Appl. Comput. Harmon. Anal. 35 (2013) 165–176

Contents lists available at SciVerse ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

Finding the shortest path by evolving junctions on obstacle boundaries (E-JOB): An initial value ODE's approach ☆

Shui-Nee Chow, Jun Lu *, Hao-Min Zhou

School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Article history:
Received 20 June 2012
Revised 29 October 2012
Accepted 6 November 2012
Available online 10 November 2012
Communicated by Charles K. Chui

Keywords:
Shortest path
Obstacles
Intermittent diffusion
Global optimization
Level set method
Stochastic differential equations

ABSTRACT

We propose a new fast algorithm (E-JOB) for finding a global shortest path connecting two points while avoiding obstacles in a region by solving an initial value problem of ordinary differential equations under random perturbations. The idea is based on the fact that every shortest path possesses a simple geometric structure. This enables us to restrict the search in a set of feasible paths that share the same structure. The resulting search set is a union of sets of finite dimensional compact manifolds. Then, we use a gradient flow, based on an intermittent diffusion method in conjunction with the level set framework, to obtain global shortest paths by solving a system of randomly perturbed ordinary differential equations with initial conditions. Compared to the existing methods, such as combinatorial methods or partial differential equation methods, our algorithm seems to be faster and easier to implement. We can also handle cases in which obstacle shapes are arbitrary and/or the dimension of the base space is three or higher.

1. Introduction

Finding the shortest path in the presence of obstacles is one of the fundamental problems in path planning and robotics. This problem can be described as follows. Given a finite set of obstacles in a region M in $\mathbf{R}^2$ or $\mathbf{R}^3$, how can one find the shortest path connecting two points X, Y in M while avoiding the obstacles. Many techniques have been developed. If the obstacles are polygonal, the problem can be reformulated as an optimization problem on a graph, and then solved by combinatoric methods. For example, in the shortest path map method, Hershberger and Suri [14] found an optimal $\mathbf{O}(n\log n)$ polynomial time algorithm where n is the total number of vertices of all polygonal obstacles in the plane $\mathbf{R}^2$. We refer to [17,20] for a survey of the results and references therein. However, Canny and Reif [5] proved that this problem in $\mathbf{R}^3$ becomes NP-hard under the framework known as "configuration space". A related study can also be found in [21]. This challenge motivates researchers to develop approximation algorithms. For example, Mitchell proposed an $\mathbf{O}(n\log n/\sqrt{\epsilon})$ complexity algorithm in [19] to find an ϵ-short path, which is a path that has length no more than $(1+\epsilon)L(\gamma_{opt})$, where $L(\gamma_{opt})$ is the length of the shortest path γ_{opt} and ϵ is a small positive number. Similar works can also be found in [1,6,7,10].

If the obstacles are not polygonal, the combinatorial methods cannot be applied directly. For this problem, commonly know methods are based on the theory of differential equations. For instance, in the planar path evolution approach, we can consider a one-parameter family of curves:

$$\gamma_t(\theta):[0,1]\to\mathbf{R}^2$$

☆ This work is partially supported by NSF Faculty Early Career Development (CAREER) Award DMS-0645266 and DMS-1042998.

* Corresponding author.
E-mail addresses: chow@math.gatech.edu (S.-N. Chow), jlu39@math.gatech.edu (J. Lu), hmzhou@math.gatech.edu (H.-M. Zhou).

1063-5203/$ – see front matter
http://dx.doi.org/10.1016/j.acha.2012.11.002

connecting $X = \gamma_t(0)$ and $Y = \gamma_t(1)$ defined by the following differential equation in t:

$$\frac{d\gamma_t(\theta)}{dt} = \left(\nabla W\left(\gamma_t(\theta)\right) \cdot \mathbf{n}(\theta)\right)\mathbf{n}(\theta) - W\left(\gamma_t(\theta)\right)\kappa(\theta)\mathbf{n}(\theta) \tag{1}$$

where $W : M \to \mathbf{R}$ is a penalty function, κ is the curvature of γ_t and $\mathbf{n}$ is the unit normal vector of γ_t. Eq. (1) is derived by shrinking the length of the path γ_t while avoiding the obstacles. If a path intersects with an obstacle, then the penalty W imposed on the path would push the curve out and towards a locally optimal path. By choosing different penalty functions, this method also works for other path planning problems. We refer to [13,28] for more discussions. However, since every point along the curve must be updated, this method may not be efficient, especially when the dimension of the problem is large. Moreover, it only leads to a locally optimal path. A different viewpoint of the path evolution method is to consider the steady state of (1) which satisfies

$$\nabla W\left(\gamma(\theta)\right) \cdot \mathbf{n} - W\left(\gamma(\theta)\right)\kappa = 0, \quad \gamma(0) = X,\ \gamma(1) = Y. \tag{2}$$

This is a two point boundary value problem. Its numerical computation may become costly, especially in three or higher dimensions.

Another PDE-based approach is called front propagation. The idea is to propagate a wave front from the starting point X with unit speed. The time the front first hits the ending point Y equals the length of the shortest path. It can be shown that the arriving time T satisfies a PDE known as the eikonal equation,

$$\left|\nabla T(x)\right| F(x) = 1, \quad T(X) = 0 \tag{3}$$

where $F(x)$ is the speed of the wave at point x. In this case, we have $F(x) = 1$. The eikonal equation can be solved efficiently by fast marching method [25,26] or fast sweeping [29,31]. Similarly, by choosing a different speed function $F(x)$, front propagation can be extended to solve other path planning problems [23].

Moreover, there are also recent developments on the shortest path problem focusing on other aspects, for example, how to read the obstacles data. For interested readers, we refer to [11,24,30].

