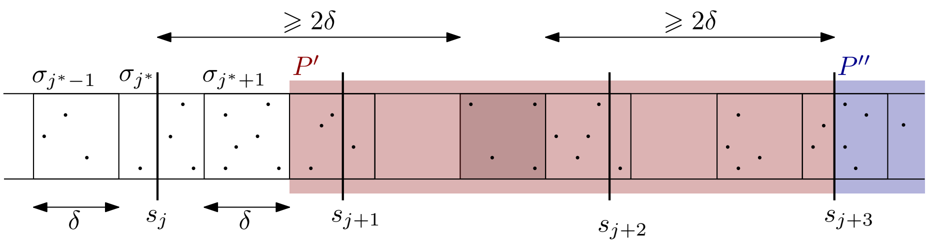


Figure 8 Tour edges crossing s_j and $x = (j^* + 1)\delta$ satisfy the Packing Property. Tour edges crossing s_j and s_{j+3} obey Lemma 6. The points in the red area form P'. The points in the blue area form P''.

Hamiltonian cycle. Following the terminology of [8], let P be our input point set, and let M be a perfect matching on a set $B \subseteq P$, where the points of B are called *boundary points*. A collection $\mathcal{P} = \{\pi_1, \ldots, \pi_{|B|/2}\}$ of paths *realizes M on P* if (i) for each edge $(p, q) \in M$ 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 total length of $\mathcal{P}$ as the length of the edges in its paths. In general, the type of problem that needs to be solved on one side of a separator is called Euclidean Path Cover. The input to such a problem is a point set $P \subset \mathbb{R}^2$, a set of boundary points $B \subseteq P$, and a perfect matching M on B. The task is to find a collection of paths of minimum total length that realizes M on P.

To get the claimed running time, we need to avoid iterating over all matchings. We can do this with the so-called *rank-based approach* [3, 7]. As our scenario is very similar to the general Euclidean TSP, we can reuse most definitions and some proof ideas from [8].

Let $\mathcal{M}(B)$ denote the set of all perfect matchings on B, and consider a matching $M \in \mathcal{M}(B)$. We can turn M into a weighted matching by assigning to it the minimum total length of any collection of paths realizing M. In other words, weight(M) is the length of the solution of Euclidean Path Cover for input (P, B, M). We use $\mathcal{M}(P, B)$ to denote the set of all such weighted matchings on B. Note that $|\mathcal{M}(P, B)| = |\mathcal{M}(B)| = 2^{O(|B| \log |B|)}$.

We say that two matchings $M, M' \in \mathcal{M}(B)$ *fit* if their union is a Hamiltonian cycle. Consider a pair P, B. Let $\mathcal{R}$ be a set of weighted matchings on B and let M be another matching on B. We define $\text{opt}(M, \mathcal{R}) := \min\{\text{weight}(M') : M' \in \mathcal{R}, M' \text{ fits } M\}$, that is, $\text{opt}(M, \mathcal{R})$ is the minimum total length of any collection of paths on P that together with the matching M forms a cycle. A set $\mathcal{R} \subseteq \mathcal{M}(P, B)$ of weighted matchings is defined to be *representative* of another set $\mathcal{R}' \subseteq \mathcal{M}(P, B)$ if for any matching $M \in \mathcal{M}(B)$ we have $\text{opt}(M, \mathcal{R}) = \text{opt}(M, \mathcal{R}')$. Note that our algorithm is not able to compute a representative set of $\mathcal{M}(P, B)$, because it is also restricted by the Packing Property and Lemma 6, while a solution of Euclidean Path Cover for a generic P, B, M may not satisfy them. Let $\mathcal{M}^*(P, B)$ denote the set of weighted matchings in $\mathcal{M}(P, B)$ that have a corresponding Euclidean Path Cover solution satisfying the Packing Property and Lemma 6.

The basis of the rank-based method is the following result.

