

## FPT-algorithms for Minimum-Bends Tours

Vladimir Estivill-Castro

*School of Information and Communication Technology  
Griffith University, Nathan, QLD, 4111, Australia  
v.estivill-castro@griffith.edu.au*

Apichat Heednacram

*Institute for Integrated and Intelligent Systems  
Griffith University, Nathan, QLD, 4111, Australia  
apichat.heednacram@student.griffith.edu.au*

Francis Suraweera

*School of Information and Communication Technology  
Griffith University, Nathan, QLD, 4111, Australia  
f.suraweera@griffith.edu.au*

Received (received date)

Revised (revised date)

Communicated by (Name)

This paper discusses the  $k$ -BENDS TRAVELING SALESMAN PROBLEM. In this NP-complete problem, the inputs are  $n$  points in the plane and a positive integer  $k$ , and we are asked whether we can travel in straight lines through these  $n$  points with at most  $k$  bends. There are a number of applications where minimizing the number of bends in the tour is desirable because bends are considered very costly. We prove that this problem is fixed-parameter tractable (FPT). The proof is based on the kernelization approach using reduction rules. We also consider the RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM, which requires that the line-segments be axis-parallel. Note that a rectilinear tour with  $k$  bends is a cover with  $k$ -line segments, and therefore a cover by lines. We introduce two types of constraints derived from the distinction between line-segments and lines. We derive FPT-algorithms with different techniques and improved time complexity for these cases.

*Keywords:* Computational geometry; fixed-parameter tractable; minimum-bends tours, covering with hyperplanes.

### 1. Introduction

The TRAVELING SALESMAN PROBLEM (and its variations) is one of the most studied NP-hard problems<sup>1</sup>. The EUCLIDEAN TRAVELING SALESMAN PROBLEM is also NP-complete<sup>2</sup>. The MINIMUM BENDS TRAVELING SALESMAN PROBLEM seeks a tour through a set of  $n$  points in 2D, consisting of the least number of straight line-segments, so that the number of bends in the tour is minimized. The  $n$  points

in the tour are visited exactly once. The bends are the turns between the pair of line-segments. Minimizing the number of bends in the tour is desirable in applications such as VLSI and the movement of heavy machinery because the turns are considered very costly. In VLSI design, the number of bends on a path affects the resistance and hence the accuracy of expected timing and voltage in chips<sup>3</sup>. There are also applications where fewer bends are preferred such as surveying an area by aircraft, traffic routing, and space travel<sup>4</sup>.

A tour with minimum bends gives rise to a line cover. A line cover is a set of lines that cover the points in  $S$ . A simple observation is that a tour with  $k$  bends is a cover with  $k$ -line segments, and therefore a cover by lines. The main idea of our algorithms is to consider the line cover of the given points, since the size of the minimum line cover provides a lower bound of the number of bends of any traveling salesman tour. Both, general and rectilinear, versions of the MINIMUM BENDS TRAVELING SALESMAN 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 are either horizontal or vertical. The general version of the problem is NP-complete<sup>5</sup>. We prove elsewhere that the rectilinear version of the problem is also NP-complete<sup>6</sup>. The rectilinear version of the problems received considerable attention during 1990's<sup>7,8,3</sup> and recently<sup>9,10,11,12,4</sup>, since much of the interest in the rectilinear setting have been motivated by applications in VLSI. The MINIMUM BENDS TRAVELING SALESMAN PROBLEM aims to minimize only the number of bends and not the distance, therefore crossings are allowed. Stein and Wagner<sup>13</sup> provided an  $O(\log(\min(z, c)))$ -approximation algorithm that solves the general version of the problem, where  $z$  is the maximum number of co-linear points and  $c$  is the minimum number of lines used to cover all points. For its rectilinear version, they described a 2-approximation algorithm that runs in  $O(n^{1.5})$  time. However, there has been little progress in exact algorithms for this problem. However, no polynomial-time exact algorithm is known for this both versions of the problem despite the motivating applications mentioned earlier. In classical complexity theory, NP-completeness is essentially a tag for intractability to find exact solutions to the problems. However, parameterized complexity theory<sup>14,15,16</sup> offers FPT (fixed-parameter tractable) 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 small parameter  $k$ .

We prove that several tour problems are in the class FPT. As such, they can be solved exactly and in polynomial time for small values of the parameter. We discuss the MINIMUM BENDS TRAVELING SALESMAN PROBLEM in 2D. We reformulate this problem as a decision problem and we call it the  $k$ -BENDS TRAVELING SALESMAN PROBLEM. We give a set of practical reduction rules and we use these rules to show that the  $k$ -BENDS TRAVELING SALESMAN PROBLEM is FPT. When restricted to only horizontal and vertical line-segments we obtain the RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM. 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. Their algorithms are based on different techniques.

## 2. Line-Segments with Arbitrary Orientation

Formally, the  $k$ -BENDS TRAVELING SALESMAN PROBLEM asks, given a set  $S$  of  $n$  points in 2D and a positive integer  $k$ , if there is a piecewise linear tour through the  $n$  points in  $S$  with at most  $k$  bends. Problems with  $k \leq 2$  are considered NO-

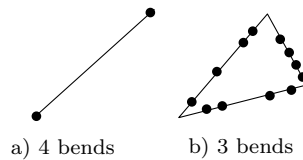


Fig. 1. The number of bends in a tour; a) a straight line-segment ( $k = 4$ ), b) a triangle ( $k = 3$ ).

instances. A bend of  $180^\circ$  is considered as 2 bends (see Fig. 1). This is the same as the turn of  $360^\circ$  angle. Thus, traversing on the same line-segment back and forward is 4 bends. Each vertex of a polygon is a bend and  $k = 3$  is smallest value with a YES-instance (for example, when  $S$  is covered by a line). Note that this is an important distinction of tours. Thus, we will assume that  $k > 2$  and  $S$  has no duplicates (it is a set).

**Lemma 1.** *There is always a minimum-bends tour without  $180^\circ$  bends.*

**Proof.** Let  $r$  be a  $180^\circ$ -bend in a tour  $T$  and  $b$  be the bend before  $r$ . Assume  $b$  is covered twice by  $T$  (else, reverse the order in  $T$ ). Let  $p$  be the point in  $T$  before  $b$ . The tour that replaces the section “from  $p$  to  $b$  to  $r$ ” by “from  $p$  to  $r$ ” has one less bend.  $\square$

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<sup>14</sup>. 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 can be solved by exhaustive search in time that depends only on the parameter (even if it is exponential time).