In this paper, we present a new algorithm by solving an initial value problem for ordinary differential equations (ODE). The method is based on the geometric structure of all shortest paths which will be given in detail in the next section. Roughly speaking, in an environment with Euclidean metric, every shortest path must consist of segments of straight lines and curves that are parts of the obstacle boundaries, and those straight lines and curves are connected by points on obstacle boundaries. This structure enables us to restrict our search to a smaller subset of all feasible paths which share the same structure with every shortest path. Any path in this subset is determined uniquely by some connecting points on the boundaries of the obstacles. In order to find a shortest path, we evolve the connecting points along the boundaries. At the connecting points determined by stationary solutions of the ODE's, we will obtain paths which are locally and/or globally optimal paths. The number of points in the ODE system may be adjusted while evolving according to the ODE's, depending on whether points are eliminated or added to the system. In this case, the number of equations (or the dimension of the ODE's) is adaptively changed. This is another special feature of our new ODE-based shortest path framework.

The ODE's may have multiple steady state solutions, with each one corresponding to a path, which may be a locally optimal solution. In order to obtain a global shortest path, we employ the intermittent diffusion (ID) method [9] to add random perturbations to the ODE's on some discontinuous time intervals. The resulting equations alternate between stochastic and deterministic in time. This procedure helps us to find a globally optimal solution with probability arbitrarily close to one.

We will refer to our algorithm as Evolving Junctions on Obstacle Boundaries, or E-JOB for short from now on. E-JOB has several advantages:

(1) Ability to deal with any shape of obstacles. We incorporate level set framework for the boundaries of the obstacles to handle complicated geometry and topology.
(2) Ability to find a globally optimal path. We use the ID method to promote solutions out of the traps of local minima and obtain a globally optimal path.
(3) Dimension independent. The strategy can be applied to arbitrary dimension.
(4) Very fast. Since we solve an initial value problem of ODE's, the results can be obtained efficiently by various established schemes.

We present our results and algorithm in $\mathbf{R}^2$ in this paper. For $\mathbf{R}^3$ or higher dimensional problems, our methods are still applicable with nominal changes provided some mild restrictions are imposed on the boundary of the obstacles. In fact, the ideas introduced in this paper can be extended to other setups such as the shortest path in a general length space (see, for example, [4]) on which generalized gradient flows can be defined [3]. For simplicity, we will not use these setups here. The paper is arranged in the following way: in Section 2, we state the main theorem concerning the structure of the shortest path. Based on this theorem, we adopt a gradient descent strategy with the ID method for global optimization and give the algorithm. In Section 3, we discuss numerical implementation and results. The proof of the main theorem is presented in the last section.

2. The new algorithm

In this section, we present our new algorithm for the shortest path problem. We start with a mathematical description of the problem, through which we introduce notation needed in the rest of the paper.

A path is a curve γ in R^n, $n \geqslant 2$, which is a continuous map:

$$\gamma(\cdot) : [0,1] \to \mathbf{R}^n.$$

The length of γ under the Euclidean metric can be defined by many equivalent ways, here we use the definition given by

$$L(\gamma) = \sup_J \sup_{0=\theta_0 \leqslant \cdots \leqslant \theta_J = 1} \sum_{k=0}^{J-1} \left\| \gamma(\theta_k) - \gamma(\theta_{k-1}) \right\|,$$

where the norm is the Euclidean distance. γ is said to be rectifiable if $L(\gamma)$ is finite.

Let $\{R_i\}_{i=1}^N$ be N rectifiable closed Jordan curves in a given open connected region M in $\mathbf{R}^2$. By the Jordan curve theorem, each R_i divides $\mathbf{R}^2$ into two open connected regions. One of the regions is bounded and is called the interior of the curve R_i, denoted by $\operatorname{Int} R_i$. The other region is unbounded and called the exterior $\operatorname{Ext} R_i$. Each $\operatorname{Int} R_i$ represents an obstacle. We assume they are all pairwise disjoint and the set of all obstacles is a subset of M.

We denote F the set of all feasible paths $\gamma : [0,1] \to M$ such that

$$\gamma \in \left(\bigcap_{i=1}^N \operatorname{Ext} R_i \right) \cup \left(\bigcup_{i=1}^N (R_i) \right)$$

and $\gamma(0) = X$, $\gamma(1) = Y$. Here X and Y are two given points outside of the obstacles in M. Then the shortest path connecting X and Y is given by:

$$\gamma_{opt} = \operatorname*{argmin}_{\gamma \in F} L(\gamma).$$

The shortest path γ_{opt} possesses a simple geometric structure described in the following theorem.

Theorem 1. *Let the boundaries R_i of the obstacles be piecewise C^2 and the total number of points of C^2-discontinuity is finite. Let γ_{opt} be an optimal solution to the shortest path problem. Then there exist intervals $\{I_k \subset [0,1]\}$ such that every $\gamma_{opt}(t)|_{t \in I_k}$ is on the boundary of one obstacle. Outside these intervals, $\gamma(t)$ is a union of straight line segments. Moreover, each line segment is tangent to the obstacles.*

We leave the proof of the theorem to the last section.

Theoretically, the number of intervals $\{I_k\}$ may be countably infinite. This may happen when the boundaries of the obstacles have infinitely many bumps. On the other hand, these bumps on the boundaries must be approximated by a finite number of bumps in order to carry out the implementation in practice. For this reason, we assume that there are finitely many intervals I_k, $1 \leqslant k \leqslant n$ in this paper. We also remark that the C^2 assumption in the theorem imposed on the boundaries is for technical reasons in the proof. This can be relaxed in the implementation.