▶ **Lemma 9** (Bodlaender et al. [3], Theorem 3.7)**.** *There exists a set $\overline{\mathcal{R}}$ consisting of $2^{|B|-1}$ weighted matchings that is representative of the set $\mathcal{M}(P, B)$. Moreover, there is an algorithm* Reduce *that, given a representative set $\mathcal{R}$ of $\mathcal{M}(P, B)$, computes such a set $\overline{\mathcal{R}}$ in $|\mathcal{R}| \cdot 2^{O(|B|)}$ time.*

Lemma 9 can also be applied for our case, where $\mathcal{R}$ is representative of $\mathcal{M}^*(P, B) \subseteq \mathcal{M}(P, B)$, the set of weighted matchings in $\mathcal{M}(P, B)$ that have a corresponding Euclidean Path Cover solution satisfying the Packing Property and Lemma 6.

We say that perfect matchings M on B and M' on B' are *compatible* if their union on $B \cup B'$ is either a single cycle or a collection of paths. The *join* of these matchings, denoted by $\text{Join}(M, M')$ is a perfect matching on the symmetric difference $B \triangle B'$ obtained by iteratively contracting edges with an incident vertex of degree 2 in the graph $(B \cup B', M \cup M')$.

The algorithm. Our algorithm is a dynamic program, where we define a subproblem for each separator index j, and each set of endpoints $B \in \mathcal{B}_j$. The value of $A[j, B]$ will be a representative set containing pairs (M, x), where M is a perfect matching on B and x is a real number equal to the total length of the path cover of $P_1 \cup \dots \cup P_j \cup B$ realizing the matching M. The length of the entire tour will be the value corresponding to the empty matching at index $t + 1$, that is, it will be the value x such that $A[t + 1, \emptyset] = \{(\emptyset, x)\}$.

Our dynamic-programming algorithm works on a block-by-block basis (which explains the parameter j) and it solves subproblems inside a block using the algorithm *TSP-repr* by De Berg et al. [8] for Euclidean Path Cover on arbitrary planar point sets. Algorithm 1 gives our algorithm in a pseudocode, which is further explained below.

Algorithm 1 NarrowRectTSP-DP(P, δ).

Input: A set P of points in $[0, |P|] \times [0, \delta]$ chosen independently, uniformly at random
Output: The length of the shortest tour through all points in P

1: Compute the separators $s_1, \dots, s_t$ using Lemma 5, as explained above.
2: $A[0, \emptyset] := \{(\emptyset, 0)\}$
3: **for** $j = 1$ to $t + 1$ **do**
4:     **for all** $B \in \mathcal{B}_j$ **do**
5:         $A[j, B] := \emptyset$
6:         **for** $B' \in \mathcal{B}_{j-1}$ where $B' \cap \text{distant}(s_j) \subseteq B$ **do**
7:             **for all** $(M, x) \in$ *TSP-repr*$(P_j \cup B' \cup B,\ B' \triangle B)$ **do**
8:                 **for all** $(M', x') \in A[j - 1, B']$ **do**
9:                     **if** M' and M are compatible **then**
10:                         Insert $(\text{Join}(M, M'), x + x')$ into $A[j, B]$
11:         *Reduce*$(A[j, B])$
12: **return** $\text{length}(A[t + 1, \emptyset])$

The goal of Lines 4–11 is to compute a representative set $A[j, B]$ of $\mathcal{M}^*(P_1 \cup \dots \cup P_j \cup B,\ B)$ of size $2^{O(\sqrt{k})}$. We say that a point $p \in P$ is *distant* (with respect to a separator s_j) if it is more than five non-empty squares after s_j, and denote the set of distant input points from s_j by $\text{distant}(s_j)$. First, we iterate over all sets $B \in \mathcal{B}_j$ in Line 4. Next, we consider certain boundary sets $B' \in \mathcal{B}_{j-1}$. Notice that if there is a distant point $p \in B' \cap \text{distant}(s_j)$, then a tour edge crossing s_{j-1} ending at p also crosses s_j, and thus p is also a (distant) point of B.

