

Homotopic rectilinear routing with few links and thick edges

Citation for published version (APA):

Speckmann, B., & Verbeek, K. A. B. (2009). Homotopic rectilinear routing with few links and thick edges. 243-246. Abstract from 25th European Workshop on Computational Geometry (EuroCG 2009), Brussels, Belgium.

Document status and date:

Published: 01/01/2009

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Homotopic Rectilinear Routing with Few Links and Thick Edges*

Bettina Speckmann[†]Kevin Verbeek[†]

Abstract

We study the problem of finding non-crossing thick minimum-link rectilinear paths homotopic to a set of input paths in an environment with rectangular obstacles. This problem occurs in the context of map schematization under geometric embedding restrictions, for example, when schematizing a highway network for use as a thematic layer. We present a 2-approximation algorithm that runs in $O(n^3 + k_{in} \log n + k_{out})$ time, where n is the total number of input paths and obstacles and k_{in} and k_{out} are the total complexities of the input and output paths, respectively. Our algorithm not only approximates the minimum number of links, but also minimizes the total length of the paths. An approximation factor of 2 is optimal when using smallest paths as lower bound.

1 Introduction

Motivation. Schematic maps are a well-known cartographic tool; they visualize a set of nodes and edges (for example, highway or metro networks) in simplified form to communicate connectivity information as effective as possible. Many schematic maps are somewhat removed from the geographic context: while the locations of nodes are usually close to their actual geographic location, edges can be routed in any way that is consistent with the network topology. However, if one wishes to use a schematized network as a thematic layer for a thematic map, then additional geographic and hence geometric restrictions apply. Consider the following example: we want to create a thematic map that focuses on traffic flow on highways. As base map we use a standard geographic map, the thematic overlay consists of a schematized highway network. The schematized network has to fulfill various requirements: (i) edges are drawn thickly, using few orientations and links, (ii) critical features (cities, lakes, etc.) of the base map are not obscured and retain their correct topological position with respect to the network. There has been little algorithmic work on network schematization under geometric embedding restrictions; in this paper we address one of the fundamental underlying problems for the first time.

Our input is a set of n_o rectangular obstacles and n_p pairs of points (a_i, b_i) together with non-crossing paths π_i that connect a_i to b_i . We want to find a set of n_p non-crossing rectilinear thick a_i - b_i paths which are homotopic to the paths π_i . We refer to this problem as the *thick routing problem* (correspondingly, the *thin routing problem* is the same problem for standard “thin” paths). Returning to the example above, the obstacles model the (bounding boxes of) critical map features and the paths π_i correspond to the actual geographic location of the highways we want to schematize. Note that the endpoints of the paths—the cities connected by the highways—necessarily also constitute obstacles when considering homotopy.

Related Work. The thick routing problem can be seen as a variation on the *thick non-crossing paths problem* studied by Mitchell and Polishchuk [11]. They find shortest non-crossing thick paths in a polygonal domain, that is, the points a_i and b_i lie on the boundary of a simple polygon. We consider general input paths, albeit with fixed homotopy classes, and study minimum-link rectilinear instead of shortest paths. There are several papers [2, 6] that find shortest paths homotopic to a given collection of input paths. However, while a set of shortest paths homotopic to a set of non-crossing input paths is necessarily non-crossing, the same does not hold for minimum-link rectilinear paths. Our problem is also related to *drawing graphs with fat edges* [5], and to *wire routing* in VLSI design [4, 7, 9, 10], although none of these papers strives to minimize the number of links.

Many variants of the thick routing problem, even without obstacles and with thin paths, are proven NP-hard by Bastert and Fekete [1] if the homotopy classes of the paths are not specified. Yang *et al.* [14] consider a very restricted version: they show how to find a pair of non-crossing minimum-link rectilinear paths inside a rectilinear polygon. Pach and Wenger [13] prove that non-crossing paths connecting n pairs of points might need $\Omega(n)$ paths to have $\Omega(n)$ links. Gupta and Wenger [8] present an approximation algorithm (with an approximation factor larger than 120) for finding non-crossing minimum-link paths inside a simple polygon; here all endpoints lie on the boundary of the polygon. Another type of related work considers map schematization and metro map construction. For example, Cabello *et al.* [3] give an algorithm that schematizes a network using 2 or 3 links per path, if possible. Nöllenburg and Wolff [12]

*Supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

[†]Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands, speckman@win.tue.nl and k.a.b.verbeek@tue.nl

use a method based on mixed-integer programming to generate metro maps using one edge per path. Both methods do not incorporate obstacles and are restricted to a small constant number of links per path.

