The Rectilinear #-Bends TSP

Author

Estivill-Castro, Vladimir, Heednacram, Apichat, Suraweera, Francis

Published

2010

Journal Title

Lecture Notes in Computer science

DOI

10.1007/978-3-642-14031-0_30

Rights statement

Downloaded from

Griffith Research Online

The Rectilinear k-Bends TSP

Vladimir Estivill-Castro, Apichat Heednacram, and Francis Suraweera

IIIS, Griffith University, Nathan, QLD, 4111, Australia
{v.estivill-castro, f.suraweera}@griffith.edu.au,
apichat.heednacram@student.griffith.edu.au

Abstract. We study a hard geometric problem. Given n points in the plane and a positive integer k, the Rectilinear k-Bends Traveling Salesman Problem asks if there is a piecewise linear tour through the n points with at most k bends where every line-segment in the path is either horizontal or vertical. The problem has applications in VLSI design. We prove that this problem belongs to the class FPT (fixed-parameter tractable). We give an algorithm that runs in $O(kn^2 + k^{4k}n)$ time by kernelization. We present two variations on the main result. These variations are derived from the distinction between line-segments and lines. Note that a rectilinear tour with k bends is a cover with k line-segments, and therefore a cover by lines. We derive FPT-algorithms using bounded-search-tree techniques and improve the time complexity for these variants.

1 Introduction

The Minimum Bends Traveling Salesman Problem seeks a tour through a set of n points in the plane, consisting of the least number of straight lines, so that the number of bends in the tour is minimized. Minimizing the number of bends in the tour is desirable in applications such as the movement of heavy machinery because the turns are considered very costly. Both, general and rectilinear, versions of this problem are studied in the literature. In the general version, the lines could be in any configuration whereas in the rectilinear version, the line-segments[1] are either horizontal or vertical. The general version of the problem is NP-complete [2]. The hardness of the rectilinear version remains open, however, Bereg et al. [3] suspect that it is NP-complete because the Rectilinear Line Cover in 3 dimensions (or higher) is NP-complete [8]. The rectilinear version of the problems received considerable attention during 1990's [5, 9, 10] and recently [1, 3, 4, 13], since much of the interest in the rectilinear setting have been motivated by applications in VLSI. In the context of VLSI design, the number of bends on a path affects the resistance and hence the accuracy of expected timing and voltage in chips [10]. Stein and Wagner [12] solved approximately the rectilinear version of the Minimum Bends Traveling Salesman Problem. They gave a 2-approximation algorithm that runs in $O(n^{1.5})$ time. However, no polynomial-time exact algorithm is known for this rectilinear tour problem despite the motivating applications in VLSI.

In classical complexity theory, NP-completeness is essentially a tag for intractability to finding exact solutions to optimization problems. However, parameterized complexity theory [6, 7, 11] offers FPT (fixed-parameter tractable)

[1] A line is unbounded whereas a line-segment is bounded.

algorithms, which require polynomial time in the size n of the input to find these exact answers, although exponential time may be required on a parameter k.

We reformulate the Rectilinear Minimum Bends Traveling Salesman Problem as a parameterized problem and we call it the Rectilinear k-Bends Traveling Salesman Problem. From the parameterized complexity perspective, we show that the problem in general belongs to the class FPT by kernelization. As such, it can be solved exactly and in polynomial time for small values of the parameter. The requirement that line-segments of the tour are hosted exclusively by a line leads to a different variant of the problem. Another variant also emerges if we require that the same line-segment orientation cover points on the same line. We provide FPT-algorithms with improved complexity for these two variants of the rectilinear tour problem. Our algorithms for these variants are based on bounded-search-tree techniques.

2 Rectilinear Tours

We define the Rectilinear k-Bends Traveling Salesman Problem formally as follows. Given a set S of n points in the plane, and a positive integer k, we are asked if there is a piecewise linear tour (which may self-intersect) through the n points in S with at most k bends where every line-segment in the path is either horizontal or vertical (the tour must return to its starting point). An instance of the Rectilinear k-Bends Traveling Salesman Problem is encoded as the pair (S, k), and we call the solution a *rectilinear tour*. In this rectilinear version, the standard convention restricts the tour to $90°$ turns. A $180°$ turn is considered two $90°$ turns with a zero-length line-segment in between. If $n \geq 3$, it is always possible to transform a tour with a $180°$ turns into a tour with only proper $90°$ turns and line-segments of positive length. With these conventions, every $90°$ turn consists of one horizontal line-segment and one vertical line-segment, both of positive length. Thus, we assume $n \geq 3$, we also accept that there are no tours with an odd number of bends and that the required number k of bends is even. A rectilinear tour must have at least 4 bends.

