Homotopic $\mathcal{C}$-Oriented Routing

Kevin Verbeek*

Dep. of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract. We study the problem of finding non-crossing minimum-link $\mathcal{C}$-oriented paths that are homotopic to a set of input paths in an environment with $\mathcal{C}$-oriented obstacles. We introduce a special type of $\mathcal{C}$-oriented paths—*smooth paths*—and present a 2-approximation algorithm that runs in $O(n^2(n + \log\kappa) + k_{in}\log n)$ time, where n is the total number of paths and obstacle vertices, k_{in} is the total number of links in the input, and $\kappa = |\mathcal{C}|$. The algorithm also computes an $O(\kappa)$-approximation for general $\mathcal{C}$-oriented paths. As a related result we show that, given a set of $\mathcal{C}$-oriented paths with L links in total, non-crossing $\mathcal{C}$-oriented paths homotopic to the input paths can require a total of $\Omega(L\log\kappa)$ links.

1 Introduction

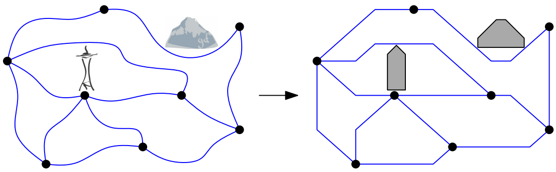

Fig. 1. Input and output

Schematic maps are an important tool in the field of cartography; they are used to visualize networks (like metro or road networks) in a highly simplified form, omitting details that are not relevant to the information the map is intended to convey. Edges of a schematic map are often drawn using few orientations (commonly rectilinear or octilinear) and with as few links as possible. Some schematic maps (like metro maps) can deviate significantly from the geographic context, retaining only the relative positions and topology of the network. Other schematic maps have to obey certain geometric restrictions imposed by the geographic context. For example, if a schematic road network is used as a thematic overlay for a geographic map, the cities must have the same positions in both maps. Fig. 1 illustrates what we want to achieve: given a set of embedded vertices (cities), edges (roads), and obstacles (landmarks), redraw the edges with fixed orientations and few links while retaining the vertex positions and the topology of the network to preserve recognizability.

Our input consists of n_p non-crossing polygonal paths and a set of polygonal obstacles with a total of n_v vertices. We are also given a set $\mathcal{C}$ of orientations. A path or obstacle is $\mathcal{C}$-oriented if every line segment, or *link*, is parallel to some orientation $c \in \mathcal{C}$. Our goal is to compute non-crossing $\mathcal{C}$-oriented paths that are homotopic to the input paths using as few links as possible. Two paths π and π' with the same endpoints are homotopic (notation $\pi \sim_h \pi'$) if π can be "continuously deformed" into π' without passing over any of the obstacles (see e.g. [6] for a formal definition).

* K. Verbeek is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

W. Didimo and M. Patrignani (Eds.): GD 2012, LNCS 7704, pp. 272–278, 2013.

We assume that the obstacles are $\mathcal{C}$-oriented and that the endpoints of the paths are considered as obstacles as well when considering homotopy. We refer to this problem as the *$\mathcal{C}$-oriented routing problem.*

Related Work. Cabello *et al.* [2] give an algorithm that schematizes a network using 2 or 3 links per path, if possible. Nöllenburg and Wolff [13] use a method based on mixed-integer programming to generate metro maps using one edge per path. Both methods do not include obstacles and are restricted to a small number of links per path. Neyer [12] and Merrick and Gudmundsson [11] give algorithms for simplifying paths with fixed orientations. The simplified path can deviate at most a distance ϵ from a given input path. Both methods consider only one path at a time and cannot give guarantees for multiple paths.

There are several papers [1,6] 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 paths. Our problem is also related to *wire routing* in VLSI design [4,7,9,10]. None of these papers strives to minimize the number of 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, where all endpoints lie on the boundary of the polygon. Their paths are not $\mathcal{C}$-oriented. In a previous paper [14] we presented a 2-approximation algorithm for the rectilinear version of our problem.