In Line 7 we call the algorithm of De Berg et al. [8] for Euclidean Path Cover within the block P_j and the boundary points $B' \cup B$. This gives us a representative set $\mathcal{R}$ of $\mathcal{M}^*(P_j \cup B' \cup B,\ B' \triangle B)$. For each weighted matching $(M, x) \in \mathcal{R}$, and for each weighted matching from the representative set $(M', x') \in A[j - 1, B']$, we check if M and M' are compatible. If so, then taking the union of the corresponding path covers gives a path cover of $P_1 \cup \dots \cup P_j \cup B$ of total length $x + x'$, which realizes the matching $\text{Join}(M, M')$ on B; we then add $(\text{Join}(M, M^*), x + x^*)$ to $A[j, B]$ in Line 10.

After iterating over all boundary sets B', the entry $A[j, B]$ stores a set of weighted matchings, which we reduce to size $2^{O(\sqrt{k})}$ using the *Reduce* algorithm [3] in Line 11.

Note that in the final iteration, when $j = t+1$, we take $B = \emptyset$. Now M and M' are compatible if and only if the union of the corresponding path covers is a Hamiltonian cycle. Line 11 then gives to a single entry the smallest weight. Therefore, the length of the only entry in $A[t+1, \emptyset]$ after the loops have ended, is the length of the optimal TSP tour. Hence, the correctness of NarrowRectTSP-DP follows from the next lemma, proved in the full version [1].

▶ **Lemma 10.** *After Step 11, the set $A[j, B]$ is a representative set of $\mathcal{M}^*(P_1 \cup \cdots \cup P_j \cup B,\ B)$.*

Analysis of the running time. The loop of Lines 3–11 has $t + 2 = O(n)$ iterations. Each set $\mathcal{B}_j$ contains $2^{O(\sqrt{k})}n$ sets. For each choice of $B \in \mathcal{B}_j$, we have $2^{O(\sqrt{k})}$ options for B', since B can have at most one point distant from s_j by Lemma 6. The running time of *TSP-repr* is $T(|P|, |B|) = 2^{O(\sqrt{|P|+|B|})}$ [8, Lemma 8]. By Lemma 5 we have $|B| = O(\sqrt{k})$, so the running time of each call to *TSP-repr* in Algorithm 1 is $2^{O(\sqrt{3k+|B|}+\sqrt{|B|})} = 2^{O(\sqrt{k})}$. The representative set returned by *TSP-repr* has $2^{O(\sqrt{k})}$ weighted matchings, and the representative set of $A[j-1, B']$ also has $2^{O(\sqrt{k})}$ matchings. Checking compatibility, joining and insertion in Lines 9 and 10 takes $\mathrm{poly}(|M|, |M'|) = \mathrm{poly}(k)$ time. Consequently, before executing the reduction in Line 11, the set $A[j, B]$ contains at most $2^{O(\sqrt{k})} \cdot 2^{O(\sqrt{k})} = 2^{O(\sqrt{k})}$ entries. The application of the *Reduce* algorithm ensures that the constant in the exponent in the $2^{O(\sqrt{k})}$ is kept under control; see Lemma 9. Hence, the total running time is $n \cdot 2^{O(\sqrt{k})}n \cdot 2^{O(\sqrt{k})} = 2^{O(\sqrt{k})}n^2$.

4 Random point sets inside a narrow rectangle

The algorithm from Theorem 4 also works efficiently on random point sets inside a narrow rectangle, as stated in the following theorem.

▶ **Theorem 11.** *Let P be a set of n points chosen independently and uniformly at random from $[0, n] \times [0, \delta]$. Then a shortest tour on P can be computed in $2^{O(\sqrt{\delta})}n^2 + O(n^3)$ expected time.*

Proof. To prove that the expected running time of our algorithm is as claimed, we need a good bound on k, the expected maximum number of points falling in any square $\sigma_i := [(i-1)\delta, i\delta) \times [0, \delta]$. Note that $n_i := |P \cap \sigma_i|$ is a random variable with binomial distribution with parameters n and δ/n, so

$$\Pr[n_i = \ell] \;=\; \binom{n}{\ell}\left(\frac{\delta}{n}\right)^{\ell}\left(\frac{n-\delta}{n}\right)^{n-\ell}.$$