Lemma 1. *If there exists a rectilinear tour with at most k bends, the number of horizontal line-segments is at most $k/2$ and the number of vertical line-segments is at most $k/2$.*

Proof. If there exists a tour with at most k bends, there are at most k line-segments. In a rectilinear tour, the number of horizontal line-segments is equal to the number of vertical line-segments. There cannot be more than $k/2$ horizontal line-segments and no more than $k/2$ vertical line-segments. □

We distinguish 3 types of rectilinear tours that derive from the distinction between line-segment and line. Fig. 1 illustrates this. In the first case, we require that if l is the line containing a line-segment s of the tour (i.e. $l \cap s = s$), then the line-segment s covers all the points in $S \cap l$. In the second case, if a point p is on a line l used by the tour, then there must be a segment of the tour with the same orientation as l covering p. The third type does not have any of the above constraints. For illustration, consider the set of points in Fig. 1 (a). Each vertical cluster or horizontal cluster of points is numerous enough to force being covered by at least one line-segment of the tour with minimum bends. Without any constraint, the two tours with 8 bends in Fig. 1 (b) are optimal; however,

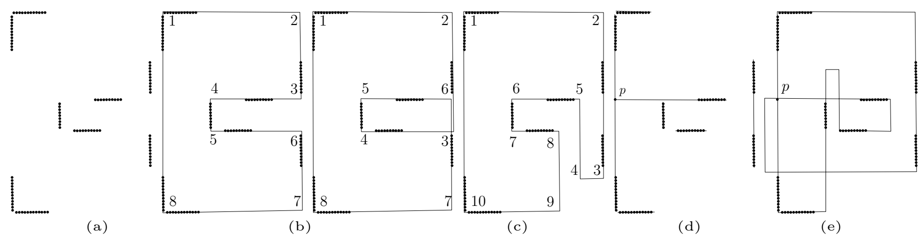

Fig. 1. Three types of the RECTILINEAR k-BENDS TRAVELING SALESMAN PROBLEM emerge as we considered the legal line-segments that are part of the tour.

in both there are two vertical line-segments that share a common line (in the second one, the two segments $\overline{2,3}$ and $\overline{6,7}$ are not drawn exactly co-linear so the reader can appreciate that the tour self intersects). The first constraint will require that all the points under these two line-segments be covered by only one line segment. In Fig. 1 (c) we see that this constraint forces the tour to travel over the large line-segment $\overline{2,3}$ and the minimal tour has now 10 bends. The second constraint can be illustrated if we add a point p to S as per Fig. 1 (d). This new point lies on lines used by two types of line-segments, one horizontal and one vertical and both of these types of line-segments must cover the point (that is, the point p cannot be already covered by the vertical line segment, because it belongs to a line where there is a horizontal line-segment of the tour). Note that this constraint is satisfied by points that are located at a bend. That is, if a bend is placed at a data point q, this constraint is automatically satisfied for q because the horizontal line-segment at q plus all other horizontal line-segments on the same horizontal line will cover q (and symmetrically for the vertical line-segment). In Fig. 1 (d) all the line-segments drawn must be contained in a line-segment of a minimum bends tour satisfying the second constraint (which now has 12 turns; see Fig. 1 (e)).

We first show that the problem in general (without any constraints) is FPT. The first variant requires that one line-segment covers all the points on the same line while the second variant requires the same orientation be represented by a line-segment that covers the points. We prove that the first variant and the second variant are also FPT but the time complexity of the FPT-algorithms here is smaller than in the general setting.

2.1 Tours without Constraints

We now proceed with the kernelization approach by presenting some reduction rules. Kernelization is central to parameterized complexity theory because a decision problem is in FPT if and only if it is kernelizable [11]. Intuitively, kernelization self-reduces the problem efficiently to a smaller problem using reduction rules. The rules are applied repeatedly until none of the rules applies. If the result is a problem of size no longer dependent on n, but only on k, then the problem is kernelizable because the kernel, the hard part, can be solved by exhaustive search in time that depends only on the parameter (even if it is exponential time).

Reduction Rule 1 *If $k \geq 4$ and all points in S lie on only one rectilinear line, then the instance (S, k) of the* Rectilinear k-Bends Traveling Salesman Problem *is a YES-instance.*

The next rule is derived from Lemma 1.
Reduction Rule 2 *If the minimum number of rectilinear line-segments needed to cover a set S of n points in the plane is greater than k, then the instance (S, k) of the* Rectilinear k-Bends Traveling Salesman Problem *is a NO-instance.*
We refer to a set of k rectilinear lines that cover the points in S as a k-cover. If (S, k) is a YES-instance of the Rectilinear k-Bends Traveling Salesman Problem, then the tour induces a k-cover. In fact, we can discover lines that host line-segments of any tour.