This structure theorem enables us to restrict our search of optimal paths in a subset H of the set F of all feasible paths. The set H is the collection of all paths that are determined by connecting points on the boundaries. More precisely, for any path $\gamma \in H$, there exists a sequence of points $(x_0, x_1, x_2, \ldots, x_{n-1}, x_n, x_{n+1})$ of γ with $x_0 = X$, $x_{n+1} = Y$ such that each x_i is a connecting point on the boundary R_{n_i} of an obstacle. It is connected to x_{i-1} and x_{i+1} by either a straight line segment outside of the obstacles, or a curve that is part of R_{n_i}. It follows from Theorem 1 that $\gamma_{opt} \in H$, i.e.

$$\operatorname*{argmin}_{\gamma \in F} L(\gamma) = \operatorname*{argmin}_{\gamma \in H} L(\gamma).$$

As shown in Fig. 1, for each point x_i there are two cases on how it is connected to the points before and after. In the first case (the left picture), x_{i-1} and x_i are connected by a straight line, and x_{i+1} is connected to x_i by a curve on the boundary. In this case, we denote x_{i-1} by x_i^s and x_{i+1} by x_i^c. The sup-index s (or c) represents that the two points are connected by a straight line (or curve). In the second case, as shown by the right picture of Fig. 1, we have $x_i^c = x_{i-1}$ and $x_i^s = x_{i+1}$. These notations can be extended to all connecting points including x_0 and x_{n+1}, if we assume $x_0^c = x_0, x_{n+1}^c = x_{n+1}$. In both cases, x_i and x_i^c divide the boundary R_{n_i} into two parts, R_i^+ and R_i^-, where R_i^+ is the arc from x_i to x_i^c with the counterclockwise direction and R_i^- the clockwise direction. Because R_i is rectifiable, the arc lengths of R_i^+ and R_i^{-1}, denoted by $d_i^+(x_i, x_i^c)$ and $d_i^-(x_i, x_i^c)$ respectively, are finite. The distance between x_i and x_i^c along the boundary is defined by

$$d_i\left(x_i, x_i^c\right) = \min\left\{d_i^+\left(x_i, x_i^c\right), d_i^-\left(x_i, x_i^c\right)\right\}.$$

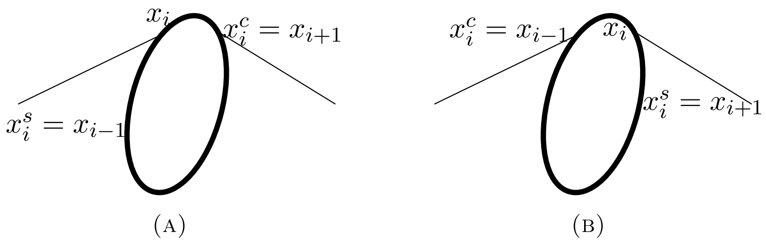


Fig. 1. Each connecting point on a boundary is connected to the points before and after it by a straight line segment and an arc of the boundary.

Furthermore, for each x_i, if we define

$$J(x_i) = \|x_i - x_i^s\| + d_{n_i}(x_i, x_i^c). \tag{4}$$

Then the length of path $\gamma \in H$ is

$$L(\gamma) = \mathcal{L}(x_1, x_2, \dots, x_n) = \frac{1}{2}\sum_{i=0}^{n+1} J(x_i). \tag{5}$$

2.1. Gradient descent

The simple form of the length functional defined on paths in H reduces the problem into a finite dimensional problem. To find the optimal path, we only need to find the optimal positions of the connecting points $\{x_i\}$. This can be obtained via the *gradient descent* of $\mathcal{L}$ in H, which leads to an initial value problem of ODE's given in the following proposition.

Proposition 2. *The gradient flow corresponding to the length functional $\mathcal{L}$ in H is*

$$\frac{dx_i}{dt} = -\nabla\mathcal{L}(x_1, \dots, x_i, \dots, x_n) = -\nabla J(x_i). \tag{6}$$

Moreover, if R_i is C^2, then

$$\nabla J(x_i) = \left(\frac{x_i - x_i^s}{\|x_i - x_i^s\|}\cdot \mathbf{T}\right)\mathbf{T} + \operatorname{sign}\big(d^+(x_i, x_i^c) - d^-(x_i, x_i^c)\big)\mathbf{T} \tag{7}$$

where $\mathbf{T}$ is the unit tangent at x_i to the boundary R_i in counter-clockwise direction.

Proof. Let $p \in R_i$ and $T_p(R_i)$ be the tangent space of R_i at p. By definition, ∇J satisfies

$$\langle \nabla J, \mathbf{X}\rangle_{x_i} = \mathbf{X}(J)_{x_i}, \quad \forall x_i \in R_{n_i},\ \mathbf{X} \in T_{x_i}(\mathbf{R}_{n_i}). \tag{8}$$

Let $\mathbf{X} = \mathbf{T}$ and $r(t), t \in [0, 1]$ be the arc length parametrization of R_{n_i} with $r(0) = x_i$ and $r'(0) = \mathbf{T}$. By the definition of J and a direct computation, we have the following for $t = 0$:

$$\frac{dJ(r(0))}{dt} = \frac{x_i - x_i^s}{\|x_i - x_i^s\|}\cdot \mathbf{T} + \operatorname{sign}\big(d^+ - d^-\big). \tag{9}$$

Combining (8) and (9), we get Eq. (6). □

Remark 1. The computation of $\nabla J(x_i)$ also holds if we only assume R_i to be piecewise C^2. At a point of C^2 discontinuity, we can choose $\mathbf{T}$ to be either the left tangent or the right tangent vector.