Results. In this paper we generalize our previous result [14] to $\mathcal{C}$-oriented paths. To this end we introduce a special type of $\mathcal{C}$-oriented paths—*smooth paths*—which are a generalization of rectilinear paths that retain several important properties of rectilinear paths. For the problem that asks for smooth paths—the *smooth routing problem*—we present a 2-approximation algorithm which runs in $O(n^2(n + \log \kappa) + k_{in} \log n)$ time, where $n = n_p + n_v$, k_{in} is the total number of links in the input, and $\kappa = |\mathcal{C}| \geq 2$. For the general $\mathcal{C}$-oriented routing problem, this algorithm computes an $O(\kappa)$-approximation.

We also study a related problem that occurs as a subproblem of our approach. Given a set of (possibly crossing) $\mathcal{C}$-oriented paths with a total of L links, how many links are necessary to untangle the paths, that is, how many links must a set of non-crossing $\mathcal{C}$-oriented paths have that are homotopic to the input paths? For smooth paths, $2L$ links are always enough and this bound is tight. For $\mathcal{C}$-oriented paths we show that $O(L\kappa)$ links are always enough and $\Omega(L \log \kappa)$ links can be necessary. Omitted proofs can be found in the full version of the paper.

2 Preliminaries

We represent a $\mathcal{C}$-oriented path by a sequence of links $\langle \ell_1, \ldots, \ell_k \rangle$ that are connected together at *bends*. We assume that $\mathcal{C}$ is given as a set of vectors $\{c_1, \ldots, c_\kappa\}$ ordered clockwise, where c_1 points in the positive y-direction and the other vectors lie in the right half-plane. We define $c(\ell)$ as the orientation of a link ℓ. So a path π is $\mathcal{C}$-oriented if and only if $c(\ell) \in \mathcal{C}$ for every $\ell \in \pi$.

We define a $\mathcal{C}$-oriented path to be *smooth* if at every bend we do not "skip an orientation". To define this formally, we split up every orientation into two directions to obtain a set $\vec{\mathcal{C}} = \{\vec{c}_1, \ldots, \vec{c}_{2\kappa}\}$, with $\vec{c}_i = c_i$ and $\vec{c}_{i+\kappa} = -c_i$ for $1 \leq i \leq \kappa$. If we direct a $\mathcal{C}$-oriented path from one endpoint to the other, every link ℓ of a $\mathcal{C}$-oriented path has a direction $\vec{c}(\ell) \in \vec{\mathcal{C}}$. We can distinguish two types of bends: clockwise (CW) bends make a right turn, and counterclockwise (CCW) bends make a left turn. A bend between two links ℓ_i and ℓ_{i+1} is smooth if $\vec{c}(\ell_i)$ and $\vec{c}(\ell_{i+1})$ are adjacent directions. A smooth $\mathcal{C}$-oriented path, or *smooth path* in short, has only smooth bends. Note that we do not give lower bounds on the length of a link; for every $\mathcal{C}$-oriented path π, there is a geometrically equivalent smooth path π', but π' can have more links than π.

Three consecutive links form a CW (CCW) *U-turn* if both bends between the links are CW (CCW). A CW (CCW) U-turn is *tight* if an obstacle is touching the right (left) side of the middle link. The overlapping part between the middle link and the obstacle is called the *support* of a U-turn (see Fig. 2). The subpaths of a smooth path between the supports of two consecutive U-turns are called *staircase chains*. Because the path is smooth, staircase chains can use only two (adjacent) directions or orientations. Two staircase chains have the same type if they use the same pair of orientations. Finally, a *shortcut* is defined by two points p and q on a smooth path π, not on the same link, with the following properties: (i) the orientation of $\overline{pq}$ is in $\mathcal{C}$, and (ii) the region enclosed by $\overline{pq}$ and the part of π between p and q, the *shortcut region*, does not contain an obstacle. We assume that $\overline{pq}$ does not cross π, for otherwise we can split $\overline{pq}$ up into multiple shortcuts. We can apply a shortcut to π, which means that we replace the subpath between p and q by $\overline{pq}$. It is easy to see that the resulting path is shorter, $\mathcal{C}$-oriented, and homotopic to π.