Lemma 2. *Let (S, k) be a YES-instance of the* Rectilinear k-Bends Traveling Salesman Problem. *Let l be a rectilinear line through $1 + k/2$ or more co-linear points. Then the line l must host a line-segment of any tour T with k or fewer bends.*
Proof. Without loss of generality, assume l is a vertical line. In contradiction to the lemma, assume there is no vertical segment on l for a tour T that covers with k bends. Then, the $1 + k/2$ points in $S \cap l$ would be covered by horizontal lines in T. According to Lemma 1, this contradicts T has k or fewer bends. □

The rectilinear line through S' in the proof above may be represented by separate line-segments of a witness tour. We now describe how to compute a k-cover if one exists. Consider a preprocessing of an input instance that consists of repeatedly finding $1 + k/2$ or more co-linear points and on a rectilinear line (that is, they are on a vertical or horizontal line). This process can be repeated until $k + 1$ rectilinear lines, each covering $1 + k/2$ or more different points are found, or no more rectilinear lines covering $1+k/2$ points are found. In the first case, we have $k + 1$ disjoint subsets of points each co-linear and each with $1 + k/2$ or more points. When this happens, we halt indicating a NO-instance. By Lemma 2, even if each of the $k + 1$ lines hosts only one line-segment in the tour, we would still have more than k line-segments. In the second case, once we discover that we cannot find a line covering $1 + k/2$ points and not exceeded k repetitions, we have a problem kernel.

Lemma 3. *Any instance (S, k) of the of the* Rectilinear k-Bends Traveling Salesman Problem *can be reduced to a kernel S' of size at most $k^2/2$.*
Proof. Let S' be the set of points after we cannot repeat the removal of points covered by a rectilinear line covering $1 + k/2$ or more points. Recall that if we repeated the removal more than k times, we know it is a NO-instance. If we repeated no more than k times and it is a YES-instance, a witness tour T would have matched hosting lines with the lines removed. Also T is a k-cover of S'. So the lines in T are rectilinear and each covers no more than $k/2$ points. This means $|S'| \leq k^2/2$. □

From the above lemma, if we have a kernel of size larger than $k^2/2$, then it is a NO-instance. Algorithmically, we can either determine that we have a NO-instance in polynomial time in k and in n, or we have a kernel where we still have to determine if it is a YES or NO-instance. What follows resolves this issue. With the next lemma we prove that each best tour is always equivalent to a tour with the same number of bends but where line-segments on the same line are not disjoint (as the two tours with 8 bends in Fig. 1 (b)).

Lemma 4. *Every optimal rectilinear tour T that has two disjoint line-segments hosted by the same line can be converted into a tour T' with the same number of bends and where the line segments have no gap.*

Proof. Consider first the case the two disjoint line segments s_1 and s_2 are traversed by T in the same direction. Fig. 2 (a) and (b) shows the transformation of the two line segments s_1 and s_2 in the same hosting line l_p into a new tour by enlarging both s_1 and s_2 and flipping the direction of two bends. Clearly, there are no more bends and although the direction of the path between these two bends is reversed, we still have a well formed tour. Note that Fig. 2 (a) deals with the case when the bends share a label[2] while (b) is the case the two bends share no label. Once this case is clear, the case where two disjoint line segments s_1 and s_2 are traveled by T in opposite directions is also clear, although now 4 bends are involved. Fig. 2 (c) illustrates this. □

Moreover, the transformation in the proof above always increases the length of the tour. So if we apply it again, it will not undo the work done by its previous application. Thus we can apply it repeatedly until there are never two disjoint line-segments in an optimal tour. In particular, we can assume optimal tours have no gap between co-linear line-segments like in Fig. 2 (a).

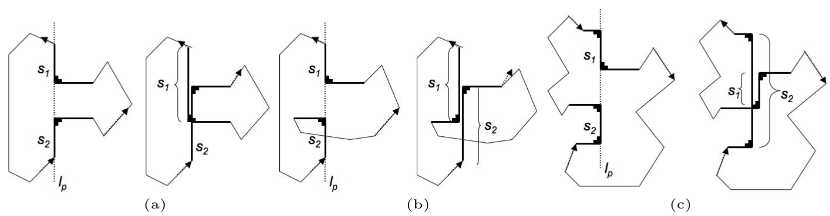

Fig. 2. The tour T is changed preserving the number of bends. Thick lines correspond to segments of the tour, while thin lines indicate the Jordan curve of the tour somewhere in the plane.

Let L_k be the set of rectilinear lines found by kernelization having at least $1 + k/2$ co-linear points. We know $|L_k| \leq k$ and all these lines have segments that are part of the tour. Given a vertical line $l \in L_k$, $cover(l) = S \cap l$. we let h_{max} be the horizontal line through the point $p_{max} \in cover(l)$ with the largest y coordinate, while h_{min} is the horizontal line through the point $p_{min} \in cover(l)$ with the smallest y coordinate. The line $h_{(max-i)}$ is the horizontal line through the i-th point in $cover(l)$ below p_{max} while $h_{(min+i)}$ is the horizontal line through the i-th point in $cover(l)$ above p_{min}, for $1 \leq i \leq (k/2-1)$.

[2] We say the turns have a common label if the turns are NE at p and NW at q (with N in common) or the turns are SE at p and SW at q (with S in common) where N, S, W, and E stand for North, South, West and East respectively.