The above proposition says that the time evolution of the points $x_1, x_2, \dots, x_n$ on the boundary R_i is the gradient flow (6) of the total length functional $\mathcal{L}$. However, the number of connecting points may change during the evolution of $(x_1, \dots, x_n)$. For example, if there exists x_k such that the line segment $\overline{x_k x_k^s}$ intersects with an obstacle R_j, i.e.

$$\overline{x_k x_k^s} \cap (\operatorname{Int} R_j \cup R_j) \neq \emptyset, \tag{10}$$

then we add the intersection points as new connecting points on R_j. Without loss of generality, let us assume $x_k^s = x_{k+1}$. We denote $\{y_k\}$ as the new set of connecting points, which are $y_i = x_i, 1 \leqslant i \leqslant k$; $y_i = x_{i-2}, i \geqslant k+3$, and $y_{k+1} = y_{k+2}$ is the touching point. Then the lengths of the curves determined by $\{x_k\}$ and by $\{y_k\}$ are the same,

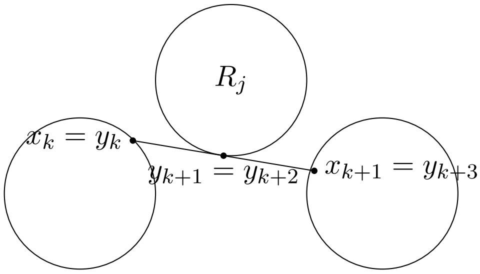


Fig. 2. New connecting points y_{k+1} and y_{k+2} can be added to the ODE system if the straight line $\overline{x_k x_k^s}$ intersects with another obstacle R_j. And the dimension of H is increased if new points are added.

$$\begin{aligned}\mathcal{L}(x_1,\dots,x_n) - \mathcal{L}(y_1,\dots,y_{n+2}) &= \|x_k - x_{k+1}\| - \big(\|y_k - y_{k+1}\| + d_j(y_{j+1}, y_{j+2}) + \|y_{k+2} - y_{k+3}\|\big)\\ &= \|x_k - x_{k+1}\| - \big(\|x_k - y_{k+1}\| + 0 + \|y_{k+2} - x_{k+1}\|\big) = 0.\end{aligned} \tag{11}$$

With the new connecting points, we have another gradient flow for $\{y_k\}$ which is also generated by (6). However, the number of equations is strictly larger. On the other hand, if two points on the same boundary meet each other, then we eliminate the points from the set of the connecting points. In this case, the number of connecting points decreases, thus the number of equations in (6) decreases. The insertion and elimination of points certainly changes the number of connecting points and their labels in the gradient descent process. For ease of presentation, we still denote them by a list $(x_1,\dots,x_n)$, even though x_i may correspond to different connecting points and n can be different integer values in the evolution steps. The details of adding or eliminating points are given in the algorithm presented in the next section.

The gradient flow (6) provides an explicit formula to move the points x_i on the boundaries. Their steady states includes all local optimal positions which define the locally optimal path. The number of steady states could be large in the shortest path problem. In certain situations, this can be estimated. For example, let the obstacles be smooth. If we have $N \geqslant 1$ obstacles and $2N$ connecting points, then the length functional $\mathcal{L}$ is actually a smooth scalar-valued function on a torus of dimension $2N$, i.e.

$$\mathcal{L} : T^{2N} \to \mathbf{R}.$$

If $\mathcal{L}$ is a Morse function (i.e., all critical points of $\mathcal{L}$ are non-degenerate), then there are at least 2^{2N} distinct critical points [8]. Each minimal point defines a minimum path. Obviously, the exponential growth of the number of critical points imposes a great challenge on the gradient descent method. Furthermore, as the number of connecting points changes, the dimension of the torus (phase space of the gradient flow (6)) also changes. Thus, a global optimization technique must be used to find a globally optimal path. In this paper, we adopt the intermittent diffusion strategy to address this problem.

2.2. Global optimization by intermittent diffusion

It is well known that the global optimization is a classical yet challenging problem. In general, it can be posed as finding the global minimizers for an objective functional

$$\min E(x), \quad x \in \Omega$$

where Ω is an admissible set. In our problem, E is the length functional $\mathcal{L}$ and Ω is the set F of all possible paths. There is a large literature on this subject and we refer to [2,18,16,27] for more information.

We will apply the intermittent diffusion strategy developed in [9] for our problem. Thus, we consider the following stochastic differential equation

$$dx(t,\omega) = -\nabla E\big(x(t,\omega)\big)\,dt + \sigma(t)\,dW(t), \quad t \in [0,\infty], \tag{12}$$

where $W(t)$ is a Brownian motion, and σ is a piecewise constant function given by

$$\sigma(t) = \sum_{i=1}^{n} \sigma_i \mathbf{1}_{[S_i,T_i]}(t),$$

with $0 = S_1 < T_1 < S_2 < T_2 < \cdots < S_n < T_n < S_{n+1} = T$, and $\mathbf{1}_{[S_i,T_i]}$ being the indicator function of interval $[S_i, T_i]$. It is proved that the intermittent diffusion can find the globally optimal solution with probability arbitrarily close to 1 provided T is large enough.

For our problem, since the points are moving on the boundaries, we add one dimensional noise to the gradient flow as follows:

$$dx_i = -\nabla J(x_i)\,dt + \sigma(t)\mathbf{T}\,dW(t). \tag{13}$$

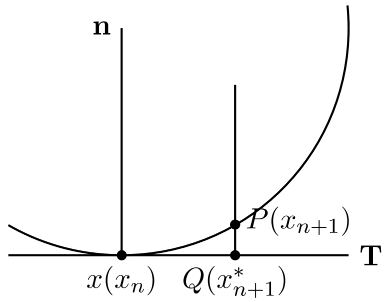


Fig. 3. The projection from the tangent direction to the boundary used in Lemma 3 and Eq. (18).

