Existence of Simple Tours of Imprecise Points

Maarten Löffler

Department of Information and Computing Sciences, Utrecht University

Technical Report UU-CS-2007-003

www.cs.uu.nl
ISSN: 0924-3275

Existence of Simple Tours of Imprecise Points *

Maarten Löffler
loffler@cs.uu.nl

Institute of Information and Computing Sciences
Utrecht University, the Netherlands

Abstract

Assume that an ordered set of imprecise points is given, where each point is specified by a region in which the point may lie. This set determines an imprecise polygon. We show that it is NP-complete to decide whether it is possible to place the points inside their regions in such a way that the resulting polygon is simple. Furthermore, it is NP-hard to minimize the length of a simple tour visiting the regions in order, when the connections between consecutive regions do not need to be straight line segments.

1 Introduction

Geographical Information Systems (GIS) are an important application of computational geometry. In these systems, information about the world is stored as geometric primitives, and geometric algorithms are used to infer information from them. However, traditionally, these algorithms assume their input to be known exactly, while in practice the data in these systems always has some imprecision, because it has been measured from the real world, converted from low-resolution grid data or manipulated by inexact procedures. As a result, artifacts can can occur in the data; for example, a polygon that describes the boundary of some region may have self-intersections.

If we know that the vertices of some polygon are imprecise, and we know the imprecision, the question arises how to place the points such that the resulting polygon indeed has no self-intersections. We show here that it is NP-complete to determine whether such a placement is possible, and hence also to find one if it exists. This is a refinement of an earlier result, where we proved that it is NP-hard to find the minimum perimeter polygon that has no self-intersections [4].

In this paper, we study the following problem. We are given an *ordered* set $\mathcal{S} \subset \mathcal{P}(\mathbb{R}^2)$ of n regions in the plane. We are looking for a tour (closed curve) that visits all regions of $\mathcal{S}$ in the correct order. We call such a tour *simple* if it does not cross itself. We call such a tour *straight* if it is a polygon with a vertex in each region, and no other vertices. We are interested in the existence of a simple straight tour. Figure 1 shows an example of an ordered set of regions and some tours through them.

For non-simple tours, it is easy to see that the shortest one is always straight. This problem has been studied by Dror *et al.* [2], and can be solved in near-linear time if the regions are convex polygons, while it is NP-hard for non-convex regions. For square regions, the shortest and longest straight tours can be computed in $O(n)$ time [4]. If the regions are all adjacent to the inner boundary of a polygon, this problem is known as the Safari Keepers problem [6, 9].

*This research was partially supported by the Netherlands Organisation for Scientific Research (NWO) through the project GOGO.

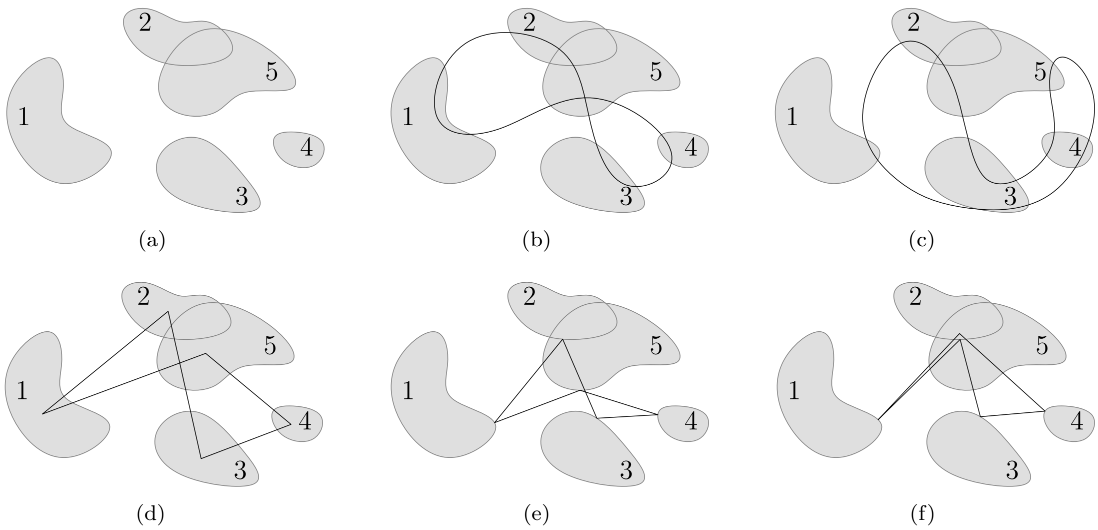


Figure 1: (a) Five regions and an order on them. (b) A tour passing through the regions in order. (c) A simple tour passing through the regions in order. (d) A straight tour passing through the regions in order. (e) The shortest tour passing through the regions in order. (f) The shortest simple tour passing through the regions in order.