**Lemma 2.** *If the set of points in  $S$  can be covered with  $k > 2$  lines, then there is a tour with at most  $2k$  bends.*

**Proof.** Let  $L$  be a set of lines that cover  $S$ . Construct a tour by linking each pair of lines in  $L$  by an extra line-segment in between (if they are not parallel, we can

4 *V. Estivill-Castro et al.*

joining them at their intersection saving a bend). In the worst case, the number of line-segments is at most  $2|L|$  (since crossings are allowed). In a tour, one line-segment contributes to one bend. Thus, the number of bends is  $2|L|$ .  $\square$

**Lemma 3.** *If there exists a tour with at most  $k$  bends through a set  $S$  of  $n$  points in the plane, then these points can be covered by no more than  $k$  line-segments.*

**Proof.** Any tour with  $k$  bends is a cover with no more than  $k$  line-segments.  $\square$

This lemma implies the following result.

**Reduction Rule 1.** If the minimum number of line-segments to cover a set  $S$  of  $n$  points in the plane is greater than  $k$ , then the instance  $(S, k)$  of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM is a NO-instance.

We refer to a set of  $k$  lines that cover the points in  $S$  as a  $k$ -cover. If  $(S, k)$  is a YES-instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM, then we must have a  $k$ -cover. The idea is to first compute the  $k$ -cover, then use this information to construct the tour with at most  $k$  bends. The following rule places some of the lines in the cover and remove them from further consideration.

**Reduction Rule 2.** Let  $l$  be a line through  $k + 1$  or more co-linear points. Let  $S' = S \setminus \text{cover}(l)$  where  $\text{cover}(l)$  are all points in  $S \cap l$ . We focus now on finding the cover with  $(S', k - 1)$ .

**Proof.** If a line through  $k + 1$  different points was not in a  $k$ -cover, then we would need more than  $k$  lines to cover just this line<sup>17</sup>. Thus, if the set admits a  $k$ -cover, any line with  $k + 1$  or more points must be in any  $k$ -cover.  $\square$

The line through at least  $k + 1$  points mentioned above is clearly in the tour (although maybe as separate line-segments), otherwise we would need more than  $k$  lines to cover just this line resulting in a tour with more than  $k$  bends (Lemma 1). Note the distinction between a line (an unbounded infinite set of points with zero-width) and a line-segment which is the bounded portion of a line that constitutes the shortest path between the extremes of the segment<sup>18</sup>. Line covers are made with lines, while tours are made with line-segments and bends at the extremes of the segments. However, finding a tour with the minimum number of bends is more complicated than finding a line cover for a set of points. The main reason is that, a tour with the minimum number of bends may use a line-segment that does not cover all the points on the same line. See Fig. 2 for an example. Although it is not difficult to compute the candidate line, computing the segments of the tour needs more attention since the same line may appear as separate line-segments in the optimal tour. Moreover, there could be many of these line-segments. Therefore, we will consider the problem in two aspects. That is, we investigate tours where we allow several line-segments in the tour to be on the same line, then we look at the problem where one line-segment in the tour covers all the points on the same line.

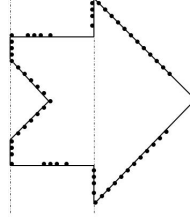


Fig. 2. A tour that has two lines that are not just one line-segment in the tour with minimum number of bends.

### 2.1. Tours where several line-segments cover points on the same line

First we describe how to compute the  $k$ -cover. Consider a preprocessing of an input instance that consists of repeatedly applying Reduction Rule 2. We apply this rule until  $k + 1$  lines, each covering  $k + 1$  (or more) different points are found, or no more lines covering  $k + 1$  points are found. In the first case, we have  $k + 1$  disjoint subsets of points each co-linear and each with  $k + 1$  or more points. When this case happens Reduction Rule 1 is invoked and we have a NO-instance. Even if each of the  $k + 1$  lines results into only one line-segment in the tour, we would still have more than  $k$  line-segments. If we find that we cannot find a line covering  $k + 1$  points, we have a problem kernel.

**Lemma 4.** *Any instance  $(S, k)$  of the  $k$ -cover can be reduced to a problem kernel  $S'$  of size at most  $k^2$ .*

**Proof.** Assume by contradiction that preprocessing of  $(S, k)$  results in a reduced instance  $|S'| > k^2$ . After Reduction Rule 2 cannot be applied anymore, each of the covering  $k$  line-segments contains at most  $k$  points in  $S'$ , therefore,  $|S'| \leq k^2$ . This contradicts that  $|S'| > k^2$ .  $\square$

Kernelization of the  $k$ -cover can be performed in polynomial time on the size of the input. It requires at most  $k + 1$  applications of Rule 2 to decide a NO-instance. If it is not a NO-instance, we keep a record of the lines found by Rule 2 so that later we can use this information to decide a YES-instance. Finding a line with more than  $k + 1$  points (or determining that one does not exist) can be done by placing all points in dual space and finding a vertex in the dual of degree  $k + 1$  or more (or determining this does not exist). In fact, the dual can be constructed once and placed in an arrangement which can be searched by a plane sweep and updated by removing the dual-lines that intersect at a vertex of degree equal or greater than  $k + 1$ . Asymptotically, this can have even faster time-complexity with the algorithm of Guibas et al. <sup>19,20</sup>, however, our point is that each application of the rule is bounded by  $O((n + k) \log n + n^2) = O(n^2)$  time, resulting in  $O(kn^2)$  total time for all the applications of Rule 2. Rule 1 is invoked if Rule 2 is applied more than  $k$  times. In total the kernelization process requires  $O(kn^2)$  time.

