THICK NON-CROSSING PATHS

JOSEPH S. B. MITCHELL AND VALENTIN POLISHCHUK

DEPARTMENT OF APPLIED MATHEMATICS AND STATISTICS, STONY BROOK UNIVERSITY

ABSTRACT. We consider the problem of finding shortest non-crossing thick paths in a polygonal domain, where a *thick* path is a Minkowski sum of a usual path and unit disk. We show that in a *simple* polygon, K shortest paths may be found in optimal $O(K(n+K))$ time and $O(n+K)$ space.

For polygons with holes we show that the problem becomes (weakly) NP-hard even for the case $K = 2$ and even when the paths are restricted to be monotone, if we wish to bound the length of the *longest* of the paths. We also observe that unless P=NP there exists no fully polynomial time approximation scheme for the problem. For the case $K = 2$ and L_1 metric we suggest a pseudo-polynomial time algorithm.

Next, we consider a special case of the *minimum cost rectilinear flow* problem. We observe that under several restrictions, the paths of minimum *total* length may be found in polynomial time by reducing the shortest paths problem to the flow problem in a path-preserving graph.

1. PRELIMINARIES

Let C be the open unit disk centered at the origin. For a set $S \subset \mathbb{R}^2$ and $r \in \mathbb{R}^+$, we let $(S)^r$ denote the Minkowski sum $S \oplus rC$. When considering rectilinear domains C is the unit square $[0,1]^2$.

Let P be a simple polygon; let $P^1 = P \setminus (bdP)^1$ be P offset by 1 inside. Let (s_k,t_k), $k = 1 \dots K$ be pairs of points on the boundary of P^1. For $u, v \in bdP^1$ let $P^1(u,v)$ be the portion of bdP^1 from u to v clockwise. WLOG, we assume that starting from s_1 and going around bdP^1 clockwise one encounters $s_1, s_2 \dots, s_K$ in this order and that for any i, s_i appears before t_i.

We consider the problem of finding s_k-t_k paths π_k each of which is as short as possible. The constraint is $(\pi_i)^1 \cap (\pi_j)^1 = \emptyset$, $i \neq j$.

The geometric thick disjoint paths problem (considered here) and graph theoretic disjoint paths problem are related, in that thick disjoint paths in polygonal domains correspond to disjoint paths in certain planar graphs. Throughout this work we exploit this connection in both directions. We use the hardness of the graph problem to establish the hardness of the geometric version, and we translate geometric problem into the problem on a (path-preserving) graph.

2. SIMPLE POLYGONS

Following [5], we map bdP^1 to the unit circle $bd\mathcal{C}$ and draw a chord between every pair $(s_i,t_i), i = 1 \dots K$. Let $\sigma_1, \dots, \sigma_{2K}$ be the set $\{s_1, \dots, s_K,\ t_1, \dots, t_K\}$ ordered clockwise around bdP^1; let $bd\mathcal{C}(\sigma_i, \sigma_{i+1})$ be the part of $bd\mathcal{C}$ between σ_i and σ_{i+1} (not including σ_i and σ_{i+1}).

Definition 2.1. *Let γ be a path within $\mathcal{C}$ from a point on $bd\mathcal{C}(\sigma_j,\sigma_{j+1})$ to a point on the chord s_it_i. The i^{th}* **depth** *of $P^1(\sigma_j,\sigma_{j+1})$, $d_i(\sigma_j,\sigma_{j+1})$, is the minimum over all γ of the number of chords that γ crosses.*

Let $\mathcal{O}_i$ be the set of obstacles, obtained when offsetting inside each part of bdP^1 by 2 times its i^{th} depth:

$$\mathcal{O}_i = \bigcup_{j=1}^{2K-1} \left(bdP^1(\sigma_j,\sigma_{j+1})\right)^{2d_i(\sigma_j,\sigma_{j+1})} \bigcup \left(bdP^1(\sigma_{2K},\sigma_1)\right)^{2d_i(\sigma_{2K},\sigma_1)}$$

If $\pi_i^*, i = 1 \dots K$ is the path routed amidst $\mathcal{O}_i$ as obstacles, than each path $(\pi_1^*)^1, \dots, (\pi_K^*)^1$ is as short as possible given the existence of the others. Moreover, the paths are non-crossing by a local optimality argument.

Theorem 2.2. *The shortest thick non-crossing paths problem can be solved in $O(K(n+K))$ time and $O(n+K)$ space.*

Proof. Finding the depths of the intervals of bdP^1 can be done in $O(n+K^2)$ time.

Let $n_j, j = 1 \dots 2K$ be the complexity of $P^1(\sigma_j,\sigma_{j+1})$. Each of the Minkowski sums $\left(bdP^1(\sigma_j,\sigma_{j+1})\right)^{2d_i(\sigma_j,\sigma_{j+1})}$, $j = 1 \dots 2K$ is the union of the Minkowski sums of the edges of P with disks of different radii; these Minkowski sums form a collection of pseudodiscs. Thus, the complexity of $\mathcal{O}_i$ is $O(\sum n_j) = O(n+K)$ ([1]) and $\mathcal{O}_i$ can be found in $O(\sum n_j) = O(n+K)$ time ([2]). Then, routing of π_i amidst $\mathcal{O}_i$ can be done in $O(n+K)$ time ([4]). □

We remark that the $O(n+K)$ space requirement of our algorithm is the *working space* requirement. In general, the complexity of i^{th} path may be as high as $\Omega(n+i)$, in which case the size of the output may reach $\Omega(Kn+K^2)$, and $\Omega(K(n+K))$ *output space* may be needed just to store the paths. We can use the approach of [5] to store a forest of the paths, of size $O(n+K)$, such that any path can be output in time proportional to its complexity. We expect that we can use the ideas from [5] to actually construct the forest in $O(n+K)$ time.