2.3. Algorithm

Now, we are ready to present our algorithm:

(1) Initialization. The initial connecting points consist of all the intersection points of the straight line $\overline{XY}$ with the boundaries of the obstacles.
(2) Update all moving point x_i by computing the stochastic equation (13) for $t \in [0, T]$ with $x(0) = x_i$ and record final state $x_T = x(T)$. In each time step of updating the moving points, add or remove points according to the following cases:
  (a) Adding moving points. If $\overline{x_i x_i^s}$ or $\overline{x_i x_i^c}$ intersects with obstacles, we add the intersection points into the set of moving points.
  (b) Eliminating moving points. If $x_i = x_i^c$, then we remove x_i and x_i^c from the set of moving points. And add the intersection points if $\overline{x_i^s x_j^s}$, where $x_j = x_i^c$, intersects with the obstacles.
(3) Update all moving point x_i by the gradient flow (6) until a convergence criterion is satisfied. And record the path connecting x_i at the final states. In each time step of updating the moving points, add or remove points according to cases (a) and (b) respectively as described in step (2).
(4) Repeat (2)–(3) N times to obtain N sample paths and then sort them to obtain the optimal one.

In the algorithm, the moving points x_i are updated only on the boundaries of obstacles, which are represented by their level set functions in our implementation. The boundary of an obstacle is the zero level set of a signed distance function. We achieve this by projecting each step of solving (13) and (6) on the zero level set of the obstacles. In addition, we must compute the intersections of the line segments with the boundaries of obstacles. We accomplished this in the same level set framework. To solve (13) and (6), different schemes can be used. In the next section, we give our numerical implementations of each step in the algorithm in detail.

3. Numerical implementation

We use a level set representation, the signed distance function [22], to implicitly express the boundaries of the obstacles. More precisely, let $d(x)$ be the distance function of the boundary of obstacle P, i.e. $d(x) = \min_{y \in \partial P} \|x - y\|$. The signed distance function $\phi(x)$is then defined as

$$\phi(x) = \begin{cases} d(x), & x \text{ is outside } P, \\ -d(x), & x \text{ is inside } P. \end{cases} \tag{14}$$

Under this representation, the outward unit normal direction at x on the boundary ∂P is simply

$$\mathbf{n} = \nabla \phi \tag{15}$$

and the curvature at x can be computed by

$$\kappa = \nabla \cdot \nabla \phi. \tag{16}$$

The connecting points should move only on the boundaries. To ensure this, we use the following lemma to project the updates along the tangent directions to the boundaries as shown in Fig. 3.

Lemma 3. *Let α be a planar curve, $\mathbf{T}$ and $\mathbf{n}$ are the tangent and normal directions at one point x on α. l is a line that is parallel to $\mathbf{n}$, and l intersects with α and $\mathbf{T}$ at P and Q respectively. Let h denote the lengths of $\overline{PQ}$, d the length of $\overline{xQ}$, and κ the curvature at x, then*

$$|\kappa| = \lim_{d \to 0} \frac{2h}{d^2}.$$

Proof. Let's assume α is arc-length parametrized and $x=\alpha(0)$. In the neighborhood of x, α has Taylor expansion

$$\alpha(t)=x+t\mathbf{T}+\frac{t^2}{2}\kappa\mathbf{n}+\mathbf{o}\big(t^2\big).$$

Therefore, $P=x+d\mathbf{T}+d^2/2\kappa\mathbf{n}+\mathbf{o}(d^2)$, $Q=x+d\mathbf{T}$, hence

$$\lim_{d\to 0}\frac{2h}{d^2}=\lim_{d\to 0}\frac{d^2|\kappa|+\mathbf{o}(d^2)}{d^2}=|\kappa|. \qquad \square$$

This lemma enables us to project the gradient flow from the tangent space to the manifold very easily. For the convenience of the presentation, let us denote

$$f(x)=-\left(\frac{x_i-x_i^s}{\|x_i-x_i^s\|}\cdot\mathbf{T}\right)-\operatorname{sign}\big(d^+\big(x_i,x_i^c\big)-d^-\big(x_i,x_i^c\big)\big).$$

Then (6) becomes

$$\frac{dx}{dt}=f(x)\mathbf{T}. \tag{17}$$

This can be computed by evolving the points in the tangent space followed by a projection to the boundary. More precisely, as shown in Fig. 3, we first compute x_{n+1}^* in the tangent space by

$$x_{n+1}^*-x_n=f(x_n)\Delta t\mathbf{T}.$$

Then the projected point x_{n+1} is the point on the boundary such that $x_{n+1}-x_{n+1}^*$ is parallel to $\mathbf{n}$. By Lemma 3,

$$\big\|x_{n+1}-x_{n+1}^*\big\|=\frac{1}{2}|\kappa|\big\|x_n-x_{n+1}^*\big\|^2=\frac{1}{2}|\kappa|\big(f(x_n)\Delta t\big)^2. \tag{18}$$

The direction of $x_{n+1}-x_{n+1}^*$ depends on the sign of the curvature. It is easy to see the direction is $-\operatorname{sign}(\phi(x_n))\mathbf{n}$. Hence, we have

$$x_{n+1}-x_{n+1}^*=-\operatorname{sign}\big(\phi(x_n)\big)\frac{|k|(f(x_n)\Delta t)^2}{2}\mathbf{n}. \tag{19}$$

We remark that the projection from the tangent space to the boundary can be accomplished in other ways, which may not depend on the curvature. This is appropriate especially when the boundaries are not smooth. The performance of the algorithm is similar when using different projection methods. One particular case is if arc length parametrization is used in the computation, one can add noise directly to the parameter without using projections.

To discretize (13), let x_{n+1}^* be the point along the tangent direction such that

$$x_{n+1}^*-x_n=f(x_n)\Delta t\mathbf{T}+\sigma_n\sqrt{\Delta t}\xi\mathbf{T}, \tag{20}$$

where $\xi\in N(0,1)$ is a standard normal random variable. σ_n here is also chosen to be random within a range $[a,b]$. The range depends on the size of the obstacles. If the obstacle is large, large σ needs to be selected in order for the path to jump out of the local traps. For the examples below, σ is taken in $[1,2]$.