On the other hand, if we want to find a simple tour, the shortest one is not always straight. We also prove that finding the shortest simple tour is NP-hard. This answers an open question posed by Polishchuk and Mitchell [7].

The problem of finding the shortest tour for an *unordered* set of regions has been well studied before and is generally called the Traveling Salesman Problem with Neighbourhoods, or (Planar) Group-TSP. This problem is obviously NP-hard. Mata and Mitchell [5] give a constant factor approximation algorithm for some region models; additional results can be found in [1, 8].

The remainder of this paper is organised as follows. The next section contains the NP-hardness proof for the problem of deciding whether a simple straight tour through an ordered set of vertical line segments exists. Section 3 shows how to extend this result to other imprecision regions like circles. Section 4 describes the NP-hardness of the problem of finding a shortest simple tour. Finally, some concluding remarks are given in Section 5.

2 Simple Straight Tours through Vertical Line Segments

Given a set of parallel line segments and a cyclic order on them, we want to choose a point on each segment such that the polygon determined by those points in the given order is simple. The decision problem of whether this is possible is NP-complete. The problem is trivially in NP, and we prove NP-hardness by reduction from planar 3-SAT [3], see Figure 2. For different components of a planar 3-SAT instance, we construct polygonal chains that will be connected into a polygon in the end. A simple polygon can be realised if and only if the 3-SAT instance is satisfiable. In the construction we give here, the simple polygon will be a degenerate one if it exists. A degenerate simple polygon is a polygon for which it is possible to move all vertices over an arbitrarily small distance to make it into a simple polygon, see Figure 3. However, we will show that the gadgets can be adapted slightly to also allow non-degenerate simple polygons.

We represent variables by scissor gadgets as in Figure 4(a). This gadget consists of two imprecise points and two precise (or degenerate imprecise) points. The dashed lines depict the order in

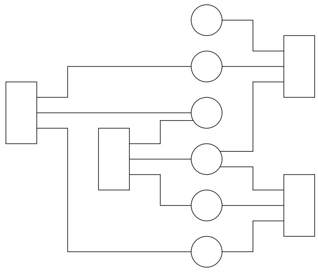


Figure 2: An instance of planar 3-SAT. The circles represent variables, the rectangles represent clauses.

which the tour should visit these regions. There are two possible ways to make a simple straight tour through this gadget, which represent the two different values of a variable. The solution with a positive sloping diagonal, see Figure 4(b), represents the value `True`, and the negative sloping diagonal, see Figure 4(c), represents the value `False`. In the remainder of the proof we use a schematic drawing for this configuration of four imprecise points, see Figure 4(d).

We can make a chain of scissor gadgets that all represent the same variable, as shown in Figure 5(a). Here each scissor gadget is represented schematically as two crossing diagonal lines and a horizontal line indicating where the gadget is connected to the polygon. For each scissor gadget one of the legs is used in a solution, the other is not. There are only two possible states to this chain; either all of the scissor gadgets use their positive sloping leg or they all use their negative sloping leg.

We can also split this chain into more chains with a junction as shown in Figure 5(b). Here scissor gadgets of two different sizes are used, but still there are only two possible states in the total structure. The chains can be split again to make as many chains for a variable as needed. Furthermore, we can make chains under a slope of almost 45°, and by zigzagging between junctions

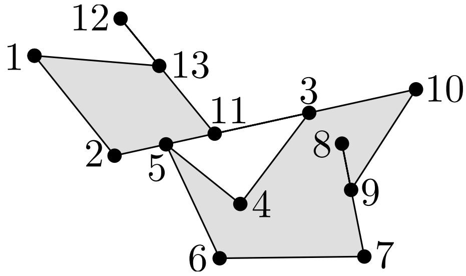
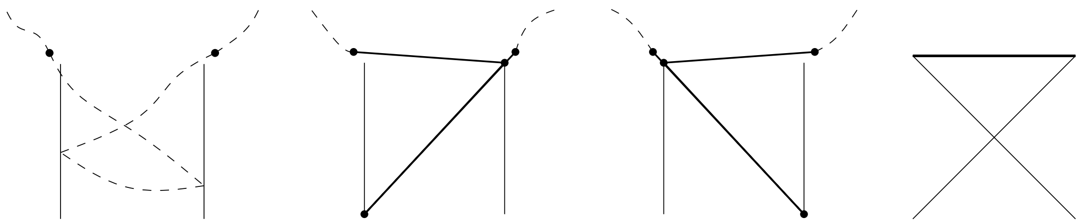


Figure 3: A degenerate simple polygon can touch itself in vertices or along edges, but has no crossings. The numbers give the order of the vertices.