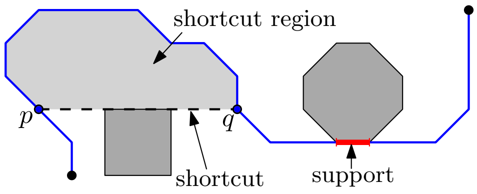

Fig. 2. Supports and shortcuts

Lemma 1. *A path π is shortest w.r.t. its homotopy iff π has no shortcuts. All shortest paths $\pi' \sim_h \pi$ have only tight U-turns and the same sequence of supports.*

Rectilinear paths allow *smallest paths*, paths that minimize both the length and the number of links simultaneously. The same property holds for smooth paths.

Lemma 2. *For every path π, there is a smooth path $\pi' \sim_h \pi$ that minimizes both the length and the number of links for all smooth paths homotopic to π.*

Proof (sketch). Let π' be the path constructed as follows. We start with a minimum-link smooth path homotopic to π. Then we keep applying shortcuts to this path until it has no more shortcuts. By Lemma 1, π' must be shortest. We need to show that applying a shortcut does not increase the number of links. Consider a shortcut between p and q on links ℓ_i and ℓ_j ($i < j$), respectively. Assume w.l.o.g. that there are at least as many CW bends as CCW bends between ℓ_i and ℓ_j. Then $\overline{pq}$ must lie to the right of ℓ_i (sketch, see Fig. 2). Because the path is smooth, all directions clockwise from $\vec{c}(\ell_i)$ to $\vec{c}(\ell_j)$ must be used. After applying the shortcut, we use each of these directions exactly once. Hence the number of links cannot increase. In the special case that q (or p) is an endpoint of π', we can replace $\vec{c}(\ell_j)$ by $\overrightarrow{pq}$ and the same argument works. □

It is NP-hard to decide if there exist non-crossing smallest paths (a proof can be found in the full version of the paper). Still, smallest paths are a good starting point for our algorithm.

3 Algorithm

Our algorithm consists of two steps. First we compute the smallest paths, and then we untangle the paths.

Computing Smallest Paths. We first compute *lowest* and *highest* paths. A staircase chain ρ between a point p and a point q is *lowest* (*highest*) if ρ is the lower (upper) envelope of all staircase chains homotopic to ρ. A smooth path π is *lowest* (*highest*) if it is shortest and all its staircase chains are lowest (highest). Note that lowest and highest paths are well-defined and unique.

Lemma 3. *Lowest (Highest) paths are non-crossing.*

To compute lowest and highest paths we use the algorithm of [6]. Their algorithm first finds the endpoints of x-monotone chains of Euclidean shortest paths. The endpoints of x-monotone chains of shortest smooth paths are the same: they are exactly at the supports of tight U-turns for which the middle link has orientation c_1. Next we compute lowest and highest paths for each x-monotone chain. By Lemma 1, lowest and highest paths must have the same sequence of supports. To obtain smallest paths, we compute minimum-link staircase chains between every two consecutive supports, which must be bounded by the lowest and highest paths. This can be computed in $O(n \log \kappa)$ time for a single x-monotone chain. Since there are only $O(n)$ homotopically different x-monotone chains, we can compute all smallest paths in $O(n^2 \log \kappa + k_{in} \log n)$ time. In the following we assume that paths are x-monotone and directed from left to right.

Lemma 4. *Given a set of shortest smooth paths, crossings can occur only between staircase chains of the same type.*