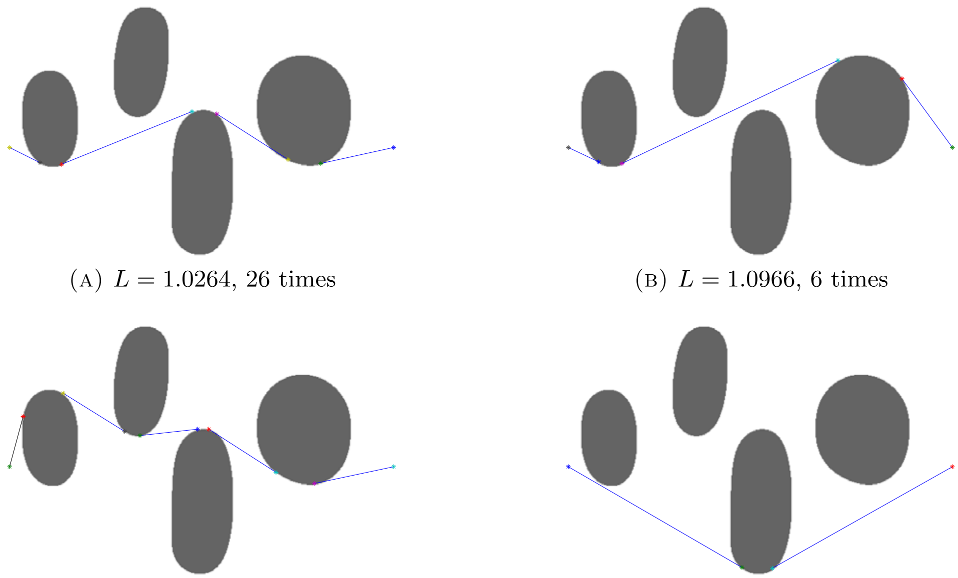
The arc lengths $d^+(x_i,x_i^c)$ and $d^-(x_i,x_i^c)$ can be computed in the same manner, i.e., solve (17) with $f(x)=1$ or $f(x)=-1$ respectively with initial condition $x(0)=x_i$, record the time t^+ or t^- it hits x_i^c, then we have $d^+(x_i,x_i^c)=t^+\Delta t$ and $d^-(x_i,x_i^c)=t^-\Delta t$.

3.1. Numerical results

In this section, we illustrate the performance of our method by showing the following examples.

Example 1. There are four obstacles in this example shown in Fig. 4. The starting point and the ending point are $X=[0.5,0.02]$ and $Y=[0.5,0.98]$ respectively. We choose $N=50$ and obtain 50 sample paths. The algorithm finds 4 different locally minimal paths as shown in Fig. 4. Among them 4(A) is the global minimal path. And it is observed 26 times, which is far more than the other locally optimal solutions.

Example 2. In this example, there are two obstacles which form a tunnel in Fig. 5. This is a non-convex case. We choose $X=[0.5,0.02]$, $Y=[0.5,0.98]$. The algorithm finds two local minimal paths with the global minimizer passing through the tunnel as shown in Fig. 5(A). The algorithm finds the global minimizer 46 times, which is much more frequent than visiting the other local minima.



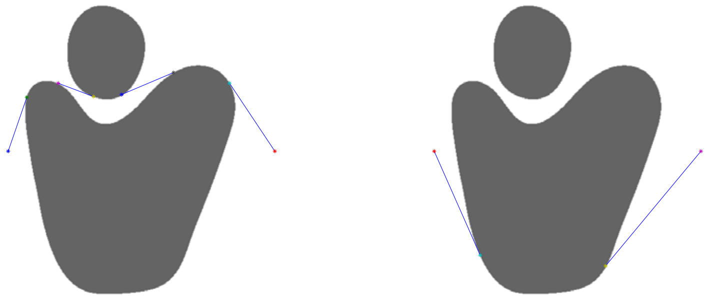
(A) $L = 1.0264$, 26 times

(B) $L = 1.0966$, 6 times

(C) $L = 1.1320$, 9 times

(D) $L = 1.1501$, 9 times

Fig. 4. Example 1, the algorithm finds 4 different shortest paths in one trials.



(A) $L = 1.1874$, 46 times

(B) $L = 1.3393$, 4 times

Fig. 5. Example 2, the algorithm finds 2 different shortest paths in one trial.

Although our algorithm is presented for 2-D cases, the idea can be extended to 3 or more dimensions with minor modifications. The main change is that the general implementation in higher dimensions needs the shortest length between two moving points on the surface which can be computed by the fast marching method [15] on the surface defined by the boundary of an obstacle. Here, we show an example in 3-D.

Example 3. In this example, the obstacles consist of two balls: one is centered at $[0.5, 5/16, 0.5]$ with radius $3/16$ and the other one is centered at $[0.5, 11/16, 0.5]$ with radius $1/8$. The starting point and the ending point are $X = [0.5, 1/16, 0.5 - \sqrt{3}/24]$ and $Y = [0.5, 15/16, 0.5]$ respectively. The algorithm finds three local minimizers in 30 runs as shown in Figs. 6 and 6(C) is the global minimal path. It was visited 27 times, which dominants the frequency of appearance.

4. The structure of the shortest path

In this section, we prove Theorem 1. It shows that the shortest path possesses a nice structure, i.e, the optimal path is a union of straight line segments outside the obstacles and portions of the boundaries otherwise. We validate this claim by defining a new metric in the region such that the metric inside the obstacles can be arbitrarily large. We show that the shortest path in the new metric can be arbitrarily close to the shortest path of the original problem. On the other hand, the shortest path in the new metric is a geodesic whose structure is described in the theorem.

For simplicity, we assume the boundaries of all obstacles to be C^2 in this section.