Date: November 2, 2005.

This research is partially supported by grants form Metron Aviation, NASA Ames (NAG2-1620) and NSF (CCR-0098172, ACI-0328930, CCF-0431030, CCF-0528209).

Corresponding author: valentin.polishchuk@stonybrook.edu.

3. Polygons with Holes

Holst and Pina ([7]) proved by reduction from Partition that finding two (internally) vertex-disjoint s-t paths in a planar graph is weakly NP-complete. Imagine that each edge of a planar graph G is a "tube" of width 1 so that two thick paths do not fit into one edge or vertex. Then vertex-disjoint paths in the graph correspond to non-crossing thick paths in polygonal domain. Since G can be drawn with rectilinear edges,

Theorem 3.1. *The problem of finding two thick non-crossing paths with bounded length in a rectilinear domain is weakly NP-complete.*

A Partition instance is specified with integers. Thus, if an instance is infeasible, the longest of the two paths in G is at least 1 longer than it would have been in a feasible instance. Knowing a second best possible solution proves

Corollary 3.2. *Unless P=NP, there exists no FPTAS for the problem.*

4. Pseudo-Polynomial Algorithm

If s_1, t_1, s_2, t_2 belong to the same obstacle, the problem can be solved in pseudo-polynomial time by a reduction to a recently solved problem in graph theory: finding two vertex-disjoint length bounded paths in a planar graph. It was shown ([7]) that the latter problem can be solved in $O(m^4L^2)$ time and $O(m^2L^2)$ space, for a graph with m nodes and maximum edge length L.

Lay down an N-by-N grid so that the vertices of the obstacles snap onto the grid, and delete from the grid the nodes i for which $(i)^1$ intersects an obstacle. Let $\mathbb{G}$ be the grid graph induced by the remaining grid nodes.

Lemma 4.1. $\mathbb{G}$ *is rich enough to search for the optimal paths.*

Proof. The proof is by a standard "snapping" argument, extensively used in rectilinear computational geometry to show path-preserving properties of finite graphs built from rectilinear domains ([3]). Care must be taken, though, to ensure that both paths are snapped onto the grid without crossing. □

The number of nodes in $\mathbb{G}$ is $O(N^2)$ and each edge length is 1. Thus, the solution to the graph problem provides a pseudo-polynomial time algorithm for the geometric problem.

5. Minimizing Total Length

We investigate the *minimum cost integer rectilinear flow*: given rectilinear obstacles and sources s_1, s_2 and sinks t_1, t_2, find two thick non-crossing paths of minimum total length, connecting s_1 to one of $\{t_1, t_2\}$, and s_2 to the other of $\{t_1, t_2\}$. We solve the problem by reducing it to the flow problem in a graph $\mathbb{G}_0$ – a sparse version of the graph $\mathbb{G}$ defined in the previous section. $\mathbb{G}_0$ is build as follows. Extend North and East edges of obstacles. Extend North (resp., East) edges shifted *up* (resp., *right*) by 1. Shift South (West) edges down (left) by 1 and 2 and extend. $\mathbb{G}_0$ is obtained by intersecting the extensions and deleting edges b such that $(b)^1$ intersects an obstacle; a super-source s and sink t are added. The capacity of nodes and edges in $\mathbb{G}_0$ are 1s.

Lemma 5.1. *It is enough to find a min-cost max-s-t flow in* $\mathbb{G}_0$.

Proof. By the flow decomposition and snapping argument. The source-sink pairing may change after a snapping, but in the problem we consider it is allowed. □

Theorem 5.2. *The minimum cost integer rectilinear flow problem can be solved in $O(n^2 polylog\, n)$ time and $O(n^2)$ space.*

Proof. $\mathbb{G}_0$ can be constructed in $O(n^2 \log n)$ time and $O(n^2)$ space. Anyway, the time of the construction is dominated by finding the minimum cost flow in $\mathbb{G}_0$, $O(n^2\, polylog\, n)$ ([6]). □

Corollary 5.3. *The minimum cost integer rectilinear flow between K sources and K sinks may be found in $O(K^2n^2 polylog(Kn))$ time and $O(K^2n^2)$ space.*

Under certain restrictions on the sources-sinks placement, the problem of minimizing the total length of the thick paths is the flow problem.

Theorem 5.4. *Under certain restrictions, minimizing the total length of K thick non-crossing paths can be done in $O(K^2n^2 polylog(Kn))$ time and $O(K^2n^2)$ space.*

The K paths of minimum total length provide a K-approximation for the problem of finding K paths of bounded length. Thus,

Corollary 5.5. *Under certain restrictions, a K-approximation to the problem of finding K disjoint thick bounded length paths can be found in $O(K^2n^2 polylog(Kn))$ time and $O(K^2n^2)$ space.*

References

[1] P. K. Agarwal and M. Sharir. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[2] F. Chin, J. Snoeyink, and C. A. Wang. Finding the medial axis of a simple polygon in linear time. *Discrete Comput. Geom.*, 21(3):405–420, 1999.

[3] D. T. Lee, C. D. Yang, and C. K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Appl. Math.*, 70:185–215, 1996.

[4] E. A. Melissaratos and D. L. Souvaine. On solving geometric optimization problems using shortest paths. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1990.

[5] E. Papadopoulou. k-pairs non-crossing shortest paths in a simple polygon. *Int. J. Com. Geom. Appl.*, 9(6):533–552, 1999.

[6] E. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.

[7] H. van der Holst and J. C. de Pina. Length-bounded disjoint paths in planar graphs. *Discrete Appl. Math.*, 120(1-3):251–261, 2002.