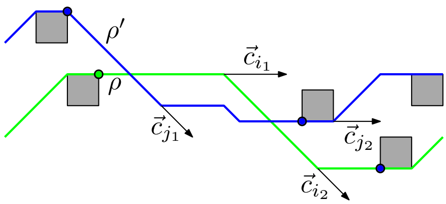

Fig. 3. Only chains of the same type cross

Proof. Let ρ and ρ' be two staircase chains that cross. If they cross only once, then the only way to remove the crossing is to move ρ over an endpoint of ρ' or vice versa. Because the paths are shortest, there must be obstacles at the endpoints of ρ and ρ', so chains homotopic to ρ and ρ' always cross. But, by Lemma 3, lowest paths are non-crossing, so ρ and ρ' must cross at least twice. Let the links of ρ and ρ' involved in the first crossing have directions $\vec{c}_{i_1}$ and $\vec{c}_{j_1}$ ($i_1 < j_1$), respectively. Similarly define i_2 and j_2 for the second crossing, for which $j_2 < i_2$ (see Fig. 3). Since ρ and ρ' can use only two adjacent directions, we get $|i_2 - i_1| \leq 1$ and $|j_2 - j_1| \leq 1$. But then $i_1 = j_2$ and $i_2 = j_1$. □

Untangling. We reduce the problem to rectilinear paths as follows. Consider a single type of staircase chain with orientations c_i and c_{i+1}. We rotate the plane such that c_{i+1} aligns with the x-axis, and then we apply a shearing transformation to align c_i with the y-axis. Because this is an affine transformation, straight lines (links) and crossings are preserved by the transformation (see Fig. 4). Now we can apply the incremental untangling algorithm of [14], and untangle the staircase chains (of this type) using at most

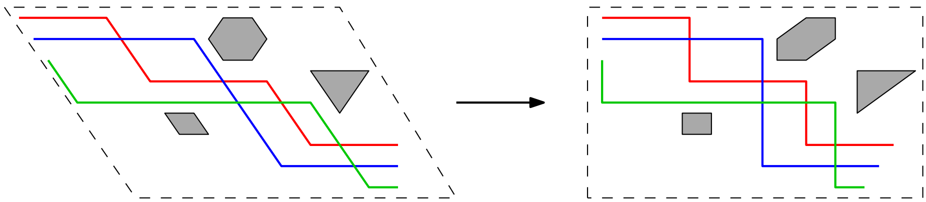

Fig. 4. Shearing transformation to rectilinear paths

twice the number of links. We do this for all types of staircase chains. By Lemma 4, all crossings will be removed.

We do not need to explicitly compute the transformation. We can apply the algorithm to complete x-monotone chains directly. If we add the paths from top to bottom, the algorithm adds two links per CW bend. In the opposite order, it adds two links per CCW bend. The result with fewest links is a 2-approximation.

Theorem 1. *We can compute a* 2*-approximation for the smooth routing problem in* $O(n^2(n + \log \kappa) + k_{in} \log n)$ *time, which is also a* $(2\kappa - 2)$*-approximation for the* $\mathcal{C}$*-oriented routing problem.*

Proof. The first part follows from the discussion above and the result in [14]. Consider a minimum-link $\mathcal{C}$-oriented path π with L links. We can make π smooth by adding at most $\kappa - 2$ links per bend, so a minimum-link smooth path homotopic to π has at most $(\kappa - 1)L$ links. This implies the second part. □

In the full version of the paper we describe how to extend our algorithm to thick paths.

4 Untangling Bounds

We have shown that a set of $\mathcal{C}$-oriented paths with a total of L links can be untangled using at most $O(L\kappa)$ links. But how many links are necessary? For smooth paths with L links, a similar construction as for rectilinear paths [14] shows that $2L$ links can be necessary. For general $\mathcal{C}$-oriented paths, we give a lower bound that is based on a construction from [8].

Theorem 2. *There is a set of orientations* $\mathcal{C}$ *and a set of* $\mathcal{C}$*-oriented paths with a total of L links such that we need* $\Omega(L \log \kappa)$ *links to untangle the paths.*

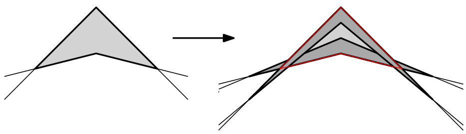

Fig. 5. Lower bound construction

Proof. Consider the quadrilateral enclosed by two paths shown in Fig. 5 (left). We can replace this quadrilateral by two crossing quadrilaterals as shown in the figure. We repeat this operation recursively for each quadrilateral. After k rounds, there are 2^k quadrilaterals formed by 2^{k+1} paths with 2 links each. To be able to untangle these paths, each quadrilateral that occurred during the construction must be free of obstacles. We assume the rest of the plane is filled with obstacles. As is shown in [8], we need at least $\Omega(k2^k)$ links to untangle the paths. Let $\mathcal{C}$ be the set of all orientations formed by the 2^{k+2} links. Then this construction consists of κ links and requires $\Omega(\kappa \log \kappa)$ links to untangle the paths. By copying this construction $\lfloor L/\kappa \rfloor$ times, we obtain the claimed bound, for L large enough. □