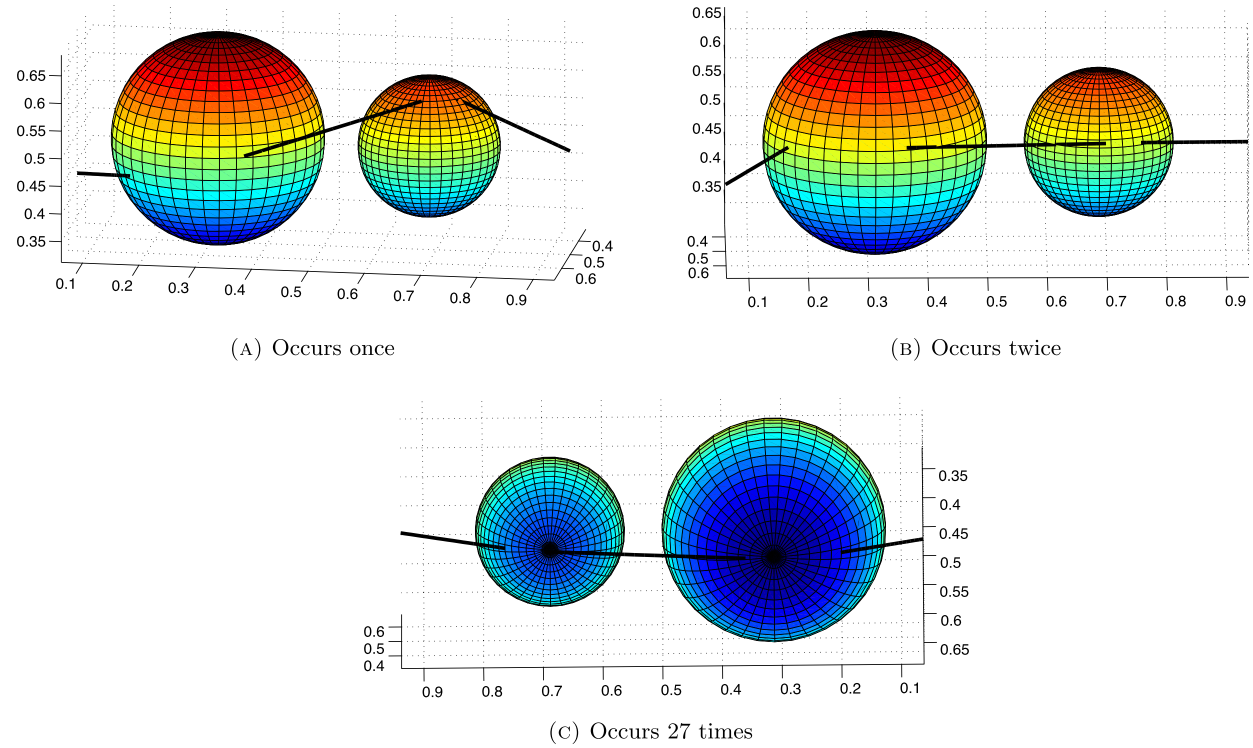


Fig. 6. Example 3, the algorithm finds 3 different shortest paths in one trial.

4.1. A new metric

Define a continuous function $g : \mathbf{R} \to \mathbf{R}$ as follows:

$$g(x) = \begin{cases} B, & x < -\epsilon, \\ \text{smooth and decreasing}, & -\epsilon \leqslant x < 0, \\ 1, & x \geqslant 0. \end{cases} \tag{21}$$

Here B is a large number which will be determined later.

Now define the following Riemannian metric: at each $p \in M$, $g_p : T_pM \times T_pM \to \mathbf{R}$ is

$$g_p(\mathbf{x}, \mathbf{y}) = g\big(d(p, \Gamma)\big)^2 \langle \mathbf{x}, \mathbf{y} \rangle.$$

Here, Γ is the union of the boundaries of all obstacles, $d(p, \Gamma)$ is the signed distance between p and Γ, and $\langle \mathbf{x}, \mathbf{y} \rangle$ is the usual inner product in $\mathbf{R}^2$.

We note that M is a smooth manifold endowed with this metric. For any feasible curve γ, we denote $\mathbb{L}_{\text{new}}(\gamma)$ and $\mathbb{L}_{\text{old}}(\gamma)$ the length of γ under the new metric and the old metric (the Euclidean metric) respectively. For example, if $\gamma : [0, 1] \to M$ is C^2,

$$\mathbb{L}_{\text{new}}(\gamma) = \int_0^1 \big|\gamma'(t)\big| g\big(d\big(\gamma(t), \Gamma\big)\big)\, dt.$$

4.2. The structure of the optimal path

Let G denote the set of paths connecting X and Y in M. Let $\alpha : [0, 1] \to M$ be the shortest path in G under the new metric, i.e.

$$\alpha(t) = \operatorname*{argmin}_{\gamma \in G} \mathbb{L}_{\text{new}}(\gamma).$$

Since M is a smooth Riemannian manifold, α is a geodesic in M [12]. Therefore, α is straight line segment outside the obstacles.

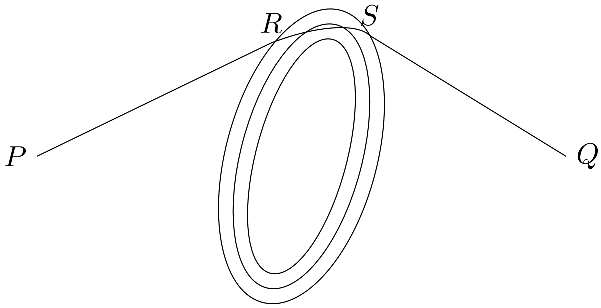


Fig. 7.