We expand the set L_k as follows. For every vertical line $l \in L_k$, we add k horizontal lines $h_{max}, h_{(max-1)}, \ldots, h_{(max-k/2+1)}$ and $h_{min}, h_{(min+1)}, \ldots, h_{(min+k/2-1)}$ to L_k. Symmetrically, for every horizontal line $l \in L_k$, we add also k vertical lines $v_{max}, v_{(max-1)}, \ldots, v_{(max-k/2+1)}$ and $v_{min}, v_{(min+1)}, \ldots,\ v_{(min+k/2-1)}$ to L_k, where v_{max} passes through $p_{max} \in cover(l)$ with the largest x-coordinate and v_{min} passes through the point $p_{min} \in cover(l)$ with the smallest x-coordinate. The lines $v_{(max-i)}$ and $v_{(min+i)}$ are defined in a similar way to that of $h_{(max-i)}$ and $h_{(min+i)}$. Note that, if l covers less than k different points, we add a line that is orthogonal to l on every point in $cover(l)$. We call L_k with all these additional lines the set L'_k and $|L'_k| \leq k^2$. Let H be all the horizontal lines through a point in the kernel S' and let V be all the vertical lines through a point in S'. Thus, $|H| \leq k^2/2$ and $|V| \leq k^2/2$. Now, we add to L'_k all the lines in V and all the lines in H. The new set $R = L'_k \cup H \cup V$ has quadratic size in k. Moreover the set I of all intersections of two lines in R has also polynomial size in k (that is, $O(k^4)$). We will argue that a rectilinear tour of k bends exists for any given instance if and only if a tour with k bends exists with bends placed on I.

Lemma 5. *If the instance has a rectilinear tour T with k or fewer bends, then a tour can be built with k or fewer bends and all the bends are at I, the set of all intersections of lines in R where $R = L'_k \cup H \cup V$.*

Proof. We will show that for every YES-instance, we can transform the witness tour T to a tour T' with the same number of bends where every line-segment in T' is hosted by a line in R. From this, it follows that the set I hosts the possible positions for the bends in T'.

Let p be any point in S. If this is a YES-instance, there is a witness tour that has a line-segment l_p covering the point p. Moreover, because of Lemma 4, we can assume that if $s_1, \ldots, s_i$ are line-segments hosted by a rectilinear line l_p, there are no gaps; that is, $\cup_{j=1}^{i} s_j$ is not disjoint. If the point p was from the kernel, we are done because the line-segment is hosted on $H \subseteq R$ or on $V \subseteq R$. Otherwise, the point p was from a line discovered in kernelization. If the point p is covered by some s_j in the same orientation as the line discovered by kernelization, we are done. If the point is covered in the k-bends witness tour by a segment orthogonal to the line discovered in kernelization, the case becomes delicate. Assume p is on a vertical line l_p discovered by the kernelization, but in the witness tour T', p is covered in a horizontal line-segment over a line h_p. If h_p was discovered by kernelization, we are done, the same if it was a line in H. Therefore, line h_p is either above or below $\cup_{j=1}^{i} s_j$ because we are in a case where p is not covered by $\cup_{j=1}^{i} s_j$ (see Fig. 3, for example). Moreover, there cannot be more than $k/2$ points in the same situation as p. Otherwise, the witness structure would have more than $k/2$ horizontal line-segments (Lemma 1) at those positions. Therefore, the rank of p from either end of l_p must be no more than $k/2$. That is, p is in one of

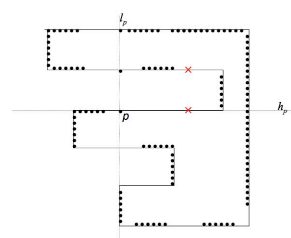

Fig. 3. The point p is covered by a horizontal line that is not found in the set H or the set L_k.

the lines $h_{max}, h_{(max-1)}, \ldots, h_{(max-k/2+1)}$ or $h_{min}, h_{(min+1)}, \ldots, h_{(min+k/2-1)}$ that we have in L'_k. Since $h_p \subseteq L'_k$, we have $h_p \subseteq R$. □

Theorem 1. *The* Rectilinear k-Bends Traveling Salesman Problem *is FPT.*

Proof. The algorithm computes R and searches exhaustively all tours over the lines in R. For example, a naive algorithm tests all subsets of size k of I to decide whether these k candidate-intersections can be completed into a tour with at most k bends and all the n points lie on the line-segments that make up the tour. Note that the k candidate-intersections of lines essentially host the bends in the tour. Since the number $|I|$ of intersections by lines in R is $O(k^4)$, the number of subsets of size k is bounded by $\binom{k^4}{k} = O(k^{4k}/k!)$. Testing all permutations to cover all tours result in time bounded by $O((k!)(n)(k^{4k}/k!)) = O(k^{4k}n)$. Kernelization can be performed in $O(kn^2)$ time. Thus, we can decide the Rectilinear k-Bends Traveling Salesman Problem in $O(kn^2 + k^{4k}n)$ time. □