If kernelization of the  $k$ -cover resulted in a large kernel (greater than  $k^2$ ), we answer as a NO-instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM in constant time (Lemma 1). However, we still have to decide the instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM if the kernel is small. Note that a YES-instance of the  $k$ -cover does not imply a YES-instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM. Unfortunately, we cannot use Reduction Rule 2 with the instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM in the usual sense since we do not know how many line-segments of the tour are going to be hosted by a line over  $k + 1$  points or more. So, we cannot provide a value of  $k$  for exhaustive search of the reduced instance of the tour. Therefore, to complete the argument that  $k$ -BENDS TRAVELING SALESMAN PROBLEM is fixed-parameter tractable we need to show that a tour with at most  $k$  bends can be constructed from the  $k$ -cover in polynomial time in the size of the input when the kernel of the  $k$ -cover is small (no more than  $k^2$  points). For this we require the following result.

**Lemma 5.** *If a tour  $T$  has the minimum number of bends, then every line segment in  $T$  covers at least two points*

**Proof.** If a line-segment in the tour covers no points, it can be replaced by a parallel line-segment that covers at least one point. Consider a tour with a line-segment  $l$  that covers only one point  $p \in S$ . Then, simply spinning  $l$  (rotating on  $p$  as an axle and sliding the bends of  $l$  until another points is met), we find an equivalent tour with the same number of bends where now  $l$  covers at least 2 points.  $\square$

We now propose the following algorithm. Let  $L_{k+1}$  be the set of lines found by the exhaustive applications of Reduction Rule 2. We know  $|L_{k+1}| \leq k$  and all these lines have segments that are part of the tour (if it is a YES-instance) so we will later use these lines to construct the tour. Consider the kernel of the  $k$ -cover which has at most  $k^2$  points. We put the two extremal points of each line of  $L_{k+1}$  into the kernel, so our reduced instance  $S'$  has at most  $k^2 + 2k$  points. Let  $R$  be all the lines through two or more points of  $S'$ . The size of  $R$  is less than the number of pairs of points in  $S'$ . Thus,  $|R| = O(k^4)$ . Now, we search exhaustively all tours over the lines in  $L_{k+1} \cup R$ . For example, a naive algorithm for this part tests all subsets of size  $k$  of intersections of lines in  $L_{k+1} \cup R$  to decide whether we can link these  $k$  candidate-intersections into a tour with at most  $k$  bends and all the  $n$  points are covered by the line-segments that make up the tour. Note that the  $k$  candidate-intersections of lines essentially host the bends in the tour. Since  $|L_{k+1} \cup R| = O(k^4)$ , the number of intersections is  $O(k^8)$ , the number of subsets of size  $k$  is bounded by  $\binom{k^8}{k} = O(k^{8k}/k!)$ . Testing all permutations of the  $k$  candidate-intersections hosting the bends in the tour results in time bounded by  $O((k!)(n)(k^{8k}/k!)) = O(k^{8k}n)$ . Therefore, we can decide the  $k$ -BENDS TRAVELING SALESMAN PROBLEM in  $O(kn^2 + k^{8k}n)$  time.

**Theorem 1.** *The  $k$ -BENDS TRAVELING SALESMAN PROBLEM is FPT.*

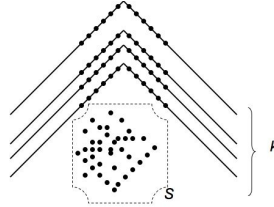
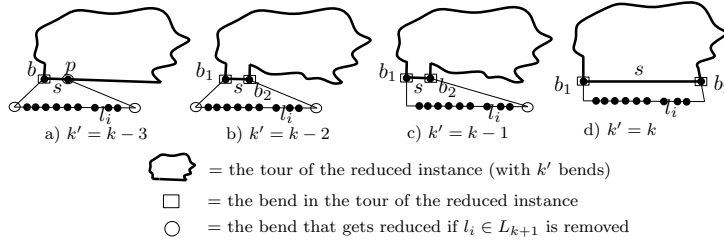


Fig. 3. Construction to reduce in polynomial time LINE COVER to  $k$ -BENDS TRAVELING SALESMAN PROBLEM with no two line-segments in the tour sharing a line.

## 2.2. One line-segment covers all the points on the same line

In this subsection we show that the bounded-search-tree technique does apply to a special case of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM. Namely, consider this problem but the tour must be such that each line-segment in the tour covers all the points on the same line. Thus, in this case a cover using a line is the same as a cover using a line-segment. By revisiting the proof by Arkin et al. <sup>5</sup>, we will show that this restricted problem remains NP-complete. This is important because it means that this restriction remains a problem where no polynomial-time exact algorithm, unless  $P=NP$ . The reduction is from LINE COVER which is known to be NP-hard <sup>21</sup> for over 20 years (in fact, APX-hard <sup>22</sup>). Consider an instance  $(S, k)$  of LINE COVER (we are asked if the set  $S$  of  $n$  points can be covered with  $k$  lines). We will produce an instance of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM where the tour with minimum bends has line-segments that correspond uniquely to lines. The idea is to surround the points in  $S$  with  $k$  parallel wedges, each wedge is made of two lines (refer to Fig. 3). By pre-computing all possible slopes of any line covering two or more points in  $S$  we can make sure that none of the lines in the wedges are parallel to lines that cover two or more points in  $S$ . We also place more than  $|S|$  points in the apex of each wedge to force tours to travel each wedge and make one bend at its apex. Call these additional points  $S'$ . The instance for  $k$ -BENDS TRAVELING SALESMAN PROBLEM has an input set  $S \cup S'$  and a number of bends is set to  $3k$ . It is easy to see that if the instance of LINE COVER is a YES-instance, then the  $k$  lines can be used to make a tour by traveling a covering line until it meets a leg of a wedge, moving along the wedge to its apex, then to the other leg of the wedge, and exiting with another covering line. The tour has  $3k$  bends because there is the bend at the apex and one bend at each leg of each wedge. More importantly, for our purposes, note that every point on a line is always covered by just one line-segment. Conversely, a tour with  $3k$  bends would need to use the wedges, and therefore, would need to cover  $S$  with  $k$  lines. However, we now show that this version of the problem is FPT.

We apply the preprocessing of the previous section to repeatedly find a line through  $k + 1$  or more co-linear points. When we can no longer find such a line, we have a reduced instance of size at most  $k^2$ . In this setting of the problem we know that once we find  $k + 1$  or more co-linear points, the entirety of these points must be in a line-segment of the tour. This enables us to be precise about Reduction Rule 2.

