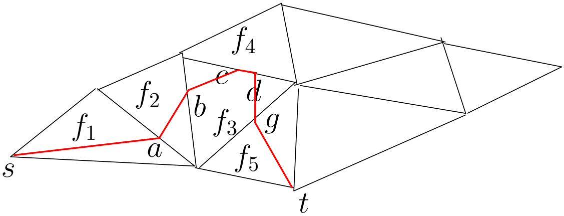
Computing an Approximate Minimum Cost Path among Weighted Regions in the Plane

R. Inkulu
http://www.iitg.ac.in/rinkulu/

(Joint work with Sanjiv Kapoor)

Problem description



cost of the red path is

$$w_{f_1}\|sa\| + w_{f_2}\|ab\| + w_{f_3}\|bc\| + \min(w_{f_3}, w_{f_4})\|cd\| + w_{f_3}\|dg\| + w_{f_5}\|gt\|$$

- Given a triangulation $\mathcal{P}$ with $O(n)$ faces, each face associated with a positive weight, find a path between two input points s and t (both belonging to $\mathcal{P}$) so that the path has minimum cost among all possible paths joining s and t that lie on $\mathcal{P}$.

- The cost of any path p is the sum of costs of all line segments in p, whereas the cost of a line segment is its Euclidean length multiplied by the weight of the face on which it lies.

Hardness of the problem

- Computing an optimal path is believed to be hard; and it is not of interest to practitioners in particular. [1]

[1] In the algebraic computation model over the rational numbers, computing an optimal path amid weighted regions in $\mathbb{R}^2$ is proven to be unsolvable (refer to De Carufel et al. CGTA 2014). In $\mathbb{R}^3$, even when every face weight belong to $0, \infty$, computing an optimal path amid weighted regions is proven to be NP-hard using a reduction from 3-SAT (refer to Canny and Reif, FOCS '87).

Hardness of the problem

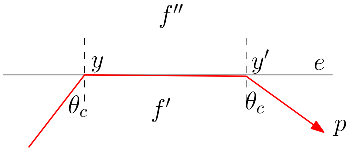
- Computing an optimal path is believed to be hard; and it is not of interest to practitioners in particular. [1]

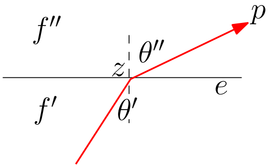
  Hence, apprx algorithms are of interest; we devise a FPTAS.

[1] In the algebraic computation model over the rational numbers, computing an optimal path amid weighted regions in $\mathbb{R}^2$ is proven to be unsolvable (refer to De Carufel et al. CGTA 2014). In $\mathbb{R}^3$, even when every face weight belong to $0, \infty$, computing an optimal path amid weighted regions is proven to be NP-hard using a reduction from 3-SAT (refer to Canny and Reif, FOCS '87).

Outline

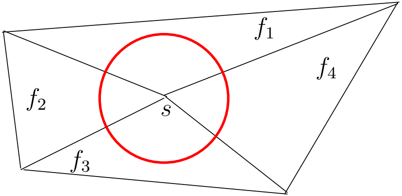
[Mitchell, Papadimitriou JACM '91]: characterized shortest paths in terms of Snell's laws



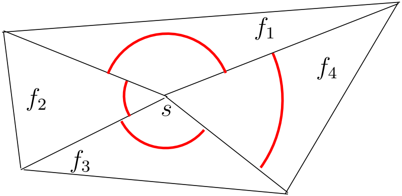


- If a geodesic path p shares a segment yy' with edge e, for y not being a vertex, then both the angle of incidence at y from f' and angle of exit into f' at y' are *critical*: $\theta_c = \sin^{-1}(\frac{w_{f''}}{w_{f'}})$. (This cases arises only when $w_{f''} < w_{f'}$.)
- If a geodesic path p crosses edge e at a point z, then p obeys *Snell's law of refraction* at z: $w_{f'} \sin_{\theta'} = w_{f''} \sin_{\theta''}$ for $\theta' < \theta_c$.
- There does not exist a least cost path whose angle of incidence is greater than θ_c.