The above construction requires a specific set $\mathcal{C}$ to work. Usually, the set $\mathcal{C}$ is uniform, that is, the angle between two adjacent orientations is constant. In the full version of the paper we show that the above bound also holds if $\mathcal{C}$ is uniform.

5 Conclusion

We presented an $O(\kappa)$-approximation algorithm for the $\mathcal{C}$-oriented routing problem. To that end, we introduced smooth paths, and gave a 2-approximation algorithm for the smooth routing problem. Although we use smooth paths only as a tool for our algorithm, smooth paths might actually be better candidates for our application. Unlike smooth paths, general $\mathcal{C}$-oriented paths can have sharp bends which are hard to follow. So perhaps, for schematic maps, smooth paths are the better choice. We also studied how many links are necessary to untangle a set of $\mathcal{C}$-oriented paths with a total of L links. We know that $O(L\kappa)$ links are always enough and that $\Omega(L \log \kappa)$ links can be necessary. Finding a tight bound is an interesting open problem.

References

1. Bespamyatnikh, S.: Computing homotopic shortest paths in the plane. J. Alg. 49(2), 284–303 (2003)
2. Cabello, S., de Berg, M., van Kreveld, M.: Schematization of networks. Comp. Geom.: Theory and Appl. 30(3), 223–238 (2005)
3. Cabello, S., Liu, Y., Mantler, A., Snoeyink, J.: Testing homotopy for paths in the plane. Discrete & Computational Geometry 31, 61–81 (2004)
4. Cole, R., Siegel, A.: River routing every which way, but loose. In: Proc. 25th Symp. Found. Comp. Sci., pp. 65–73 (1984)
5. Duncan, C.A., Efrat, A., Kobourov, S.G., Wenk, C.: Drawing with fat edges. Intern. J. Found. Comp. Sci. 17(5), 1143–1163 (2006)
6. Efrat, A., Kobourov, S.G., Lubiw, A.: Computing homotopic shortest paths efficiently. Comp. Geom.: Theory and Appl. 35(3), 162–172 (2006)
7. Gao, S., Jerrum, M., Kaufmann, M., Mehlhorn, K., Rülling, W.: On continuous homotopic one layer routing. In: Proc. 4th Symp. Comp. Geom., pp. 392–402 (1988)
8. Gupta, H., Wenger, R.: Constructing pairwise disjoint paths with few links. ACM Trans. Alg. 3(3), 26 (2007)
9. Leiserson, C.E., Maley, F.M.: Algorithms for routing and testing routability of planar VLSI layouts. In: Proc. 17th Symp. Theory. Comp., pp. 69–78 (1985)
10. Maley, F.M.: Single-layer wire routing and compaction. MIT Press, Cambridge (1990)

11. Merrick, D., Gudmundsson, J.: Path Simplification for Metro Map Layout. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 258–269. Springer, Heidelberg (2007)
12. Neyer, G.: Line Simplification with Restricted Orientations. In: Dehne, F., Gupta, A., Sack, J.-R., Tamassia, R. (eds.) WADS 1999. LNCS, vol. 1663, pp. 13–24. Springer, Heidelberg (1999)
13. Nöllenburg, M., Wolff, A.: Drawing and labeling high-quality metro maps by mixed-integer programming. IEEE Transactions on Visualization and Computer Graphics 17(5), 626–641 (2011)
14. Speckmann, B., Verbeek, K.: Homotopic Rectilinear Routing with Few Links and Thick Edges. In: López-Ortiz, A. (ed.) LATIN 2010. LNCS, vol. 6034, pp. 468–479. Springer, Heidelberg (2010)