8 *V. Estivill-Castro et al.*

 Fig. 4. The minimum number of bends ( $k'$ ) after applying Reduction Rule 2.

However, to complete the algorithm, there would be several cases and this time we can use the bounded-search-tree technique to ensure all possibilities of linking the  $k + 1$  or more points into a tour. We discuss this in the next paragraph.

Recall that  $L_{k+1}$  is a set of all lines that cover at least  $k + 1$  points. The minimum number of bends required to cover the tour when  $l_i$  was removed, is denoted by  $k'$ . Although the rule at the moment tells us that we can remove the points covered by  $l_i \in L_{k+1}$  from consideration in the reduced instance, it does not tell us what value for  $k'$  should be considered in the reduced instance. That is, from the instance  $(S, k)$  of the  $k$ -BENDS TRAVELING SALESMAN PROBLEM, we move to an instance of the form  $(S \setminus \text{cover}(l_i), k')$ . Interestingly, the parameter  $k'$  that is the number of bends in the tour of the new (reduced) instance, can appear in 4 different ways depending on how  $l_i$  is connected to the reduced instance tour in order to build a tour for the original instance. Fig. 4 shows the four possibilities of reducing the number of bends in the tour. Each reduction of the instance  $(S, k)$  is required to meet some conditions in order to ensure correctness. We define 4 conditions that the tour of the reduced instance may have to meet. For the correctness of the algorithm, we will recursively test all of the following claims.

**Claim 1.** *If the reduced instance can be covered with  $k' = k - 3$  bends, then the original instance can be covered with  $k$  bends (see Fig. 4a).*

**Proof.** Let  $t$  be a tour of  $S \setminus \text{cover}(l_i)$  with  $k' = k - 3$  or less bends and with line-segments in a bijection with lines. A tour of  $S$  can be constructed as follows. Let  $p$  be the next point in  $t$  after a bend  $b$ . Construct a tour of  $S$  by replacing a line-segment  $s$  (a line-segment from  $b$  to  $p$ ) by a path of 3 line-segments as follows. A segment from  $b$  to the first point in  $l_i$ , a segment along  $l_i$  covering all these points and then a segment from the last point in  $l_i$  to  $p$ . This tour adds two bends in  $l_i$  and one in  $p$ , so the new tour has  $k$  bends. The line for  $l_i$  is a new line because all points in  $S$  covered by  $l_i$  were removed in the reduced instance. The line-segments  $b$  to the first point in  $l_i$  and the last point in  $l_i$  to  $p$  can be shifted along  $l_i$  away from points in  $l_i$  to ensure these are new lines.  $\square$

**Claim 2.** *If the reduced instance can be covered with  $k' = k - 2$  bends and the tour has a line-segment with no points (except maybe at the extremes) and no two*



line-segments share a line, then the original instance can be covered with  $k$  bends (see Fig. 4b).

**Proof.** Let  $b_1$  and  $b_2$  be the extremes of a line-segment  $s$  in a tour  $t$  of the reduced instance with  $k - 2$  bends and there are no points in  $s$  (except maybe at  $b_1$  or at  $b_2$ ). Construct a tour for the original instance from  $t$  but replacing  $s$  by a path that goes from  $b_1$  to the first point in  $l_i$ , then traveling  $l_i$  to cover all points of  $l_i$  and then a segment from the last point of  $l_i$  to  $b_2$ . This new tour adds only 2 bends at  $l_i$  so it has  $k$  bends. The line  $l_i$  is new and the other two segments can be shifted away from points in  $l_i$  to ensure they are also in new lines.  $\square$

**Claim 3.** *If the reduced instance can be covered with  $k' = k - 1$  bends of a tour  $t$ , and there is one bend  $b_1$  where the line-segment  $s'$  at  $b_1$  can be extended to meet  $l_i$  without partitioning the points covered by  $l_i$ , then the original instance can be covered with  $k$  bends (see Fig. 4c).*

**Proof.** Let  $s$  be the segment after  $s'$  in  $t$ . Similarly to Claim 1 and Claim 2 above, the path  $s's$  in  $t$  is replaced to construct a tour of the original instance  $S$ . First  $s'$  is extended to meet  $l_i$ , then travel along  $l_i$  covering all the points and then a new segment is made from the last point of  $l_i$  to the extreme of  $s$ . This adds one bend. The extension of  $s'$  is the same line, the line  $l_i$  is new and the last segment can be shifted on  $l_i$  to make sure that it is hosted by a new line.  $\square$

**Claim 4.** *If the reduced instance can be covered with  $k' = k$  bends, and there are two bends  $b_1$  and  $b_2$  with a line-segment  $s$  between them, and the previous line-segment  $s'$  and the next line-segment  $s''$  can both be extended to meet  $l_i$  without partitioning the points but the points in  $\text{cover}(l_i)$  lie in between the intersection, then the original instance can be covered with  $k$  bends (see Fig. 4d).*

**Proof.** Again, the segment  $s$  in  $t$  is replaced by the extensions of the segment before it, the traveling of  $l_i$  and the extension of the segment after  $s$ . This adds no bends since  $b_1$  and  $b_2$  are moved to sit at  $l_i$  and the new line-segment along  $l_i$  is in a new line.  $\square$

Fig. 5 shows this corresponds to bounded-search-tree technique as the recursion tree is bounded. If there is no line with  $k + 1$  points or more, we solve the instance  $(S, k)$  exhaustively. Otherwise, we remove a line  $l_i \in L_{k+1}$  from the tour and recursively explore the four possibilities of reducing the number of bends in the tour when such a line is removed. The parameters are passed by value and the returned result is a sequence of all tours with  $k$  or fewer bends that have one line-segment covering all the points on the same line (modulus the equivalence of Lemma 5). To simplify the pseudo-code we assume we have a Boolean function  $\text{CHECKCLAIM}(t, C)$  that given a tour  $t$ , returns **true** if the tour  $t$  satisfies Claim  $C$  (and provides also

10 *V. Estivill-Castro et al.*

K-BENDS( $S, bound, depth$ )