[Mitchell, Papadimitriou JACM '91]: algorithm based on continuous Dijkstra

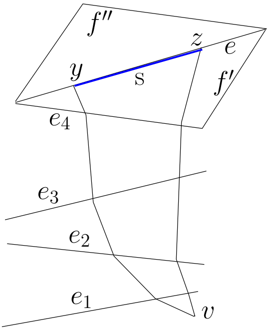


weights of all the faces are same



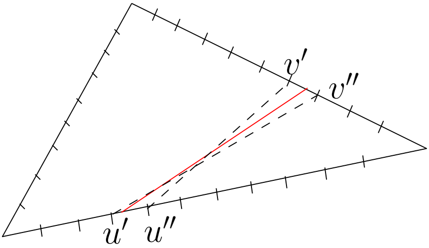
weights of faces are not same

[Mitchell, Papadimitriou JACM '91]: progresses intervals of optimality



- Simulates the wavefront progress by the progression of *intervals of optimality* over the faces of $\mathcal{P}$
  - each such interval s denotes one maximal subsection of an edge (wrt a face f') for which the shortest path to any point on s has the same discrete structure

[Lanthier et al. Algorithmica '01]: reduced to a graph-theoretic problem



- For each face f_i of $\mathcal{P}$ a graph G_i is constructed: $\Theta(n^2)$ Steiner points are evently placed along each edge of f_i; a node pair u and v is connected in G_i whenever u and v belong to distinct edges of f_i or they are neighbors on an edge.

outputs an apprx shortest path with additive error

Variants of [Lanthier et al. Algorithmica '01]

- [Aleksandrov et al. SWAT'98]
  Steiner points are placed in a geometric progression along the edge
- [Aleksandrov et al. STOC'00]
  Based on Snell's laws of refraction, prunes edges through which Dijkstra's wavefront need to progress
- [Sun and Reif JAlgo '06]
  Prunes further by exploiting the non-crossing property of shortest paths
- [Aleksandrov et al. JACM '05]
  Steiner points are placed in a geometric progression along the three bisectors of each face
- [Cheng et al. SIAMJC '10, Cheng et al. SODA '15]
  Prunes $\mathcal{P}$ based on the intersection of an ellipse (whose size relies on the unweighted geodesic distance between s and t) and $\mathcal{P}$ before applying [Aleksandrov et al. JACM '05]; handles convex distance functions

Time complexity comparison chart[2]

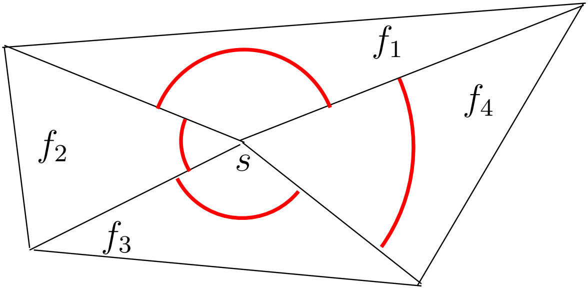
[Mitchell, Papadimitriou JACM '91]	$O(n^8 \lg \frac{nN\mu}{\epsilon})$
[Mata and Mitchell SoCG '97]	$O(\frac{\mu}{\epsilon\theta_{min}} n^3)$
[Sun and Reif JAlgo '06]	$O(\frac{nN^2}{\epsilon} \lg(N\mu) \lg \frac{n}{\epsilon} \lg \frac{1}{\epsilon})$
[Aleksandrov et al. JACM '05]	$O(\frac{nN^2}{\sqrt{\epsilon}} \lg(N\mu) \lg \frac{n}{\epsilon} \lg \frac{1}{\epsilon})$
[Cheng et al. SODA '15]	$O(\frac{kn+k^4 \lg(k/\epsilon)}{\epsilon} \lg^2 \frac{\rho n}{\epsilon})$
Our result	$O(n^5 \lg n + n^4 \lg(\frac{\mu}{\epsilon}(1 + \frac{1}{\sin\theta_{min}})))$

Like [Mitchell, Papadimitriou JACM '91], our algorithm is polynomial in n.