Results. We present a 2-approximation algorithm for the thick routing problem which runs in $O(n^3 + k_{in} \log n + k_{out})$ time, where $n = n_o + n_p$ is the total number of obstacles and input paths and k_{in} and k_{out} are the total complexities of the input and output paths, respectively. As a lower bound for the minimum number of links any solution must have we use the total number of links of the *smallest paths* that are homotopic to the input paths (a smallest path is a rectilinear path that is both shortest and minimum-link). Our algorithm not only approximates the minimum number of links, but also minimizes the total length of the paths. An approximation factor of 2 is optimal when using smallest paths as lower bound.

Our algorithm is based on a surprisingly simple incremental construction, which we first explain for thin paths in Section 3, before extending it to thick paths in Section 4. Intuitively, any incremental approach to this problem should be doomed due to a cascading number of links, but we show how to move already inserted paths without increasing the number of links or creating crossings to make room for each new path. To efficiently move homotopic (sub-)paths we use a bundling technique as described in [6]. Furthermore, we “grow” the thick paths by extending the results of [5] to rectilinear paths and rectangular obstacles.

2 Preliminaries

Every rectilinear path π consists of a sequence of horizontal and vertical links. We can also see a rectilinear path as a sequence of staircase chains separated by U-turns. We distinguish two types of staircase chains: *positive staircase chains* which go right and down (or left and up) and *negative staircase chains* which go left and down (or right and up). If there is an obstacle on the inside of a U-turn touching the middle link, then we call this U-turn a *tight U-turn*.

A *smallest path* is a rectilinear path that is both shortest and minimum-link. Every staircase chain between two points has the same length. Hence a rectilinear path is shortest iff it has only tight U-turns. Every collection of non-crossing rectilinear paths can be made shortest by making the U-turns tight one-by-one. So we can find a solution to the thin (or thick) routing problem that is also shortest. However, rectilinear shortest paths are not unique and can cross. Therefore we consider *rectilowest paths*, which are rectilinear shortest paths of which all staircase chains are as low as possible. *Rectihighest paths* are defined similarly. We can show that rectilowest paths or rectihighest paths are always non-crossing. Unfortunately they can have $O(n)$ times more links than smallest

paths, so we use smallest paths as a starting point for the algorithm instead. We observe that a positive and a negative staircase chain of rectilinear shortest paths can never cross.

Consider a collection of non-crossing rectilinear y-monotone chains π_i and a collection of obstacles ω_i . We can represent the homotopy classes of π_i by a total order \mathcal{O} on the paths and obstacles. Use $\mathcal{O}(\pi_i)$ for the position of π_i in this order. Let π'_i and π'_j be paths homotopic to π_i and π_j ($\pi'_i \sim_h \pi_i$ and $\pi'_j \sim_h \pi_j$). If $\mathcal{O}(\pi_i) < \mathcal{O}(\pi_j)$, then an *intersection region* of π'_i and π'_j is a region enclosed by π'_i and π'_j at y -coordinates where the paths are out of order, i.e. where π'_j is to the left of π'_i . Intersection regions are y -monotone rectilinear polygons and do not contain obstacles.

3 Thin Paths

We solve the thin routing problem in two steps. First we compute the smallest paths homotopic to the input paths π_i . Then we untangle these paths.

Computing smallest paths. We compute *pushed* smallest paths that are homotopic to the input paths. A *pushed* rectilinear shortest path has all horizontal links pushed down as much as possible (while keeping it shortest) and all vertical links pushed right (left) as much as possible for positive (negative) staircase chains. Pushed smallest paths have only rectangular intersection regions, which simplifies the untangling algorithm. To compute them efficiently, we first compute rectilowest and rectihighest paths. For that we modify the algorithm by Efrat *et al.* [6], which also bundles homotopic y -monotone chains. The rectilowest and rectihighest paths are represented by $O(n)$ bundles with at most $O(n)$ links each. From that we can easily compute a bundled representation of the pushed smallest paths using a plane sweep algorithm. In total this step takes $O(n^2 + k_{in} \log n)$ time.