Input: A set  $S$  of  $n$  points and a positive integer  $k$

Output: A list of tours covering  $S$  with  $k$  bends or less where no two line-segments are hosted by the same line; the list is empty if there is no such tour.

```

1  if  $bound \leq 2 \vee depth > k$ 
2    then return  $\emptyset$ 
3  if  $S$  can be covered by a triangle
4    then return {triangle}
5  if there is no line with  $k + 1$  points or more
6    then (*Solving the instance of at most  $k^2$  points*)
7    return all tours of  $(S, k)$  by exhaustive search
8  else (*Checking 4 claims recursively*)
9    Let  $l_i$  be a line that covers at least  $k + 1$  points in  $S$ 
10    $R \leftarrow \emptyset$ ;
11    $T \leftarrow$  K-BENDS( $S \setminus cover(l_i), bound - 3, depth + 1$ )
12   for  $t \in T \wedge CHECKCLAIM(t, Claim 1)$ 
13     do  $R \leftarrow R \cup BUILDTOURS(t, Claim 1)$ 
14    $T \leftarrow$  K-BENDS( $S \setminus cover(l_i), bound - 2, depth + 1$ )
15   for  $t \in T \wedge CHECKCLAIM(t, Claim 2)$ 
16     do  $R \leftarrow R \cup BUILDTOURS(t, Claim 2)$ 
17    $T \leftarrow$  K-BENDS( $S \setminus cover(l_i), bound - 1, depth + 1$ )
18   for  $t \in T \wedge CHECKCLAIM(t, Claim 3)$ 
19     do  $R \leftarrow R \cup BUILDTOURS(t, Claim 3)$ 
20    $T \leftarrow$  K-BENDS( $S \setminus cover(l_i), bound, depth + 1$ )
21   for  $t \in T \wedge CHECKCLAIM(t, Claim 4)$ 
22     do  $R \leftarrow R \cup BUILDTOURS(t, Claim 4)$ 
23   return  $R$ 

```

Fig. 5. Recursive FPT-algorithm with bounded depth.

in a parameter by reference the list of bends where the claim is satisfied). We also assume that we have a function  $BUILDTOURS(t, C)$  that builds the corresponding list of tours for the original instances from a solution tour of the reduced instances that satisfy Claim  $C$ . Note that from a solution of the reduced instance, there are at most  $k$  ways to construct a solution of the original instance by any of the constructions of the 4 claims above, because all of these constructions replace at most one line-segment in the tour and the reduced instance must have a tour with less than  $k$  line-segments.

The K-BENDS function above is called as  $K-BENDS(S, k, 0)$  and note that  $k$  is maintained as a global value while the third parameter accounts for the depth in the recursion tree (ensuring bounded depth of  $k$ ). We analyze the asymptotic running time of the K-BENDS function in Fig. 5. First, checking if  $bound \leq 2$  or if  $S$  can be covered by a triangle can be done trivially in linear time. The running time of checking 4 claims recursively, however, is not that obvious. We show that this task can be carried out in polynomial time in  $n$  (although exponential in  $k$ ). The fan-out of the tree is 4. Hence, the worst-case number of leaves in the recursion tree is  $O(4^k)$  and the number of internal nodes in the tree is  $O(4^{k-1})$ . The work at each internal node is polynomial in  $n$ . At the leaves, there is no line with  $k + 1$  points or more, we

solve the problem kernel exhaustively by testing all subsets of size  $k$  of  $\binom{k^2}{2}$  lines. That is,  $\binom{k^4}{k} = O(k^{4k}/k!)$  subsets to be tested. When we consider all the leaves, the work becomes  $O((4^k)(k^{4k}/k!)(k!)(k^2)) = O(4^k k^{4k+2})$  time. Note that the list passed back by recursive calls has size bounded by a function of  $k$  (and independent of  $n$ ), the work in the nodes by CHECKCLAIM and BUILDTOURS depends only on  $k$  and not on  $n$ . A careful analysis gives the total running time of  $O(kn^2 + 4^k k^{4k+2})$ .

**Theorem 2.** *The  $k$ -BENDS TRAVELING SALESMAN PROBLEM with line-segments in unique lines is FPT.*

We note that the complexity of the fixed-parameter tractable algorithm here is smaller than when we allow line-segments to share a line.

### 3. Rectilinear Line-Segments

First, we define the RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM formally as follows. Given a set  $S$  of  $n$  points in 2D, and a positive integer  $k$ , we are asked if there is a piecewise linear tour through the  $n$  points in  $S$  with at most  $k$  bends where every line in the path is either a horizontal or a vertical line-segment. 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*. Note that in this rectilinear version of the problem, the standard convention is to restrict the tour to  $90^\circ$  turns. A  $180^\circ$  turn is considered two  $90^\circ$  turns with a zero-length line-segment in between. If  $n \geq 3$ , it is always possible to transform a tour with a  $180^\circ$  turns into a tour with only proper  $90^\circ$  turns and line-segments of positive length. With these conventions, every  $90^\circ$  turn consists of one horizontal line-segment and one vertical line-segment, both of positive length. Therefore, we will assume  $n \geq 3$ , we also accept that there are no tours with an odd number of bends and that the required number of bends is even. A rectilinear tour must have at least 4 bends.

**Lemma 6.** *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 equals 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.  $\square$

We distinguish 3 types of rectilinear tours that derive from the distinction between line-segment and line. Fig. 6 illustrates this. In the first case, we require that if  $l$  is a line hosting a line-segment  $s$  of the tour, then the line-segment  $s$  covers all the points in  $S \cap l$ . In the second case, the tour may have more than one line-segment covering a subset  $S'$  of co-linear points in  $S$ , however, all points in  $S'$  must be covered by at least one line-segment coinciding with the line through  $S'$ . That is, for

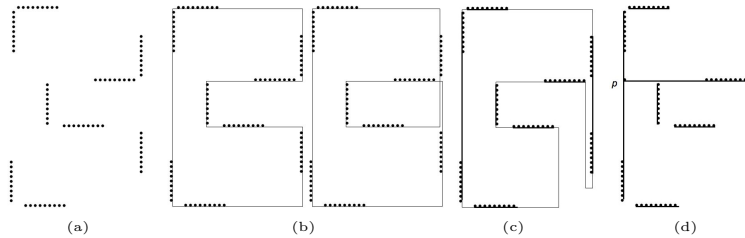
12 *V. Estivill-Castro et al.*

Fig. 6. Three types of the RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM emerge as we considered the legal line-segments that are part of the tour.