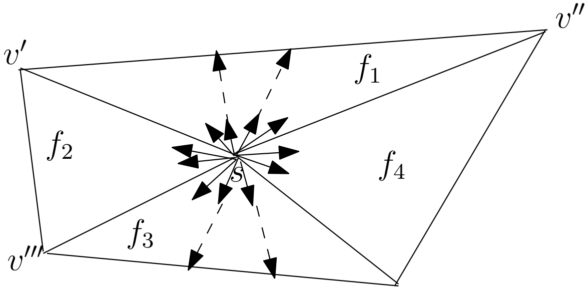
[2] n: number of vertices defining $\mathcal{P}$; L: length of the longest edge bounding any face of $\mathcal{P}$; N: maximum coordinate value used in describing $\mathcal{P}$; w_{max}: maximum non-infinite weight associated with any triangle; w_{min}: minimum weight associated with any triangle; θ_{min}: minimum among the internal face angles of $\mathcal{P}$; and, μ: ratio of w_{max} to w_{min}; k is the smallest integer such that the sum of the k smallest angles in $\mathcal{P}$ is at least π

Outline

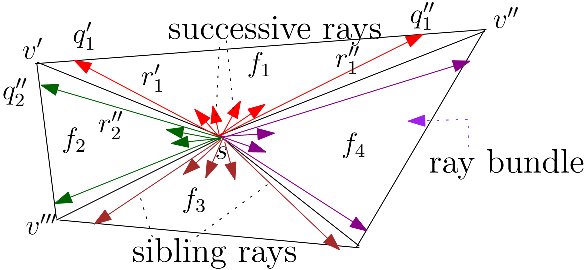
Progressing wavefront: continuous Dijkstra in weighted domains



Progressing wavefront: discretized Dijkstra

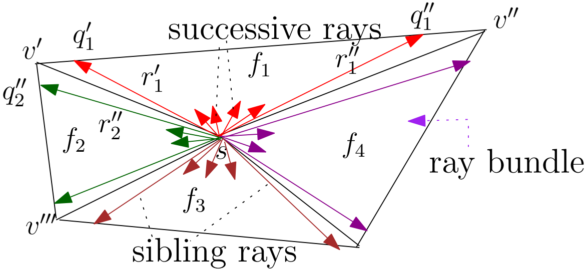


Tracing discrete wavefront



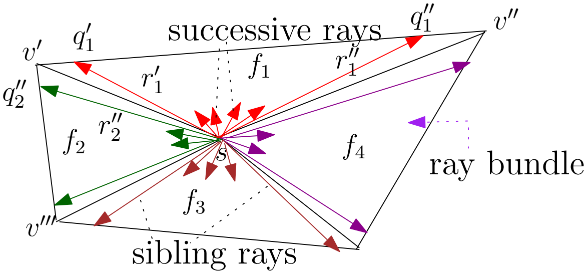
- initiate many rays from s but trace only few

Tracing discrete wavefront



- initiate many rays from *s* but trace only few
- we upper bound the number of rays initiated and the ones that get traced for the worst-case

Events corresponding to tracing of ray bundles



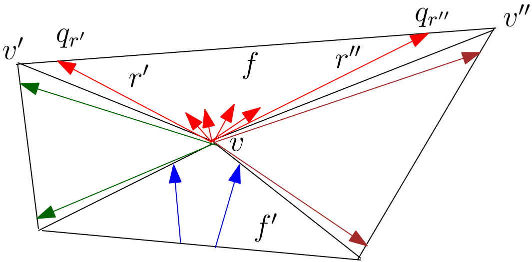
event point pairs are pushed to min-heap:

q_1'-q_1'', etc.,

q_1'-q_2'', etc.,

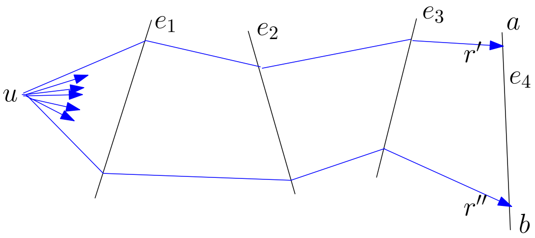
Note that the bundle of rays are pairwise divergent.