(a) (b) (c) (d)

Figure 4: (a) The input for a pair of scissors. (b) One of the solutions, representing the state `True`. (c) The other solution, representing the state `False`. (d) Schematic representation.

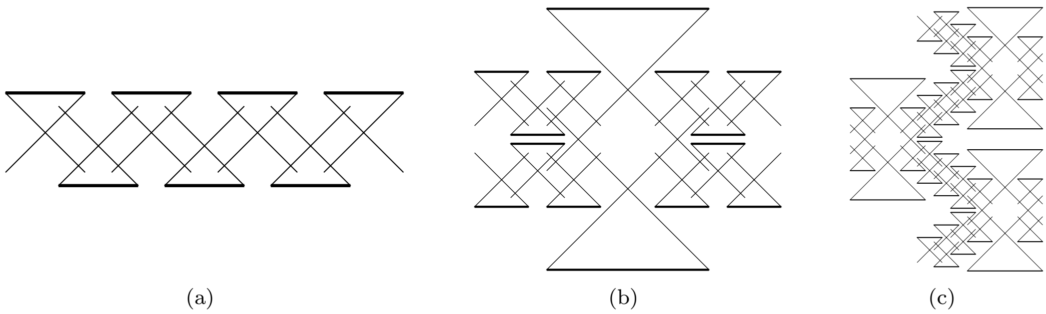


Figure 5: (a) A chain of scissors. (b) A junction to split the chain of scissors. (c) Going vertical.

we can move over vertical distances, see Figure 5(c).

We represent the clauses of the 3-SAT formula by clause gadgets, as in Figure 6(a). This configuration consists of one imprecise point and four precise points. For clauses, there are three unconnected polygonal chains that visit the gadget, which are represented by dashed lines. The three possible solutions for this situation can be seen in Figures 6(b), 6(c) and 6(d). The idea is that in order to find a total solution, at least one of the three solutions to the clause must be possible. For example, if we want to build the clause `a` $\vee$ `b` $\vee$ $\neg$`c`, we intersect the negative sloping leg of the variable `a` with one of the three solution paths of the clause, the negative sloping leg of the variable `b` with another path, and finally the positive sloping leg of `c` with the remaining path, and the clause can be solved if and only if the logical clause is satisfied, see Figure 6(e).

Now that we have structures for variables and clauses, we can build an instance of planar 3-SAT

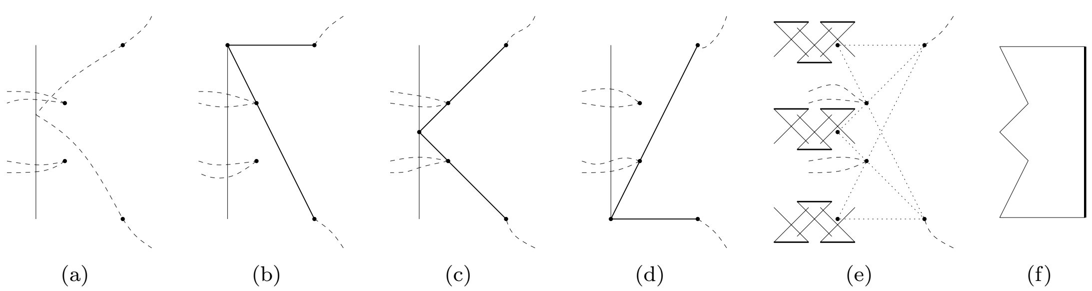


Figure 6: (a) The input for a clause. (b) One of the solutions. (c) Another solution. (d) The third solution. (e) The clause attached to the three variables. (f) Schematic representation.