all subsets of co-linear points  $S'$  under horizontal or vertical alignment, there is a cover with only one orientation of line-segments from the tour. The third type does not have any of the above constraints. For illustration, consider the set of points in Fig. 6 (a). Without any restriction, the two tours with 8 bends in Fig. 6 (b) are optimal; however, in both there are two vertical line-segments that share a common line (in the second one, the two segments are not drawn exactly co-linear so the reader can appreciate that the tour self intersects). The first restriction will require that all the points under these two line segments be covered by only one line segment. In Fig. 6 (c) we see that this constraint forces the tour to travel over several line-segments 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. 6 (d). This new point lies on lines used by two types of line-segments, one horizontal and one vertical and both of these must cover the point (that is, the point  $p$  cannot be already covered by the vertical line segment, because it belong to a line where there is a horizontal line segment of the tour). Note that 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).

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 line-segment orientation cover points on the same line. 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 the general setting.

### 3.1. Tours without constraints

We start our discussion with the following series of reduction rules.

**Reduction Rule 3.** 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 6.

**Reduction Rule 4.** If the minimum number of line-segments 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 we must have a  $k$ -cover. We will first compute the  $k$ -cover, then use this information to construct the rectilinear tour with at most  $k$  bends. The following rule places some of the rectilinear lines in the cover and removes them from further consideration.

**Reduction Rule 5.** Let  $l$  be a rectilinear line through  $(k/2 + 1)$  or more co-linear points. Let  $S' = S \setminus \text{cover}(l)$  where  $\text{cover}(l)$  are all points in  $S \cap l$ . We focus now on finding the cover for  $(S', k - 1)$ .

**Proof.** If a vertical line through  $(k/2 + 1)$  different points was not in a  $k$ -cover, then we would need more than  $k/2$  horizontal lines to cover just this line. Similarly, if a horizontal line through  $(k/2 + 1)$  different points was not in a  $k$ -cover, then we would need more than  $k/2$  vertical lines to cover this line.  $\square$

Similar to the general version of the problem, we repeatedly find a rectilinear line through  $(k/2 + 1)$  or more co-linear points in the preprocessing stage. When we can no longer find such a line, we have a problem kernel.

**Lemma 7.** *Any instance  $(S, k)$  of the  $k$ -cover can be determined to be a NO-instance, or it can be reduced to a problem kernel  $S'$  of size at most  $k^2/2$ .*

**Proof.** Let  $S'$  be the set of points after Reduction Rule 5 cannot be applied any more. If it is empty, then we have a YES-instance with empty kernel. If the reduction rule was applied more than  $k$  times, we know we have a NO-instance. Otherwise, if it were a YES-instance, because any tour with  $k$  bends implies a  $k$ -cover of  $S$  and a  $k$ -cover of  $S$  is a  $k$ -cover of  $S' \subseteq S$ , we would have a covering with  $k$  lines each with at most  $k/2$  points from  $S'$ . Thus, if  $|S'| > k^2/2$ , we have a NO-instance. The case we cannot determine by this preprocessing is when  $|S'| \leq k^2/2$ . That is we have a small kernel as claimed.  $\square$

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

$1 \leq i \leq (k/2 - 1)$ . Symmetrically, for every horizontal line  $l \in L_k$ , we add  $k$  vertical lines  $v_{max}, v_{(max-1)}, \dots, v_{(max-k/2+1)}$  and  $v_{min}, v_{(min+1)}, \dots, v_{(min+k/2-1)}$ . 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  $cover(l)$ . The set  $L'_k$  is  $L_k$  with all these additional lines. The set  $L'_k$  has quadratic size in  $k$ .

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 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 in the set  $I$  of intersections.

**Lemma 8.** *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. 7 (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. 7 (a) deals with the case when the bends share a label<sup>a</sup> 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. 7 (c) illustrates this.  $\square$

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 the previous application. Thus we can apply it repeatedly until there are never disjoint two-line segments in an optimal tour. In particular, we can assume optimal tours have no ears like in Fig. 7 (a).

**Lemma 9.** *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  $R$ .*

<sup>a</sup>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.

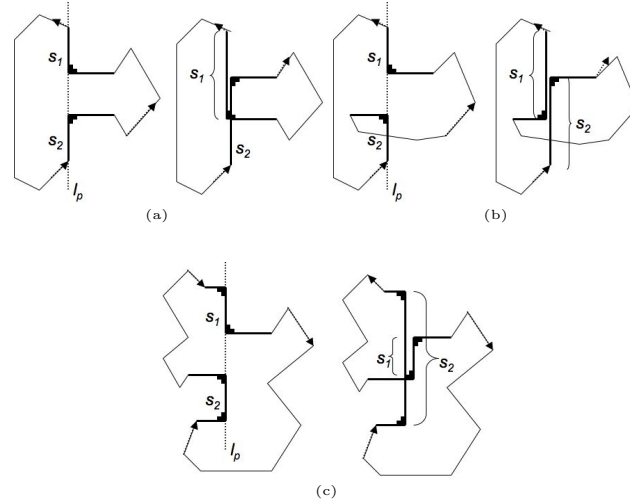


Fig. 7. The tour  $T$  is changed preserving the number of bends.

**Proof.** We will show that for every YES-instance, we can transform the witness tour  $T$  to a tour  $T'$  with the same numbers 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 8, we can assume that if  $s_1, \dots, s_i$  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 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. 8, 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 6) 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 the lines  $h_{max}, h_{(max-1)}, \dots, h_{(max-k/2+1)}$  or  $h_{min}, h_{(min+1)}, \dots, h_{(min+k/2-1)}$  that we have in  $L'_k$ . Since  $h_p \subseteq L'_k$ , we have  $h_p \subseteq R$ .  $\square$

Now, we search exhaustively all tours over the lines in  $R$ . For example, a naive

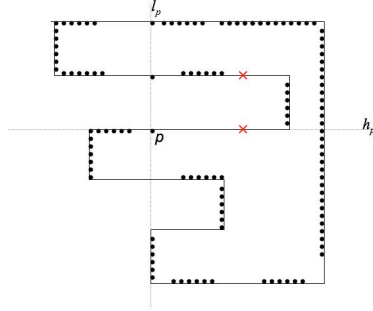


Fig. 8. The point  $p$  is covered by a horizontal line that is not found in the set  $H$  or the set  $L_k$ .