Initiating ray bundles from a vertex



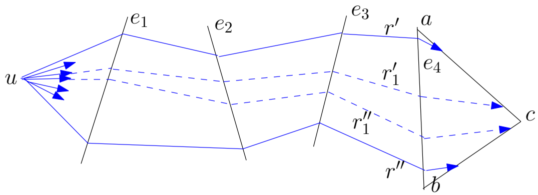
initiate a discrete wavefront from *v* when blue ray bundle strikes *v* while exploiting the non-crossing property of shortest paths

Rays in ray bundle



two rays belong to a ray bundle if they traverse across the same *edge sequence* whenever traced

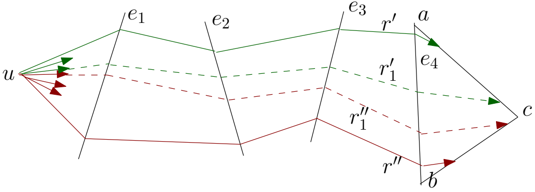
Ray bundle split due to a vertex



successive rays r'_1 and r''_1 are identified with binary search over the rays in blue ray bundle

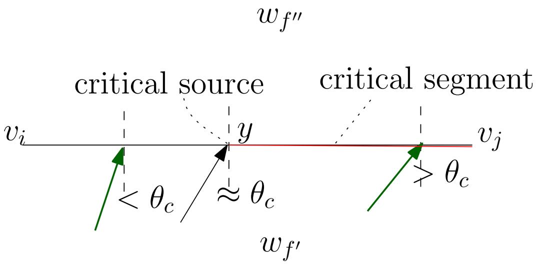
- The rays that belong to the same ray bundle, the edge sequence that they traverse across is same.

Ray bundle split due to a vertex



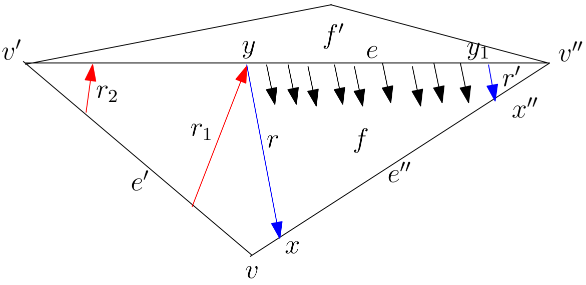
new ray bundles are formed and the corresponding sibling pairs are defined

Detecting critical incidence



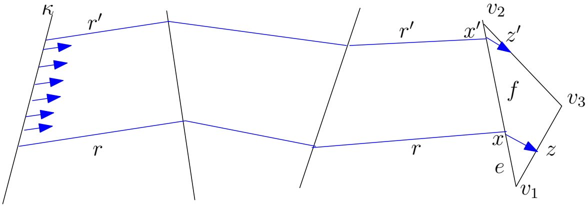
here the critical angle θ_c is $\sin^{-1}(\frac{w_{f''}}{w_{f'}})$ wherein $w_{f''} < w_{f'}$

Initiating rays from a critical segment

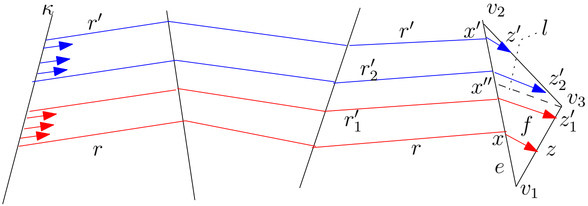


number and position of points from which rays are generated is a function of ϵ

Tracing rays from a critical segment

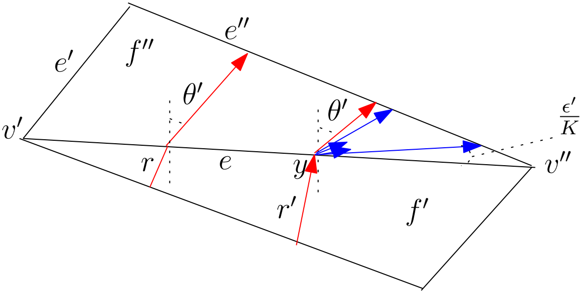


Split of a ray bundle initiated at a critical segment



- linear interpolation in finding x'' suffice instead of tracing rays from κ

Rays initiated from a critical source



- helps in having sparser sets of rays initiated from vertex and critical segment sources
- these rays are traced similar to the way rays initiated from a vertex source