Untangling smallest paths. To untangle the smallest paths, we use the fact that positive staircase chains do not cross negative staircase chains. Hence we can restrict ourselves to positive staircase chains. As these chains have a total order, we assume the input paths to be ordered positive staircase chains π_i ($1 \leq i \leq n$).

We untangle the paths incrementally, adding paths in order from left to right. The first two paths π_1 and π_2 can have only rectangular intersection re-

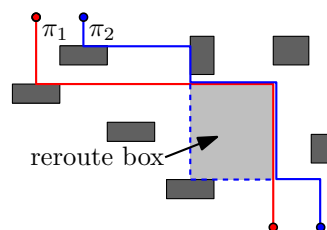


Figure 1: Untangling the first two paths.

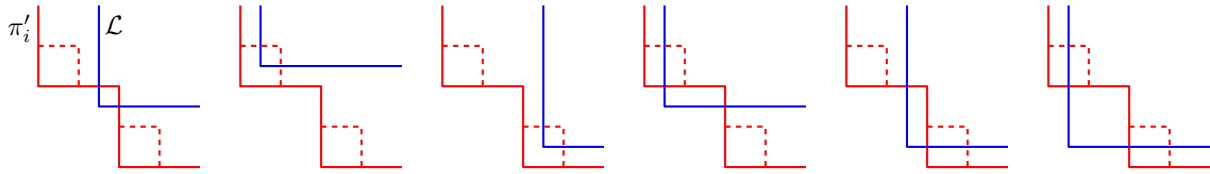


Figure 2: The possible shapes of intersection regions of \mathcal{L} and a path π'_i . Dashed lines denote reroute boxes.

gions. We can add two links to π_2 (for each intersection region) to remove the crossings. Unfortunately this might make intersection regions with π_k ($k > 2$) non-rectangular such that rerouting requires more links. To avoid this we keep track of where paths are rerouted using *reroute boxes* (Fig. 1). Reroute boxes are rectangular and contain no obstacles.

We maintain the following invariants after adding path π_k (rerouted paths are denoted by π'_i): (i) $\pi'_1 \dots \pi'_k$ are non-crossing, (ii) $\pi_1 \dots \pi_k$ have at most one reroute box per lower-left bend, and (iii) all upper-right bends of a path π_j are to the right of π'_i with $i < j$. A new path π_{k+1} is added one L-segment \mathcal{L} at a time. \mathcal{L} can cross multiple other paths. The original paths π_i have only rectangular intersection regions. \mathcal{L} is unchanged and we have added only reroute boxes to $\pi'_1 \dots \pi'_k$, hence there are only a few shapes possible for the intersection regions with \mathcal{L} (Fig. 2): one, two, or three upper-right bends.

Now assume that \mathcal{L} is crossed by the paths $\pi'_a \dots \pi'_b$ ($a \leq b$). We want to make the intersection region of \mathcal{L} with π'_b rectangular such that we can add a reroute box to \mathcal{L} to remove all crossings. We do this incrementally for all intersection regions, starting with π'_a and ending with π'_b . Assume an intersection region \mathcal{R}_i of \mathcal{L} and $\pi'_i \in [\pi'_a \dots \pi'_b]$ is rectangular. To make the intersection region \mathcal{R}_{i+1} rectangular, we move the links of π'_{i+1} through \mathcal{R}_{i+1} (intersection regions contain no obstacles), but not through \mathcal{R}_i (this would introduce crossings with π'_i). By Invariant (i) π'_{i+1} does not cross \mathcal{R}_i initially. If \mathcal{R}_{i+1} contains one upper-right bend, we do nothing as \mathcal{R}_{i+1} is already rectangular. If \mathcal{R}_{i+1} contains two upper-right bends, then we either move the first vertical link of π'_{i+1} to the left (onto \mathcal{L}) or the last horizontal link down. Due to the rectangular shape of \mathcal{R}_i , we can always do one of the two moves without crossing \mathcal{R}_i . If \mathcal{R}_{i+1} contains three upper-right bends, then either \mathcal{R}_i is in the middle corner and we move the first vertical link of π'_{i+1}

to the left and the last horizontal link down, or \mathcal{R}_i is not in the middle corner, in which case we can simplify the middle corner and handle this as a case with two upper-right bends (see Fig. 3 for an example).