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. 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 results 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.

**Theorem 3.** *The RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM is FPT.*

### 3.2. 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 a candidate 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 test if the candidate set can constitute a tour based on the  $k$  lines in the set. To find a set  $L$  of candidate covering lines, we consider a search tree as illustrated by Fig. 9. In the root of the tree  $L$  is empty. In each internal node of the tree, we choose a point  $p \in S \setminus \text{cover}(L)$  and we explore the two possibilities of a partial answer at  $p$ . Note that if the given instance is a YES-instance, every point in  $S$  is covered by a horizontal line (H), or a vertical line (V) (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 has been 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 meets 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).

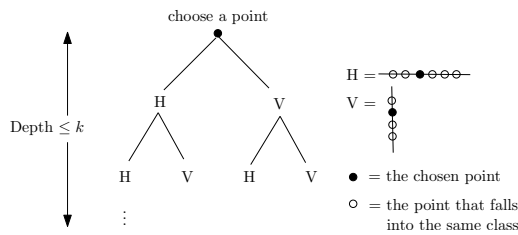


Fig. 9. A search tree for computing candidate sets of  $k$  lines.

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<sup>b</sup>. 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 prove the following lemma.

**Lemma 10.** *If the instance has a tour  $T$  with  $k$  or fewer bends, 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  $lines(T)$  (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  $lines(T)$ .  $\square$

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

<sup>b</sup>If a line-segment in the tour covers no points, it can be translated in parallel until it is placed over one point.

### 3.2.1. Tours where one line-segment covers all the points on the same line

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. 9), results in a candidate set of line-segments since we can simply connect the extreme points in  $\text{cover}(l)$  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,

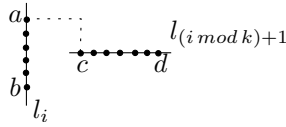


Fig. 10. One of four possible ways of joining two consecutive lines.

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  $i \in \{1, 2, \dots, k\}$ . 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. 10). 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. 10. These extra line-segments cover no points, but they can be added in constant time when constructing the tour. Note that if the total number of line-segments in the final tour exceeds  $k$ , we simply answer no (Lemma 6).

We now analyze the total running time of our algorithm. It is obvious that the search tree for the first phase in Fig. 9 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  $H$ -line and  $V$ -line. 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. 9 is a word of length at most  $k$  with exactly  $k/2$  symbols  $H$  and  $k/2$  symbols  $V$ . Based on this analysis we conclude that the number of different paths from the root to a leaf (and therefore leaves) is given by

$$\binom{k}{k/2} = \frac{k!}{(k/2)!(k/2)!} = O(2^k / \sqrt{k})$$

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 results in time bounded by

$$\begin{aligned} O\left(\frac{2^k}{\sqrt{k}}(k!)(4^k)n\right) &= O\left(\frac{8^k}{\sqrt{k}}\sqrt{2\pi k}(k/e)^kn\right) \\ &= O((2.95k)^kn). \end{aligned}$$

The time complexity is exponential in the parameter but linear in the size of the input. This gives the following result.

**Theorem 4.** *The RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM where one line-segment covers all the points on the same line is FPT.*

### 3.2.2. 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 possible 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 sets  $L$  of no more than  $k$  lines and  $L \subseteq \text{lines}(T)$ . 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  $p \in \text{cover}(l)$  with largest  $y$  coordinate, while  $h_{min}$  is the horizontal line through the point  $p \in \text{cover}(l)$  with 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 11.** *If the instance has a tour  $T$  with  $k$  or fewer bends (and meeting the no distinct type of line-segment condition), 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 tour  $T$  to a tour  $T'$  with the same numbers of bends, where every line-segment covers at least one point and  $\text{lines}(T') \subseteq C$ . 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  $\text{lines}(T) \setminus C$ , 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

20 *V. Estivill-Castro et al.*

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.  $l \in \text{lines}(T) \setminus C$ ). Without loss of generality, assume  $\overline{pq}$  is horizontal (if  $\overline{pq}$  were vertical, we rotate  $S$  and the entire discussion by  $90^\circ$ ). 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 5.** *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  $p' \in \text{cover}(l)$ .*

**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.  $\square$

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).

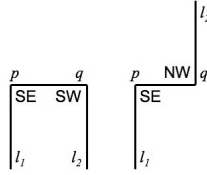


Fig. 11. The line  $l \notin L$  hosts  $\overline{pq}$  from  $T$ .

We will now distinguish two cases (refer to Fig. 11). 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 subcases can be ignored and we can assume the cases are as the two drawings of Fig. 11.

**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  $S \cap \{(x, y) | p_x \leq x \leq q_x \wedge y \geq p_y\}$ <sup>c</sup>, by shifting  $\overline{pq}$  vertically (up) we can (without incrementing the

<sup>c</sup>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  $\overline{pq}$ .

number of bends) overlaps with  $h_{max}$  for some vertical line in  $L$ . In fact, the set  $S \cap \{(x, y) | p_x \leq x \leq q_x \wedge y \geq p_y\}$  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. 12 (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).

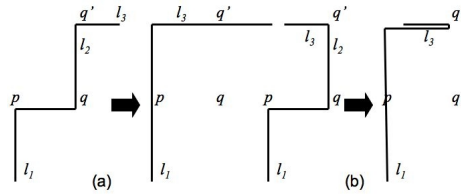


Fig. 12. The subcases if  $l_2 \notin L$  can be converted and eliminate  $l_2$ .

Thus, the turn at  $q'$  must be a SW turn. Fig. 12 (b) shows that a tour with a  $180^\circ$ -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.

**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  $l_2$ . And also, by an analogous argument, there must be points in  $S$  below  $p$  covered  $l_1$ . The tour  $T$  must use a vertical line-segment to cover the points in  $l_2$  below  $q$ ,<sup>d</sup> because the tour complies with the no distinct type of line-segment condition. Assume the tour  $T$  is traveled in the direction  $l_1$ , then  $\overline{pq}$ , then  $l_2$  and

<sup>d</sup>An analogous argument happens if the tour  $T$  uses a vertical line-segment to cover points above  $p$  covered by  $l_1$ .