Recap

sources of ray bundles:

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments
- critical sources

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments
- critical sources

event points of interest:

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments
- critical sources

event points of interest:

- initiating rays from sources

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments
- critical sources

event points of interest:

- initiating rays from sources
- tracing ray bundles

Recap

sources of ray bundles:

- vertices of $\mathcal{P}$, including s
- critical segments
- critical sources

event points of interest:

- initiating rays from sources
- tracing ray bundles
- ray bundle splits due to new ray bundle sources

Algorithm

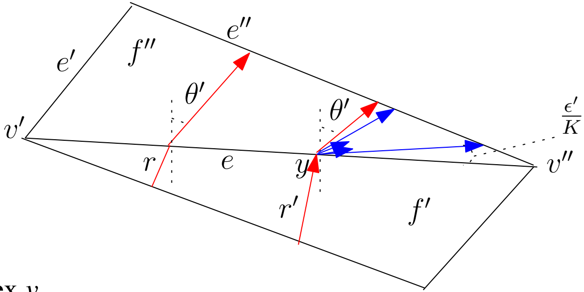
(1) initiate a set of ray bundles from s

(2) while (t is not struck by a ray bundle)

    (i) push new event points to min-heap

    (ii) handle event points popped from min-heap

Few optimizations: tree of rays

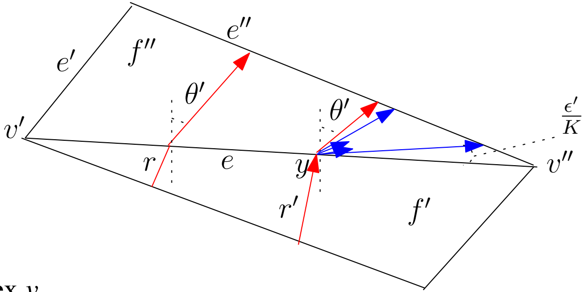


For each vertex v,

- ray bundles from v
- ray bundles from critical sources whose nearest ancestor vertex is v

are organized into a tree, $\mathcal{T}_R(v)$.

Few optimizations: tree of rays



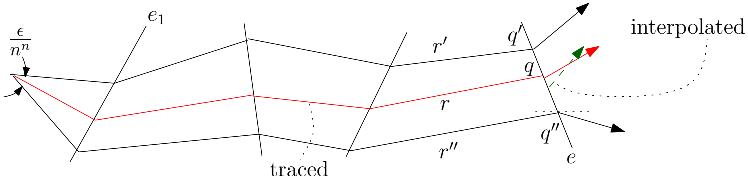
For each vertex v,

- ray bundles from v
- ray bundles from critical sources whose nearest ancestor vertex is v

are organized into a tree, $\mathcal{T}_R(v)$.

Two rays in a ray bundle are *siblings* whenever the edge sequence associated with one is a suffix of the edge sequence of the other;
binary search for ray pairs in $T_R(v)$ is possible due to pairwise divergence of rays in $T_R(v)$.

Few more optimizations: interpolate when the angle is small



- avoid tracing rays across lengthy ($O(n^2)$) edge sequence: instead interpolate when the angle between traced rays is small

Properties exploited in the analysis

- Let p be a geodesic path. Then either (i) between any two consecutive vertices on p, there is at most one critical point of entry to an edge e, and at most one critical point of exit from an edge e' (possibly equal to e); or (ii) the path p can be modified in such a way that case (i) holds without altering the length of the path.

Properties exploited in the analysis

- Let p be a geodesic path. Then either (i) between any two consecutive vertices on p, there is at most one critical point of entry to an edge e, and at most one critical point of exit from an edge e' (possibly equal to e); or (ii) the path p can be modified in such a way that case (i) holds without altering the length of the path.