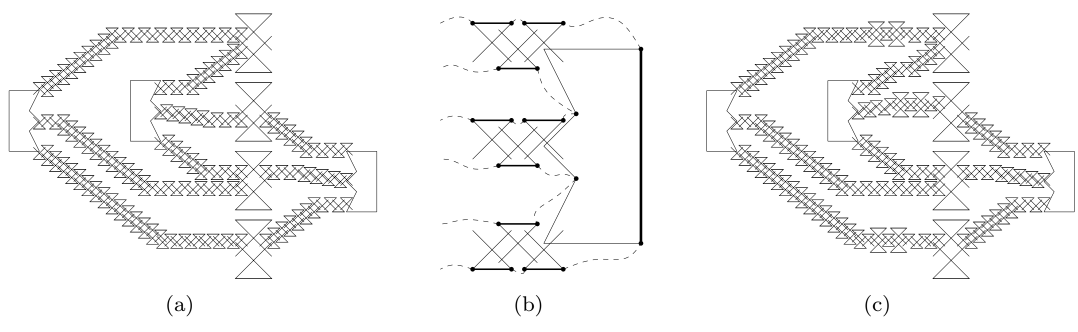


Figure 7: (a) Part of a network of variables and clauses to represent planar 3-SAT. (b) Connecting the gadgets. (c) The network contains bridges to connect cycles.

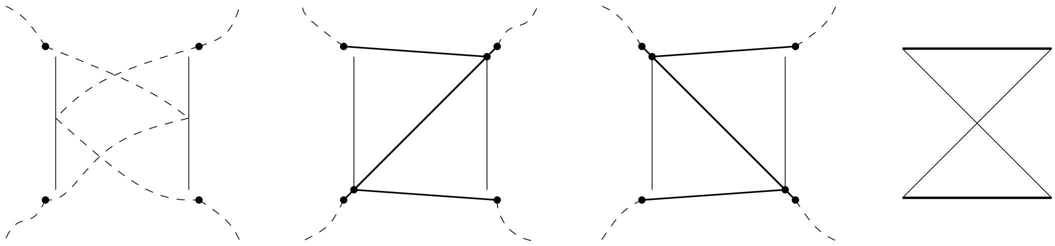


Figure 8: (a) The input for a bridge. (b) One of the solutions, representing the state `True`. (c) The other solution, representing the state `False`. (d) Schematic representation.

by embedding the graph in the plane and making it wide enough to fit all the structures such that they do not interfere. An example of a (part of a) resulting structure can be seen in Figure 7(a). However, this does not complete our construction yet. The scissor and clause gadgets have some precise points where the tour is supposed to enter and leave the gadget. We need to construct a tour that visits all gadgets in any order, but in such a way that it does not interfere with the gadgets. We can easily do this by linking neighbouring gadgets together, see Figure 7(b). However, doing this will result in a number of smaller tours instead of one big tour, because the 3-SAT instance partitions the plane into a number of faces. We need one tour to visit all gadgets, and therefore all faces. This means we need to cross the scissor chains.

We design another primitive, see Figure 8(a). This bridge gadget consists of two imprecise points and four precise points, and has two chains passing through it, again indicated by the dashed lines. Like the scissor gadget, the bridge gadget has two possible solutions representing the values `True` and `False` of a variable, see Figures 8(b) and 8(c). A schematic representation is shown in Figure 8(d). Bridge gadgets can be embedded in chains of scissor gadgets, and they preserve the property that the whole chain uses either positive or negative sloping legs, see Figure 9. However, we now have two parts of the tour that cross the chain.

Now we can include bridges into the network such that all faces of the embedded planar 3-SAT graph are connected by bridges, see Figure 7(c). All we need to do now is connect the fat edges to each other with a fixed part of the tour (a part that only contains precise points), and we have a valid input for the problem. The number of imprecise points in the construction is clearly polynomial in the length of the 3-SAT instance, which completes the proof.

Theorem 1 *Given an ordered set of n vertical line segments, it is NP-hard to decide whether it is possible to choose a point on each segment such that the resulting polygon is simple.*

It is easy to adapt the gadgets slightly to also allow non-degenerate polygons, without damaging the proof. For the scissor gadgets, just make the vertical line segments slightly longer; for the clauses, move the two central precise points slightly towards the imprecise point.

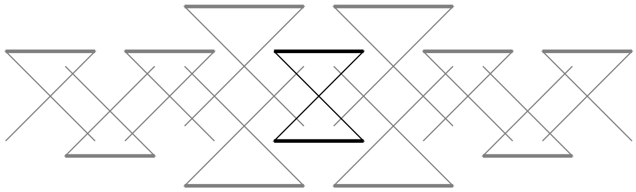


Figure 9: A bridge embedded in a chain of scissors.