22 *V. Estivill-Castro et al.*

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. 13).

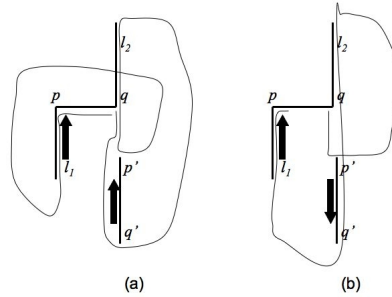


Fig. 13. The tour  $T$  is changed so there is no zig-zag.

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  $180^\circ$ -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  $90^\circ$ -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  $O((43.5k)^kn)$ . This has clearly exponential complexity in  $k$ , but the overall algorithm has polynomial complexity in  $n$  and demonstrates the following result.

**Theorem 5.** *Under the constraint that a tour satisfies the no distinct type of line-segment condition, the RECTILINEAR  $k$ -BENDS TRAVELING SALESMAN PROBLEM is FPT.*

#### 4. Conclusions

We have presented several FPT-algorithms for a hard geometric problem where we have to answer if there is a tour through a set  $S$  of  $n$  points with at most  $k$  bends. We summarize the results of this problem in Table 1. It is apparent that

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

Tour and Its Conditions	Time Complexities
<u>General version</u>	
several line-segments cover points on the same line	$O(kn^2 + k^8 n)$
one line-segment covers all the points on the same line	$O(kn^2 + 4^k k^{4k+2})$
<u>Rectilinear version</u>	
general case (without constraints)	$O(kn^2 + k^4 n)$
same line-segment orientation cover points on the same line.	$O((43.5k)^k n)$
one line-segment covers all the points on the same line	$O((2.95k)^k n)$

the complexity of the algorithms is slightly better as more constraints are placed on the solution. In the rectilinear tour, for 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. The kernelization approach required additional analysis of the witness structure and could be considered to follow the ideas of M. Fellows on extremal structure. 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, and the number  $k'$  of covering lines is bounded by  $O(k)$  where  $k$  is the minimum number of bends. That is, we can approximate the minimum in FPT-time with respect to the sought minimum. Then the decision problem can be used to find the exact value of the minimum in FPT-time as well. We note that to further improve the time complexity of the algorithms we may need further algorithmic engineering. However, even vertex cover was first found FPT by a total impractical argument using the Minors Theorem and steady progress in the parameterized complexity field has now delivered practical implementations. Our aim for this paper is to establish that the NP-complete problems we presented are in the class FPT.

#### References

1. Applegate, D., Bixby, R., Chvátal, V., Cook, W.: The Traveling Salesman Problem. Princeton University Press, Princeton, NJ, USA (2006)
2. Papadimitriou, C.: The Euclidean traveling salesman problem is NP-complete. Theor. Comput. Sci. **4** (1977) 237–244

24 *V. Estivill-Castro et al.*

3. Lee, D., Yang, C., Wong, C.: Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics* **70(3)** (1996) 185–215
4. Wagner, D.: *Path Planning Algorithms under the Link-Distance Metric*. PhD thesis, Dartmouth College, Hanover, NH, USA (February 2006)
5. Arkin, E., Mitchell, J., Piatko, C.: Minimum-link watchman tours. *Inf. Process. Lett.* **86(4)** (2003) 203–207
6. Estivill-Castro, V., Heednacram, A., Suraweera, F.: Advancing NP-complete and FPT results for rectilinear covering problems in higher dimensions. Submitted (April 2009)
7. de Berg, M., van Kreveld, M., Nilsson, B., Overmars, M.: Shortest path queries in rectilinear worlds. *International Journal Computational Geometry and Applications* **2(3)** (1992) 287–309
8. Lee, D., Chen, T., Yang, C.: Shortest rectilinear paths among weighted obstacles. In: 6th ACM Symposium on Computational Geometry (SCG 90), ACM, NY (1990) 301–310
9. Arkin, E., Bender, M., Demaine, E., Fekete, S., Mitchell, J., Sethia, S.: Optimal covering tours with turn costs. *SIAM J. Comput.* **35(3)** (2005) 531–566
10. 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
11. Collins, M.: Covering a set of points with a minimum number of turns. *International Journal Computational Geometry and Applications* **14(1-2)** (2004) 105–114
12. Drysdale, R., Stein, C., Wagner, D.: An  $O(n^{5/2} \log n)$  algorithm for the rectilinear minimum link-distance problem. In: 17th Canadian Conference on Computational Geometry, University of Windsor, Ontario, Canada (2005) 97–100
13. Stein, C., Wagner, D.: Approximation algorithms for the minimum bends traveling salesman problem. In Aardal, K., Gerards, B., eds.: 8th International IPCO Conference. Vol. 2081 of LNCS., Springer, Berlin (June 13-15 2001) 406–422
14. Downey, R., Fellows, M.: *Parameterized Complexity*. Monographs in Computer Science. Springer, New York (1999)
15. Flum, J., Grohe, M.: *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Springer, Berlin (2006)
16. Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications 31. Oxford University Press, New York (2006)
17. Langerman, S., Morin, P.: Cover things with things. *Discrete and Computational Geometry* **33(4)** (2005) 717–729
18. Preparata, F., Shamos, M.: *Computational Geometry An Introduction*. Texts and Monographs in Computer Science. Springer, New York (1985)
19. Guibas, L., Overmars, M., Robert, J.: The exact fitting problem for points. In: 3rd Canadian Conference on Computational Geometry, Simon Fraser University, Vancouver, British Columbia (August 6-10 1991) 171–174
20. Guibas, L., Overmars, M., Robert, J.: The exact fitting problem in higher dimensions. *Computational Geometry: Theory and Applications* **6** (1996) 215–230
21. Megiddo, N., Tamir, A.: On the complexity of locating linear facilities in the plane. *Operations Research Letters* **1(5)** (November 1982) 194–197
22. Kumar, V., Arya, S., Ramesh, H.: Hardness of set cover with intersection 1. In: 27th International Colloquium on Automata, Languages and Programming, ICALP. Vol. 1853 of LNCS., Springer, Berlin (2000) 624–635