2.2 Tours with Constraints

Now, we decide the problem for the rectilinear tour with at most k bends for the two variants introduced earlier. In fact, for both variants, the first phase computes several k-covers (candidates set L of k lines that cover all the n points in S). However, instead of kernelization, in order to identify hosting lines, the technique here will be a bounded-search-tree. The second phase checks if each candidate k-cover can constitute a tour based on such k-cover. To find these k-covers, we consider a search tree as illustrated by Fig. 4. In every node of the tree, the set L is a partial cover of S, and in some leaves it will be a k-cover. In the root of the tree, the set L is empty. In each internal node of the tree, we choose a point $p \in S \setminus cover(L)$ and we explore the two possibilities of enlarging L by analyzing the situation at p. Note that if the given instance were a YES-instance, every point in S is covered by a horizontal line (H-line), or a vertical line (V-line) (or both, if a bend or crossing is placed at the point). The point p is marked as covered with an H-line or V-line, depending on the branch of the tree. Also, the chosen line is added to L. The points that fall into the same class with p are also marked as covered so that we do not consider these points again in the next recursive call.

We keep track of how many vertical and horizontal assignments have been made and we emphasize that we do not assign more than $k/2$ horizontal lines and also no more than $k/2$ vertical lines. Each branch of the tree stops when the upper bound is reached or when every point is marked as covered. At the leaves of the tree, we have at most k lines that cover the set of n points, or exactly k lines that do not cover; in this case, we know the candidate set of lines cannot be completed into a covering tour. Therefore, the depth of the tree is at most k. This matches the pattern of a bounded-search-tree. The depth of the tree is bounded by the parameter and the fan-out is a constant (in this case the constant is 2).

Let L be the set of lines that cover the set S at a leaf of the tree where $|L| \leq k$. We only consider tours where every line-segment covers at least one point because for every tour T that covers S, there is an equivalent tour T' where every line-segment of the tour covers at least one point[3]. If T is a tour, we let $lines(T)$ be the set of lines used by the line-segments in T. Each line in $lines(T)$ covers at least one point and $lines(T)$ is a cover with k or fewer orthogonal lines. These observations allow us to state the following lemma.

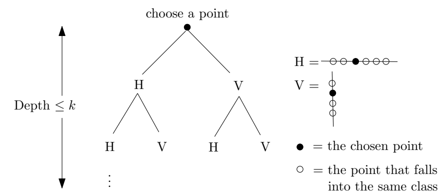

Fig. 4. Computing candidate sets of k lines.

Lemma 6. *If the instance has a tour T with k or fewer bends, then there is a leaf of the tree at the end of phase one where the cover made by the lines in L is consistent with the cover by $lines(T)$ (i.e. $L \subseteq lines(T)$).*

Proof. The algorithm of the first phase picks up a point p at each node. This point must be covered by $lines(T)$ with a horizontal or a vertical line. Therefore, there is a child that agrees with $lines(T)$ for this node. Traveling the path from the root of the tree down the child that agrees with $lines(T)$, we reach a leaf where each line in L must be in $lines(T)$. □

In the second phase, we investigate if those leaves of the first phase that cover the n points can result in a tour with the constraints of the variants.

Tours where one line-segment covers all the points on the same line: Here, we require that every line-segment s of a tour T must cover all the points in S on the line that includes s. In this special case of the Rectilinear k-Bends Traveling Salesman Problem, each candidate set L of lines at a leaf of the tree (recall Fig. 4) results in a candidate set of line-segments since we can simply connect the extreme points in $cover(l)$ for each line $l \in L$ to get the candidate line-segments. We can explore exhaustively if these candidate line-segments result in a tour. In the worst case, we have k line-segments from the first phase. These k line-segments are organized into $k!$ orders in a possible tour. In each of these orders, we can connect the line-segments (in the order chosen) one by one to form a tour. There are at most 4 ways to connect the line-segment l_i to the consecutive line-segment $l_{(i \bmod k)+1}$ for $i \in \{1, 2, \ldots, k\}$. Let a and b be the extreme points of l_i, while c and d are the extreme points of $l_{(i \bmod k)+1}$. We can connect these two line-segments as ac, ad, bc or bd (see Fig. 5). This means a total of $(k!)(4^k)$ tests. In some cases when we connect two line-segments together, the extension of these two line-segments may be enough to make a turn, therefore no additional line-segment is required. In some cases, it requires two additional line-segments as shown in Fig. 5. These extra line-segments cover no points, but they can be added in constant time when constructing the tour. Note

[3] If a line-segment in the tour covers no points, it can be translated in parallel until it is placed over one point.

that if the total number of line-segments in the final tour exceeds k, we simply answer no (Lemma 1).