Since the algorithm maintains all invariants, the output paths are non-crossing. They are also homotopic to the input paths, because we change paths only by adding reroute boxes or by moving links through intersection regions. We add only two links per reroute box and there is at most one reroute box per lower-left bend. Because a positive staircase chain with L links has at most $L/2$ lower-left bends, the algorithm at most doubles the number of links.

We can easily extend this algorithm to work on the y -monotone chains of the smallest paths. Because we can add an L-segment in $O(n)$ time and we have only $O(n)$ chains with at most $O(n)$ links each, untangling the chains can be done in $O(n^3)$ time. Finally we can unbundle the paths in $O(k_{out})$ time. So in total we can compute a 2-approximation for the thin routing problem in $O(n^3 + k_{in} \log n + k_{out})$ time.

4 Thick Paths

In this section we extend the algorithm of the previous section to thick paths. For $\Delta > 0$ let \mathcal{S}_Δ be the square of size Δ centered at the origin. A *thick rectilinear path* with *spine* π_i and thickness Δ_i is defined as the Minkowski sum of π_i and \mathcal{S}_{Δ_i} : $(\pi_i)^{\Delta_i} = \pi_i \oplus \mathcal{S}_{\Delta_i}$. We say that two paths π_i and π_j have the *proper distance* if $(\pi_i)^{\Delta_i}$ and $(\pi_j)^{\Delta_j}$ are interior disjoint.

The main difference with thin paths is that fixing a thick path might make it impossible for another thick path to be routed. We say that a path $\pi'_i \sim_h \pi_i$ is *feasible* if there exist paths $\pi'_j \sim_h \pi_j$ for $1 \leq j \leq n_p$ and $i \neq j$ such that all paths have the proper distance. In a valid solution of the thick routing problem every path is feasible. Hence we use feasible smallest paths as a lower bound.

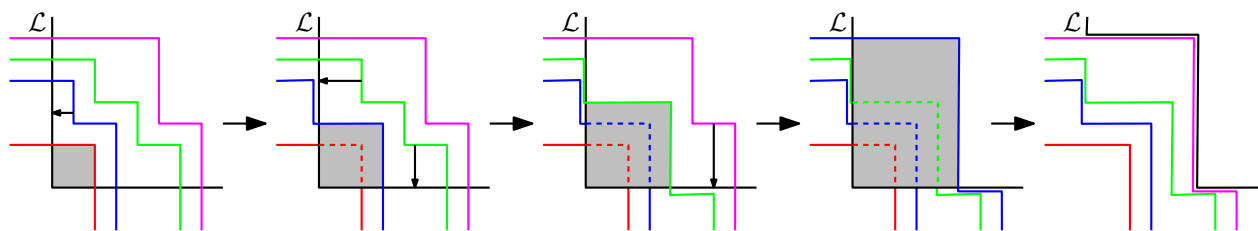


Figure 3: Making intersection regions rectangular and rerouting \mathcal{L} .

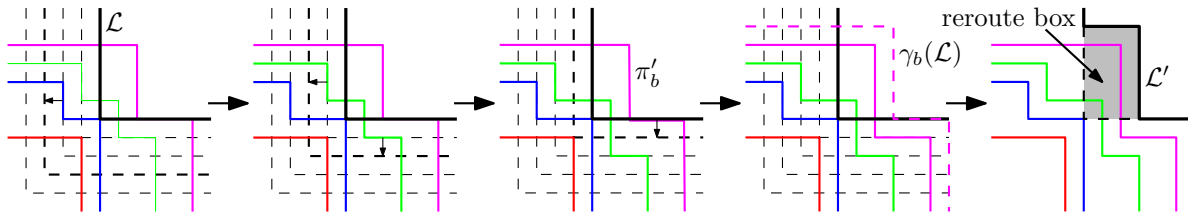


Figure 4: Making guarded regions rectangular.

Feasible smallest paths. To compute feasible smallest paths, we first compute the feasible rectilowest and rectihighest paths. For this we use a variant of the algorithm by Duncan *et al.* [5]. Like in Section 3, we first compute the thin rectilowest and rectihighest paths. Then, using a similar approach as in [5], we grow the paths until they have the required thickness. By construction the resulting paths are the feasible rectilowest and rectihighest paths. After that it is easy to compute feasible smallest paths. Like for thin paths, we bundle homotopic y-monotone chains, so we get $O(n)$ bundles with at most $O(n)$ links each. This can be computed in $O(n^3 + k_{in} \log n)$ time.