As above, let $k := \max_i n_i$. We need a strong upper bound on k. We have that

$$\Pr[k \geqslant \ell] \;=\; \Pr[\text{there is an } i \text{ such that } n_i \geqslant \ell] \;\leqslant\; \sum_{i=1}^{\lceil n/\delta \rceil} \Pr[n_i \geqslant \ell] \;=\; n \Pr[n_1 \geqslant \ell].$$

We use the Chernoff-Hoeffding theorem [14]: for a binomially distributed random variable x with parameters n, p and for $\ell > np$, we have that $\Pr(x \geqslant \ell) \leqslant \exp\left(-n \cdot D\left(\frac{\ell}{n}||p\right)\right)$, where $D(\frac{\ell}{n}||p) = \frac{\ell}{n}\ln(\frac{\ell/n}{p}) + \frac{n-\ell}{n}\ln(\frac{(n-\ell)/n}{1-p})$. Consequently,

$$\begin{aligned} n \Pr[n_1 \geqslant \ell] \;&\leqslant\; n \cdot \exp\left(-n\left(\tfrac{\ell}{n}\ln\tfrac{\ell/n}{\delta/n} + \tfrac{n-\ell}{n}\ln\tfrac{(n-\ell)/n}{(n-\delta)/n}\right)\right) \\ &=\; n \cdot \exp\left(-\ell\ln\tfrac{\ell}{\delta} - (n-\ell)\ln\tfrac{n-\ell}{n-\delta}\right). \end{aligned}$$

Assuming $e^2\delta < \ell$, we get

$$\begin{aligned}
\Pr[k \geqslant \ell] &< n \cdot \exp\left(-\ell \ln \tfrac{\ell}{\delta} + (n-\ell)\ln \tfrac{n-\delta}{n-\ell}\right)\\
&< n \cdot \exp\left(-\ell \ln \tfrac{\ell}{\delta} + (n-\ell)\ln \tfrac{n}{n-\ell}\right)\\
&< n \cdot \exp\left(-\ell \ln \tfrac{\ell}{\delta} + (n-\ell)\tfrac{\ell}{n-\ell}\right)\\
&= n \cdot \exp(-\ell(\ln \tfrac{\ell}{\delta} - 1))\\
&< ne^{-\ell},
\end{aligned}$$

where the third inequality uses that $\ln(x) < x - 1$ for $x > 1$. The running time of the algorithm can now be bounded the following way.

$$\begin{aligned}
\mathbb{E}[\text{running time}] &\leqslant \Pr[k \leqslant e^2\delta] \cdot 2^{O(\sqrt{\delta})}n^2 + \sum_{\ell=\lfloor e^2\delta+1\rfloor}^{n} \Pr[k=\ell] \cdot 2^{O(\sqrt{\ell})}n^2\\
&\leqslant 2^{O(\sqrt{\delta})}n^2 + n^2 \sum_{\ell=\lfloor e^2\delta+1\rfloor}^{n} \Pr[k \geqslant \ell] \cdot 2^{O(\sqrt{\ell})}\\
&\leqslant 2^{O(\sqrt{\delta})}n^2 + n^3 \sum_{\ell=\lfloor e^2\delta+1\rfloor}^{n} e^{-\ell}2^{O(\sqrt{\ell})}\\
&\leqslant 2^{O(\sqrt{\delta})}n^2 + n^3 \sum_{\ell=0}^{\infty} e^{-\ell+O(\sqrt{\ell})}\\
&\leqslant 2^{O(\sqrt{\delta})}n^2 + O(n^3)
\end{aligned}$$ ◀

5 Concluding remarks

Our paper contains two main results on EUCLIDEAN TSP. First, we proved that for points with integer x-coordinates in a strip of width δ, an optimal bitonic tour is optimal overall when $\delta \leqslant 2\sqrt{2}$. The proof of this bound, which is tight in the worst case, is partially automated to reduce the potentially very large number of cases to two worst-case scenarios. It would be interesting to see if a direct proof can be given for this fundamental result. Furthermore, we note that the proof of Theorem 1 can easily be adapted to point sets of which the x-coordinates of the points need not be integer, as long as the difference between x-coordinates of any two consecutive points is at least 1.