We now analyze the total running time of our algorithm. It is obvious that the search tree for the first phase in Fig. 4 has at most $O(2^k)$ leaves and $O(2^{k-1})$ internal nodes. However, not every branch in the tree has to be explored. We explore only the branches that have the equal number of H-lines and V-lines. This is equivalent to choosing (among the k levels) a subset of size $k/2$ where to place the H-lines (or V-lines). Another way to recognize this is that each path from the root to a leaf in Fig. 4 is a word of length at most k with exactly $k/2$ symbols H and $k/2$ symbols V. Based on this analysis we reduce the size of the tree to $\binom{k}{\frac{k}{2}}$ which is simplified to $O(2^k/\sqrt{k})$ using Stirling's approximation. The work at each internal node is to choose a point, record the line and mark the associated points that are covered by that line. The dominant work is the computation at the leaves of the tree. Here we perform the tests to cover all tours that require time bounded by $O(\frac{2^k}{\sqrt{k}}(k!)(4^k)n)$ which is simplified to $O((2.95k)^k n)$. The time complexity is exponential in the parameter but linear in the size of the input. This gives the following result.

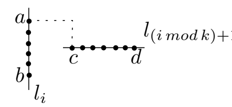

Fig. 5. One of four possible ways of joining two consecutive lines.

Theorem 2. *The* Rectilinear k-Bends Traveling Salesman Problem *where one line-segment covers all the points on the same line is FPT.*

Tours where the same line-segment orientation cover points on the same line: In this special case, a point that lies on a line hosting a horizontal-segment of the tour must be covered by a horizontal line-segment of the tour (possibly another horizontal-line segment, and possibly also a vertical line-segment). The trick is that it cannot be covered only by a vertical line-segment. The symmetric condition holds for points in a line hosting a vertical line-segment. We call this the *no distinct type of line-segment condition*.

In the first phase, a leaf that may hold a YES-instance has a candidate set L of no more than k lines and $L \subseteq lines(T)$. We expand this set of candidate lines as follows. For every vertical line $l \in L$ we add two horizontal lines h_{max} and h_{min}. The line h_{max} is the horizontal line through the point $p \in cover(l)$ with the largest y coordinate, while h_{min} is the horizontal line through the point $p \in cover(l)$ with the smallest y coordinate. Symmetrically, for every horizontal line $l \in L$, we add two lines v_{max} and v_{min} where v_{max} passes through $p \in l$ with the largest x-coordinate and v_{min} passes through the point $p \in l$ with the smallest x-coordinate. Note that L with all these additional lines has size linear in k. In what follows, we call the set L at a leaf with these additional lines, the set C of lines. Our aim is the next result.

Lemma 7. *If the instance has a tour T with k or fewer bends (and meeting the no distinct type of line-segment condition), then there is a leaf of the tree at the end of phase one where a tour can be built with k or fewer bends and all the bends are at intersections of lines in C.*

The proof shows that if we have a YES-instance, we can transform the witness tour T to a tour T' with the same numbers of bends, where every line-segment covers at least one point and $lines(T') \subseteq C$. From this, it follows that the intersections of all lines in C hosts the possible positions for the bends in T'.

The argument shows that every time we have a line-segment $\overline{pq}$ in a tour with its hosting line in $lines(T) \setminus C$, we can find a covering tour T' with the same bends and leading to the same leaf in phase one, the line-segment $\overline{pq}$ is not used and more line-segments in T' have their hosting lines in C. Consider a line-segment $\overline{pq}$ in a tour T that is a witness that the leaf is a YES-instance, but the line l hosting $\overline{pq}$ in the tour is such that $l \notin C$ (i.e. $l \in lines(T) \setminus C$). Without loss of generality, assume $\overline{pq}$ is horizontal (if $\overline{pq}$ were vertical, we rotate S and the entire discussion by 90°). Also, we can assume that $\overline{pq}$ covers at least one point in S and T has minimal number of bends. Let l_1 be the line-segment in the tour before $\overline{pq}$ and l_2 the line-segment in the tour after $\overline{pq}$.

Claim 1 *For all $p' \in S \cap \overline{pq} \setminus \{p, q\}$, there is a vertical line $l \in L$ (and thus $l \in C$) such that $p' \in cover(l)$.*

Proof. Let $p' \in S \cap \overline{pq} \setminus \{p, q\}$. Then p' is covered by a vertical line in L, because L covers S, and if p' was covered by a horizontal line, then the line hosting $\overline{pq}$ would be in L and $L \subseteq C$. This contradicts that the line hosting $\overline{pq}$ is not in C. If $\emptyset = S \cap \overline{pq} \setminus \{p, q\}$ the claim is vacuously true. □

Points in $S \cap \overline{pq} \setminus \{p, q\}$ are covered in T by vertical line-segments (if any point p' was covered only by $\overline{pq}$ and not a vertical segment in T through the vertical line at p', then T' would not satisfy the no distinct type of line-segment condition).