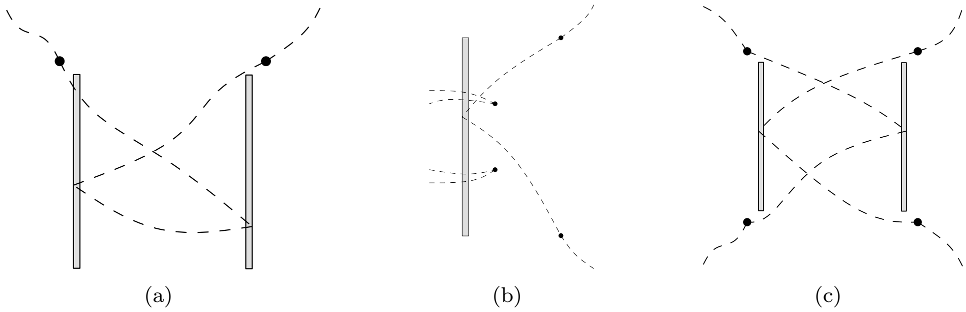


Figure 10: Replacing the line segments by narrow rectangles does not affect the gadgets.

3 Simple Straight Tours through General Regions

If we model the imprecise points as circles, squares, or any other connected region, the problem is still NP-complete.

We can adapt the proof by using vertical axis-aligned rectangles of aspect ratio 1 : 32 instead of vertical line segments. The three basic gadgets remain conceptionally the same, see Figure 10. The two (or three, in the case of the clause gadget) different possible solutions for the gadgets have some more freedom now, but they remain clearly distinct. To make chains of scissors, junctions, and connections between chains and clauses the gadgets need to be placed more carefully to avoid the possibility of two parts of the tour that should intersect escaping each other, but this is clearly possible. As a consequence, the problem for rectangles of aspect ratio 1 : 32 is also NP-hard.

Of course, the proof also works if there are some rectangles with even higher aspect ratios than 1 : 32. Now suppose the input is a general set of axis-aligned rectangles. Let r be the largest ratio between the width and height of any rectangle. Now scale the plane in the x-direction with a factor $\frac{1}{32r}$, and the input has become a set of rectangles with aspect ratio 1 : 32 or higher. Note that the existence of a simple tour is not affected by this scaling operation.

If we model the points as scaled copies of any connected shape, for example as circles or regular polygons, the same proof can also be used. Again we scale the plane to bound the bounding box of the shape, see Figure 11. Now we have narrow rectangles where we can place points only at some subset of its interior. However, we know that for any y-value that intersects a given rectangle, there is at least one point at that y-value that we can select, and that is enough to make the gadgets work.

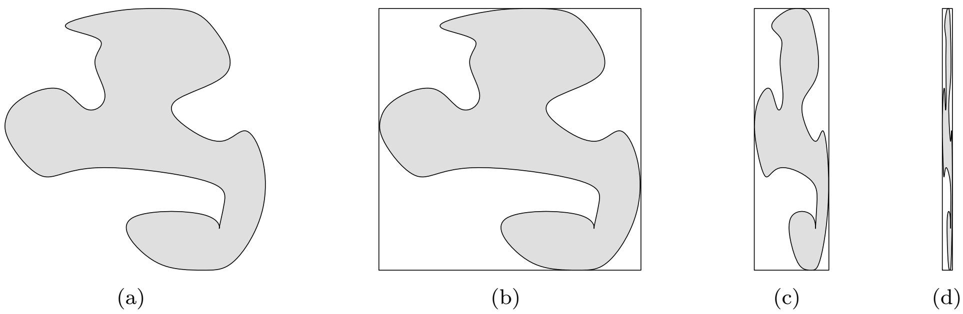


Figure 11: (a) A general region. (b) The region in its bounding box. (c, d) The region scaled to a narrow rectangle.

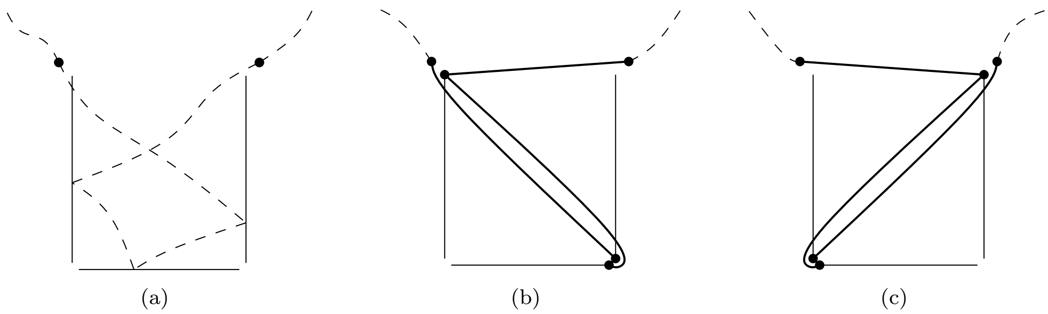


Figure 12: Adapted scissors.