Second, we gave a $2^{O(\sqrt{\delta})}n^2$ algorithm for sparse point sets, which also works in $2^{O(\sqrt{\delta})}n^2 + O(n^3)$ expected time for random point sets. For $\delta = \Theta(n)$ the running time becomes $2^{O(\sqrt{n})}$, which is optimal under ETH. For small δ it would be interesting to improve the dependency on n in the running time. Another direction for future research is to study the problem in higher dimensions. We believe that our algorithmic results may carry over to $\mathbb{R}^d$ to points that are almost collinear, that is, that lie in a narrow cylinder. Generalizing the results to, say, points lying in a narrow slab will most likely be more challenging.

References

1 H Alkema, M. de Berg, and S. Kisfaludi-Bak. Euclidean TSP in narrow strips. *arXiv*, 2020. arXiv:2003.09948.

2 S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

3 H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.

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

5 T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms (3rd edition)*. MIT Press, 2009.

6 M. Cutler. Efficient special case algorithms for the n-line planar traveling salesman problem. *Networks*, 10:183–195, 1980. doi:10.1002/net.3230100302.

7 M. Cygan, S. Kratsch, and J. Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

8 M. de Berg, H.L. Bodlaender, S. Kisfaludi-Bak, and S. Kolay. An ETH-tight exact algorithm for Euclidean TSP. In *Proc. 59th IEEE Symp. Found. Comput. Sci. (FOCS)*, pages 450–461, 2018. doi:10.1109/FOCS.2018.00050.

9 M. de Berg, K. Buchin, B.M.P. Jansen, and G. Woeginger. Fine-grained complexity analysis of two classic TSP variants. In *Proc. 43rd Int. Conf. Automata Lang. Prog. (ICALP)*, pages 5:1–5:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.5.

10 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 2008. doi:10.1007/978-3-540-77974-2.

11 V.G. Deineko, M. Hoffmann, Y. Okamoto, and G.J. Woeginger. The traveling salesman problem with few inner points. *Oper. Res. Lett.*, 34(1):106–110, 2006. doi:10.1016/j.orl.2005.01.002.

12 V.G. Deineko, R. van Dal, and G. Rote. The convex-hull-and-line traveling salesman problem: a solvable case. *Inf. Proc. Lett.*, 51:141–148, 1994. doi:10.1016/0020-0190(94)00071-9.

13 V.G. Deineko and G. Woeginger. The convex-hull-and-k-line traveling salesman problem. *Inf. Proc. Lett.*, 59(3):295–301, 1996. doi:10.1016/0020-0190(96)00125-1.

14 J. Doe. Probability inequalities for sums of bounded random variables. *The Collected Works of Wassily Hoeffding*, pages 409–426, 1994. doi:10.1007/978-1-4612-0865-5_26.

15 H. Edelsbrunner, G. Rote, and E. Welzl. Testing the necklace condition for shortest tours and optimal factors in the plane. *Theoret. Comput. Sci.*, 66:157–180, 1989. doi:10.1016/0304-3975(89)90133-3.

16 M.R. Garey, R.L. Graham, and D.S. Johnson. Some NP-complete geometric problems. In *Proc. 8th ACM Symp. Theory Comp. (STOC)*, pages 10–22, 1976. doi:10.1145/800113.803626.

17 R.Z. Hwang, R.C. Chang, and R.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.

18 R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

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

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

21 C.H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoret. Comput. Sci.*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.

22 S. Rao and W. D. Smith. Approximating geometrical graphs via 'spanners' and 'banyans'. In *Proc. 30th ACM Symp. Theory Comp. (STOC)*, pages 540–550, 1998. doi:10.1145/276698.276868.

23 A.G. Reinhold. Some results on minimal covertex polygons. *Manuscript, City College of New York*, 1965.

24 G. Rote. The n-line traveling salesman problem. Networks, 22:91–108, 1992. doi:10.1002/net.3230220106.

25 D. Sanders. *On extreme circuits*. PhD thesis, City University of New York, 1968.

26 W.D. Smith and N.C. Wormald. Geometric separator theorems and applications. In *Proc. 39th IEEE Symp. Found. Comput. Sci. (FOCS)*, pages 232–243, 1998. doi:10.1109/SFCS.1998.743449.