- The length of any edge sequence of a shortest locally f-free path p to a point on the boundary of f is $O(n^2)$.

Properties exploited in the analysis

- Let p be a geodesic path. Then either (i) between any two consecutive vertices on p, there is at most one critical point of entry to an edge e, and at most one critical point of exit from an edge e' (possibly equal to e); or (ii) the path p can be modified in such a way that case (i) holds without altering the length of the path.

- The length of any edge sequence of a shortest locally f-free path p to a point on the boundary of f is $O(n^2)$.

- Any shortest geodesic path p, passes through $O(n)$ critical points of entry on any given edge e.

Properties exploited in the analysis

- Let p be a geodesic path. Then either (i) between any two consecutive vertices on p, there is at most one critical point of entry to an edge e, and at most one critical point of exit from an edge e' (possibly equal to e); or (ii) the path p can be modified in such a way that case (i) holds without altering the length of the path.

- The length of any edge sequence of a shortest locally f-free path p to a point on the boundary of f is $O(n^2)$.

- Any shortest geodesic path p, passes through $O(n)$ critical points of entry on any given edge e.

- *Non-crossing property of shortest paths*: Any two shortest geodesic paths with the same source point cannot intersect in the interior of any region.

Bounding the ray density at sources to obtain a PTAS

Considering refraction/reflection paths of any two successive rays initiated from any type of source, initiating $O(\frac{2\mu}{\epsilon'}(\frac{1}{\epsilon'})^{n^2})$ suffice to achieve ϵ-apprximation, where $\epsilon' = \frac{\epsilon}{n^3\mu(1+\frac{1}{\sin\theta_{min}})}$.

Time complexity

Time complexity

- ray bundle splits at vertices

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources
- number of ray bundles from critical sources

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources
- number of ray bundles from critical sources
- splits of ray bundles from critical segments

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources
- number of ray bundles from critical sources
- splits of ray bundles from critical segments
- tracing ray bundles across edge sequences

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources
- number of ray bundles from critical sources
- splits of ray bundles from critical segments
- tracing ray bundles across edge sequences

  savings due to tree of rays and interpolations

Time complexity

- ray bundle splits at vertices
- ray bundle splits at critical sources
- number of ray bundles from vertex sources
- number of ray bundles from critical sources
- splits of ray bundles from critical segments
- tracing ray bundles across edge sequences

  savings due to tree of rays and interpolations

Takes $O(n^5 \lg n + n^4 \lg(\frac{\mu}{\epsilon}(1 + \frac{1}{\sin_{\theta_{min}}})))$ time to find an ϵ-approximate shortest path from s to t.

Major ideas

Major ideas

- discrete wavefront as sets of rays

Major ideas

- discrete wavefront as sets of rays
- partitioning the wavefront into ray bundles

Major ideas

- discrete wavefront as sets of rays
- partitioning the wavefront into ray bundles
- lazy tracing of rays in ray bundles

Major ideas

- discrete wavefront as sets of rays
- partitioning the wavefront into ray bundles
- lazy tracing of rays in ray bundles
- binary search within tree of rays

Major ideas

- discrete wavefront as sets of rays
- partitioning the wavefront into ray bundles
- lazy tracing of rays in ray bundles
- binary search within tree of rays
- interpolating instead of tracing wherever it is possible

Single-source apprx shortest path queries

- Achieves constructing a least cost path in $O(n^4 \lg \frac{n}{\epsilon})$ query time with $O(n^5 (\lg \frac{n}{\epsilon})(\lg \frac{\mu}{\sqrt{\epsilon}})(\lg N))$ preprocessing time.

  while the best polynomial query time stands at $O(n^7 polylog)$ ([Mitchell, Papadimitriou JACM '91])

Outline

Take-homes

- A generalization of well-known Euclidean shortest path problem

Take-homes

- A generalization of well-known Euclidean shortest path problem
- Continuous vs discrete-Dijkstra wavefront

Take-homes

- A generalization of well-known Euclidean shortest path problem
- Continuous vs discrete-Dijkstra wavefront
- Reducing the geometric problem to a graph-theoretic one *vs* solving the problem in the geometric domain itself

Open problems

- since the known worst-case lower bound on the number of event points in the continuous Dijkstra amid weighted regions is known to be $\Omega(n^4)$ (from [Mitchell, Papadimitriou JACM '91]) , the next objective could be to design an algorithm with $O(n^4 \text{ polylog})$ time.