We will now distinguish two cases (refer to Fig. 6). In the first case, the tour T makes a U-return shape reversing direction, while in the second case, the tour makes a zig-zag shape and continues in the same direction. The bends at p and q of the line-segment $\overline{pq}$ make a U-return if they have one common label. The bends at p and q make a zig-zag of the line-segment $\overline{pq}$ if they have no common label. In this case the turns are NE at p and SW at q (with no letter label in common) or SE at p and NW at q (with no letter label in common). Without loss of generality, note that also a horizontal reflection along $\overline{pq}$ can be made so the other subcases can be ignored and we can assume the cases are as the two drawings of Fig. 6.

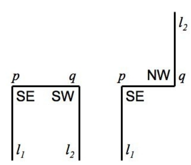

Fig. 6. The line $l \notin L$ hosts $\overline{pq}$ from T.

Case 1: In this case, we obtain the corresponding equivalent tour by shifting $\overline{pq}$ vertically. This is possible because all points in $S \cap \overline{pq} \setminus \{p, q\}$ are already covered by other vertical lines of T. In fact, if there is any point in $S \cap \{(x, y) | p_x \leq x \leq q_x \wedge y \geq p_y\}$[4], by shifting $\overline{pq}$ vertically (up) we can (without increasing the number of bends) overlaps with h_{max} for some vertical line in L. In fact, the set

[4] Here p_x is the x-coordinate of point x, thus $S \cap \{(x, y) | p_x \leq x \leq q_x \wedge y \geq p_y\}$ is all points in S above or on $\overline{pq}$.

$S \cap \{(x, y) | p_x \leq x \leq q_x \wedge y \geq p_y\}$ is not empty because $\overline{pq}$ has at least one point covered by a vertical line in L (L is a cover and we assumed the horizontal line hosting $\overline{pq}$ is not in L). It is important to note that our tour T' may self-intersect, but that is not a constraint for the problem.

Case 2: This setting has the following subcases. First we show that if $l_2 \notin L$, we can also change to a tour T' where now the set $lines(T') \setminus C$ is smaller. The symmetric argument shows that we can do this also if $l_1 \notin L$. Finally, the case left is when both $l_1, l_2 \in L$.

Subcase 2.a: If $l_2 \notin L$, then $q \notin S$. Because the line-segment $\overline{qq'}$ hosted by l_2 must cover one point $q' \in S$ and L is a cover, the point q' is covered by a horizontal line $l_3 \in L$. Fig. 7 (a) shows that the tour cannot have a SE turn at q', because then we can make a tour with 2 fewer bends contradicting the minimality of the witness tour (neither $\overline{qq'}$ nor $\overline{pq}$ are needed).

Thus, the turn at q' must be a SW turn. Fig. 7 (b) shows that a tour with a 180°-bend is equivalent and l_2 and the line hosting $\overline{pq}$ are not need. This makes $lines(T') \setminus C$ smaller by two lines.

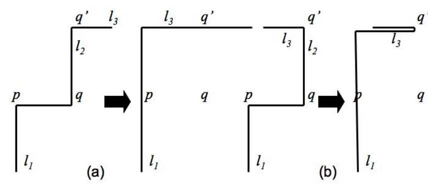

Fig. 7. The subcases if $l_2 \notin L$ can be converted and eliminate l_2.

Subcase 2.b: If $l_1 \notin L$, then $p \notin S$. The arguments is analogous to the previous case.

Subcase 2.c: Now we must have $l_1, l_2 \in L$. In this case, we make a cut and join operation to obtain a new tour that now is as in Case 1 (that is, we have a U-turn and not a zigzag).

First note that in this case there must be points in S covered by l_2 below or including q. Otherwise, the line h_{min} from l_2 can be made to coincide with $\overline{pq}$ and we are done. Similarly, there must be points in S covered by l_1 above p, otherwise $\overline{pq}$ can be made to coincide with h_{max} for l_1. Note also that there must be points in S above q covered by l_2 otherwise shifting $\overline{pq}$ can be made to coincide with h_{max} for l_2. Also, by an analogous argument, there must be points in S below p covered by l_1. The tour T must use a vertical line-segment to cover the points in l_2 below q,[5] because the tour complies with the no distinct type of line-segment condition. Assume that the tour T is traveled in the direction l_1, then $\overline{pq}$, then l_2 and let $\overline{p'q'}$ be a segment of this tour under q hosted by l_2, with $p'_y > q'_y$. In the chosen traversal of the tour T, $\overline{p'q'}$ may be traveled from p' to q' or from q' to p' (refer to Fig. 8).

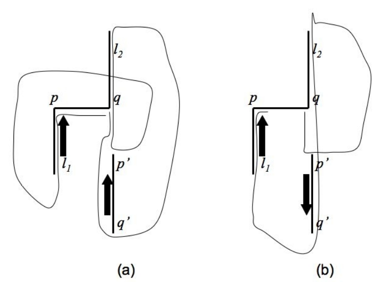

Fig. 8. The tour T is changed so there is no zig-zag.

[5] An analogous argument happens if the tour T uses a vertical line-segment to cover points above p covered by l_1.