Theorem 2 *Given an ordered set of n arbitrarily scaled copies of any connected region, it is NP-hard to decide whether it is possible to choose a point in each region such that the resulting polygon is simple.*

4 Simple Tours through Line Segments

If we drop the requirement that the edges between two consecutive points need to be straight line segments, a simple tour always exists. In this context, it is interesting to consider *shortest* tours. Finding such a shortest tour is also NP-hard.

In this case, we need to make slightly more complicated gadgets. In the original scissor gadget, we implicitly used the fact that all connections need to be straight line segments, in order to ensure that the tour goes down to the bottom end of one of the two vertical line segments. When the connections do not need to be straight, we need to explicitly ensure that the tour goes down, so we add a horizontal segment, see Figure 12.

We also need to adapt the clause gadget. We now don't require the solutions to be straight, so there are always three solutions. However, we need to ensure that all three solutions have exactly the same length. We can do this by moving the two central precise points, see Figure 13. The three solutions all touch the imprecise point in a different point, and we can connect the variables to the clause as before.

The bridge gadget does not need to be adapted. With these adapted gadgets, the proof still works in essentially the same way. If the 3-SAT instance is satisfiable, a tour exists that is considerably shorter than if the instance is not satisfiable.

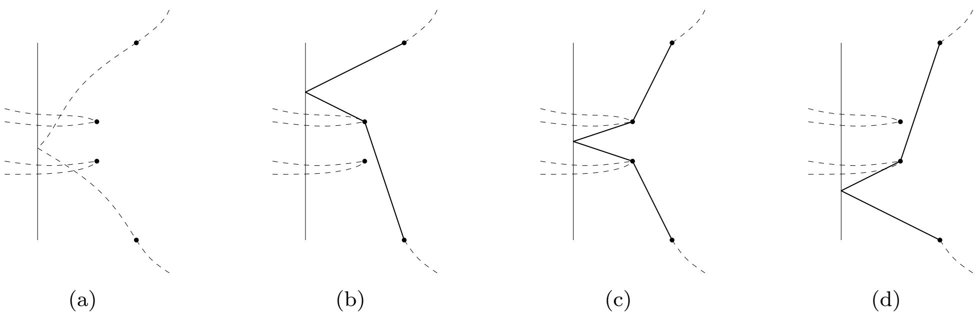


Figure 13: Adapted clause.

Because we used a horizontal segment as well as vertical segments in the adapted scissor gadget, this proof cannot be extended to more specific imprecision regions in the same way as in Section 3. However, we can extend the proof to squares by noting that in all gadgets the given segments might as well be sides of squares, without allowing any shorter solutions.

Theorem 3 *Given an ordered set of n axis-parallel line segments or squares, it is NP-hard to find a tour that visits all segments or squares in order such that this tour is simple and as short as possible.*

5 Conclusions

We studied the problem of finding a simple straight tour through a sequence of regions, and proved that it is NP-complete to decide whether this is possible. We also proved that it is NP-hard to find the shortest non-straight tour, resolving an open problem from [7].

It still remains open whether a shortest simple tour for a set of single points, instead of regions, can be found efficiently.

Acknowledgements

The author thanks Marc van Kreveld for helpful discussions.

References

[1] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, and A. F. van der Stappen. TSP with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005.

[2] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM Sympos. Theory Comput.*, pages 473–482, 2003.

[3] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[4] M. Löffler and M. van Kreveld. Largest and smallest tours and convex hulls for imprecise points. In *Proc. 10th Scandinavian Workshop on Algorithm Theory*, LNCS 4059, pages 375–387, 2006.

[5] C. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1995.

[6] S. Ntafos. Watchman routes under limited visibility. *Comput. Geom. Theory Appl.*, 1(3):149–170, 1992.

[7] V. Polishchuk and J. S. B. Mitchell. Touring convex bodies - a conic programming solution. In *Proc. 17th Canad. Conf. on Comput. Geom.*, pages 290–293, 2005.

[8] S. Safra and O. Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. *Computational Complexity*, 14(4):281 – 307, 2006.

[9] X. Tan and T. Hirata. Finding shortest safari routes in simple polygons. *Information Processing Letters*, 87:179–186, 2003.