Open problems

- since the known worst-case lower bound on the number of event points in the continuous Dijkstra amid weighted regions is known to be $\Omega(n^4)$ (from [Mitchell, Papadimitriou JACM '91]) , the next objective could be to design an algorithm with $O(n^4 \text{ polylog})$ time.

- more efficient single-source queries and two-point queries

Open problems

- since the known worst-case lower bound on the number of event points in the continuous Dijkstra amid weighted regions is known to be $\Omega(n^4)$ (from [Mitchell, Papadimitriou JACM '91]) , the next objective could be to design an algorithm with $O(n^4 \text{ polylog})$ time.

- more efficient single-source queries and two-point queries

- extending to polyhedral weighted surfaces and to $\mathbb{R}^3$

Open problems

- since the known worst-case lower bound on the number of event points in the continuous Dijkstra amid weighted regions is known to be $\Omega(n^4)$ (from [Mitchell, Papadimitriou JACM '91]) , the next objective could be to design an algorithm with $O(n^4 \text{ polylog})$ time.
- more efficient single-source queries and two-point queries
- extending to polyhedral weighted surfaces and to $\mathbb{R}^3$
- using more complicated weight functions, ex. anistropic ones

Open problems

- since the known worst-case lower bound on the number of event points in the continuous Dijkstra amid weighted regions is known to be $\Omega(n^4)$ (from [Mitchell, Papadimitriou JACM '91]) , the next objective could be to design an algorithm with $O(n^4 \text{ polylog})$ time.
- more efficient single-source queries and two-point queries
- extending to polyhedral weighted surfaces and to $\mathbb{R}^3$
- using more complicated weight functions, ex. anistropic ones
- several optimization problems in weighted regions, ex. tours, matchings, transportation, routing

References: polynomial time algo

Joseph Mitchell and Christos Papadimitriou.
The weighted region problem: Finding shortest paths through a weighted planar subdivision.
Journal of the ACM, 38(1):18–73, 1991.

R. Inkulu and Sanjiv Kapoor
A polynomial time algorithm for finding an approximate shortest path amid weighted regions.
Under review.
Available at CoRR abs/1501.00340.

References

Christian Mata and Joseph Mitchell.
A new algorithm for computing shortest paths in weighted planar subdivisions (extended abstract).
SoCG, pages 264–273, 1997.

Lyudmil Aleksandrov, Anil Maheshwari, and Jörg-Rüdiger Sack.
Determining approximate shortest path on weighted polyhedral surface.
Journal of the ACM, 52(1):25–53, 2005.

Zheng Sun and John H. Reif.
On finding approximate optimal paths in weighted regions.
Journal of Algorithms, 58(1):1–32, 2006.

S.-W. Cheng, H.-S. Na, A. Vigneron, and Y. Wang.
Querying approximate shortest paths in anisotropic regions.
SIAM Journal on Computing, 39(5):1888–1918, 2010.

Few more references

Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack.
Approximating shortest paths on weighted polyhedral surfaces.
Algorithmica, 30(4):527–562, 2001.

Lyudmil Aleksandrov, Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack.
An ϵ-approximation for weighted shortest paths on polyhedral surfaces
SWAT, pages 11-22, 1998.

Lyudmil Aleksandrov, Anil Maheshwari, and Jörg-Rüdiger Sack.
Approximation algorithms for geometric shortest path problems
STOC, pages 286-295, 2000.

Siu-Wing Cheng, J. Jin, and A. Vigneron.
Triangulation Refinement and Approximate Shortest Paths in Weighted Regions.
SODA, pages 1626–1640. SIAM, 2015.

Thanks!