If T travels from p' to q', we build a new tour that travels from q to p' and continues along T but in reverse, which ensures we will meet l_2 and then q again, but we now continue until q'. From here we continue along T in the same direction, which ensures we reach p. This new tour now makes a U-turn at $\overline{pq}$ and has the same set L as T and the same number of bends. For the case that T travels from q' to p', the new tour travels from $\overline{pq}$ to q'. Then along T in reverse order which guarantees arriving at q from l_2. We continue until p' and then in reverse order in T until we reach p again. Once again, this conversion reduces this case to Case 1 before.

We can now carry out the following procedure at each leaf. The number of intersections between any pair of lines in C is bounded by $O(k^2)$. We enumerate all words with repetitions of length k over the alphabet of intersections. We require repetitions since we need to consider tours with 180°-bends. A word like this encodes where the bends of a tour will be. For each of these words, we generate each of the possible placements given by the 4 types of 90°-bends (NE, NW, SE, SW) at each intersection. A placement of bends can be tested (in polynomial time) if it can be in fact completed into a tour and whether such tour covers S. The running time is bounded by $O(\frac{2^k}{\sqrt{k}}\frac{(2k^2)^k}{k!}(4^k)kn)$ which is simplified to $O((43.5k)^k n)$. This has clearly exponential complexity in k, but the overall algorithm has polynomial complexity in n and demonstrates the following result.

Theorem 3. *Under the constraint that a tour satisfies the no distinct type of line-segment condition, the* Rectilinear k-Bends Traveling Salesman Problem *is FPT.*

3 Conclusions

We have presented three FPT algorithms for different variants of the Rectilinear k-Bends Traveling Salesman Problem. We summarize these results in Table 1. It is apparent that the complexity of the algorithms is slightly better as more constraints are placed on the solution. Also, as we argued with an example, a solution for the general case may not be a solution for either of the constrained cases, and a solution of the constrained cases may require more bends although it is a solution for the general case. Therefore, the fact that one variant is FPT does not imply the other variant is FPT. While we have discussed the decision-version of the problem, it is not hard to adapt to an FPT-algorithm for the optimization version. For example, we can use an optimization-version FPT-algorithm to find a k'-cover where k' is the number of covering lines and

Table 1. FPT algorithms for finding a rectilinear tour with at most k bends.

Types of Rectilinear Tour	Time Complexities
1) general case (without constraints)	$O(kn^2 + k^{4k}n)$
2) same line-segment orientation cover points on the same line	$O((43.5k)^k n)$
3) one line-segment covers all the points on the same line	$O((2.95k)^k n)$

$O(k')$ is a bound for k where k is the minimum number of bends. In fact, it has been shown [12] that $k \leq 2k' + 2$. That is, we can approximate the minimum number of bends in FPT-time with respect to the sought minimum (or use a polynomial-time approximation algorithm [12] to obtain the approximate minimum). Then the decision problem can be used to find the exact value of the minimum in FPT-time (using binary search, for example).

References

1. Arkin, E., Bender, M., Demaine, E., Fekete, S., Mitchell, J., Sethia, S.: Optimal covering tours with turn costs. SIAM Journal of Computing **35(3)** (2005) 531–566
2. Arkin, E., Mitchell, J., Piatko, C.: Minimum-link watchman tours. Information Processing Letters **86(4)** (2003) 203–207
3. Bereg, S., Bose, P., Dumitrescu, A., Hurtado, F., Valtr, P.: Traversing a set of points with a minimum number of turns. Discrete and Computational Geometry **41** (2009) 513–532
4. Collins, M.: Covering a set of points with a minimum number of turns. Int. J. Comput. Geometry Appl. **14(1-2)** (2004) 105–114
5. de Berg, M., van Kreveld, M., Nilsson, B., Overmars, M.: Shortest path queries in rectilinear worlds. Int. J. Comput. Geometry Appl. **2(3)** (1992) 287–309
6. Downey, R., Fellows, M.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
7. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. Springer, Berlin (2006)
8. Hassin, R., Megiddo, N.: Approximation algorithms for hitting objects with straight lines. Discrete Applied Mathematics **30(1)** (1991) 29–42
9. Lee, D., Chen, T., Yang, C.: Shortest rectilinear paths among weighted obstacles. In: 6th ACM Symp. on Comput. Geometry (SCG 90), ACM, NY (1990) 301–310
10. Lee, D., Yang, C., Wong, C.: Rectilinear paths among rectilinear obstacles. Discrete Applied Mathematics **70(3)** (1996) 185–215
11. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and its Applications 31. Oxford University Press, New York (2006)
12. Stein, C., Wagner, D.: Approximation algorithms for the minimum bends traveling salesman problem. In: 8th IPCO. Springer, Berlin (2001) 406–422
13. Wagner, D., Drysdale, R., Stein, C.: An $O(n^{5/2} \log n)$ algorithm for the rectilinear minimum link-distance problem in three dimensions. Comput. Geom. Theory Appl. **42(5)** (2009) 376–387