Untangling thick paths. The thick paths need to be untangled in such a way that the resulting paths are feasible and have the proper distance. To achieve this, we use *guards*. A *guard* $\gamma_i(j)$ of a path π_i with respect to another path π_j is a path representing the closest position at which the path π_j can be with respect to π_i while all paths have the proper distance. A guard ignores all obstacles. In other words, the guard $\gamma_i(j)$ defines the region where π_j is feasible, if we fix the path π_i and ignore the obstacles. This way a guard $\gamma_i(j)$ is a shifted version of π_i , where the distance depends on the thickness of the paths between π_i and π_j . This distance can differ along the path π_i .

To ensure that we do not change the homotopy classes during untangling, we use the intersection regions between a guard $\gamma_i(j)$ and a path π_j , which we call *guarded regions*. We can show that also guarded regions contain no obstacles and, if the paths are pushed, all guarded regions are rectangular as well.

We now untangle the paths as before, changing two of the invariants: (i') $\pi'_1 \dots \pi'_k$ have the proper distance and (iii') upper-right bends of a guard $\gamma_j(i)$ are to the right of paths π'_i with $i < j$. When adding an L-segment \mathcal{L} we now look at the guarded regions of $\gamma_{\mathcal{L}}(i)$ and π'_i instead of the ordinary intersection regions. Computing the guards and guarded regions is nontrivial and the details can be found in the full paper. Next, we have to make the guarded regions rectangular, which can be done similarly as for thin paths. While doing this, we push the links of a path π'_i onto $\gamma_{\mathcal{L}}(i)$ instead of \mathcal{L} . This way Invariant (i') is maintained by definition of the guards. Finally we can add a reroute box to \mathcal{L} , such that \mathcal{L} has the proper distance to the other paths (see Fig. 4).

Theorem 1 We can compute a 2-approximation for the thick routing problem that also minimizes the lengths of the paths in $O(n^3 + k_{in} \log n + k_{out})$ time.

References

- [1] O. Bastert and S. P. Fekete. Geometrische Verdrahtungsprobleme. Technical Report 247, Mathematisches Institut, Universität zu Köln, 1996.
- [2] S. Bessamyatnikh. Computing homotopic shortest paths in the plane. *J. Alg.*, 49(2):284–303, 2003.
- [3] S. Cabello, M. de Berg, and M. van Kreveld. Schematization of networks. *Comp. Geom.: Theory and Appl.*, 30(3):223–238, 2005.
- [4] R. Cole and A. Siegel. River routing every which way, but loose. In *Proc. 25th Symp. Found. Comp. Sci.*, pp. 65–73, 1984.
- [5] C. A. Duncan, A. Efrat, S. G. Kobourov, and C. Wenk. Drawing with fat edges. *Intern. J. Found. Comp. Sci.*, 17(5):1143–1163, 2006.
- [6] A. Efrat, S. G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. *Comp. Geom.: Theory and Appl.*, 35(3):162–172, 2006.
- [7] S. Gao, M. Jerrum, M. Kaufmann, K. Mehlhorn, and W. Rülling. On continuous homotopic one layer routing. In *Proc. 4th Symp. Comp. Geom.*, pp. 392–402, 1988.
- [8] H. Gupta and R. Wenger. Constructing pairwise disjoint paths with few links. *ACM Trans. Alg.*, 3(3):26, 2007.
- [9] C. E. Leiserson and F. M. Maley. Algorithms for routing and testing routability of planar vlsi layouts. In *Proc. 17th Symp. Theory Comp.*, pp. 69–78, 1985.
- [10] M. Malley. *Single-layer wire routing and compaction*. MIT Press, Cambridge, MA, USA, 1990.
- [11] J. S. Mitchell and V. Polishchuk. Thick non-crossing paths and minimum-cost flows in polygonal domains. In *Proc. 23rd Symp. Comp. Geom.*, pp. 56–65, 2007.
- [12] M. Nöllenburg and A. Wolff. A mixed-integer program for drawing high-quality metro maps. In *Proc. 13th Intern. Symp. Graph Drawing*, LNCS 3843, pp. 321–333, 2005.
- [13] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17(4):717–728, 2001.
- [14] C. D. Yang, D. T. Lee, and C. K. Wong. The smallest pair of noncrossing paths in a rectilinear polygon. *IEEE Trans. Comp.*, 46(8):930–941, 1997.