For each point x on the boundary of P_i, we can associate it with a point inside P_i in the normal line with a distance of ϵ to x. All those points form another C^2 curve. Denote the domain enclosed by such curves by P_i^{ϵ}. Similarly, we define $P_i^{2\epsilon}$ and hence $P_i^{2\epsilon} \subset P_i^{\epsilon} \subset P_i$. It is not hard to show that ∂P_i^{ϵ} is simple provided ϵ is sufficiently small. Also we have the following:

Lemma 4. *There exists a constant B in (21) such that α is entirely outside $P_i^{2\epsilon}$.*

Proof. For any $p, q \in \partial P_i^{\epsilon}$, denote $\overline{pq}$ the line segment connecting p, q. Let

$$S = \left\{(p, q) \in \partial P_i^{\epsilon} \times \partial P_i^{\epsilon} \;\middle|\; \overline{pq} \cap P_i^{2\epsilon} \neq \emptyset\right\}.$$

S is compact since S is equivalent to

$$S = \left\{(p, q) \in \partial P_i^{\epsilon} \times \partial P_i^{\epsilon} \;\middle|\; d\left(\overline{pq}, P_i^{2\epsilon}\right) \leqslant 0\right\}$$

and d is continuous. Moreover, the function $h(p, q) = \mathbb{L}_{\text{old}}(\overline{pq})$ where $(p, q) \in S$ is always positive. Therefore, h has a positive minimum l. Choose any path in the set of feasible paths F, denote its length under the new metric by L. Select B such that $lB > L$. Now for any two points p, q on ∂P_i^{ϵ}, if the line segment connecting them intersects with $P_i^{2\epsilon}$, then the length under the new metric should be greater than $lB > L$. This implies that α is outside $P_i^{2\epsilon}$ everywhere since α is straight in P_i^{ϵ}. □

Intuitively, when B becomes larger, the penalty imposed on the portion inside obstacles also becomes larger. Hence in order to reduce the length, the path mustn't pass through obstacles too much. Now restrict α on the interval $[S, T]$ where $\alpha(S), \alpha(T) \in \partial P_i$ and the image of α $\operatorname{Img}\alpha \subset P_i$ for $t \in [S, T]$. Denote $R = \alpha(S)$, $S = \alpha(T)$. See Fig. 7.

Denote $\tilde{\beta}(t)$ the arc length parametrization of ∂P_i between R and S and $\tilde{\alpha}(t)$ the part of α between R and S.

Lemma 5.

$$\mathbb{L}_{\text{old}}\left(\tilde{\alpha}(t)\right) \geqslant (1 - 4\epsilon\kappa)\mathbb{L}_{\text{old}}\left(\tilde{\beta}(t)\right) \tag{22}$$

where κ is the maximal curvature.

Proof. First we show that $\tilde{\alpha}(t)$ can be expressed as

$$\tilde{\alpha}(t) = \tilde{\beta}(t) + \epsilon(t)\mathbf{n}(t),$$

where $\epsilon(t) < 2\epsilon$ and $\mathbf{n}(t)$ is the unit inward normal, i.e., the geodesic α will intersect with the normal line only once. If not, for any $t \in [0, 1]$, let

$$I(t) = \left\{s \in [0, 1] \;\middle|\; \left(\tilde{\alpha}(s) - \tilde{\beta}(t)\right) \cdot \tilde{\beta}'(t) = 0\right\}$$

and $t_0 = \inf I(t)$, $t_1 = \sup I(t)$. Since $\tilde{\alpha}(1) = S$ is not on the normal line, we have $t_1 < 1$. Moreover, it is easy to see that $t_0, t_1 \in I(t)$ by the continuity of $(\tilde{\alpha}(s) - \tilde{\beta}(t)) \cdot \tilde{\beta}'(t)$. Thus $\tilde{\alpha}(s), s \in [t_0, t_1]$ must be entirely on the normal line due to its optimality. Now choose Δt small enough such that $\overline{\tilde{\alpha}(t_0)\tilde{\alpha}(t_1 + \Delta t)}$ is in the region formed by $\tilde{\alpha}$ and $\tilde{\beta}$. But $\overline{\tilde{\alpha}(t_0)\tilde{\alpha}(t_1 + \Delta t)}$ has smaller length, which is a contradiction.

This representation of $\tilde{\alpha}$ gives

$$\frac{d\tilde{\alpha}(t)}{dt} = \frac{\tilde{\beta}(t)}{dt} + \epsilon(t)\frac{d\mathbf{n}(t)}{dt} + \epsilon'(t)\mathbf{n}(t)$$

and

$$\begin{aligned}\left|\frac{d\tilde{\alpha}}{dt}\right|^2 &= \left(\tilde{\beta}'(t), \tilde{\beta}'(t)\right) + \epsilon(t)^2\left|\mathbf{n}'(t)\right|^2 + \epsilon'(t)^2 + 2\left(\tilde{\beta}'(t), \epsilon(t)\mathbf{n}'(t)\right) \\ &\quad + 2\left(\tilde{\beta}'(t), \epsilon'(t)\mathbf{n}(t)\right) + 2\epsilon(t)\epsilon'(t)\left(\mathbf{n}'(t), \mathbf{n}(t)\right) \\ &= 1 + \epsilon(t)^2\left|\mathbf{n}'(t)\right|^2 + \epsilon'(t)^2 + 2\epsilon(t)\left(\tilde{\beta}'(t), \mathbf{n}'(t)\right) \\ &\geqslant 1 - 2\epsilon(t)\kappa \geqslant 1 - 4\epsilon\kappa \geqslant (1 - 4\epsilon\kappa)^2\end{aligned}$$

for sufficiently small ϵ. Consequently,

$$\mathbb{L}_{\text{old}}\left(\tilde{\alpha}(t)\right) \geqslant \mathbb{L}_{\text{old}}\left(\tilde{\beta}(t)\right) - 4\epsilon\kappa\mathbb{L}_{\text{old}}\left(\tilde{\beta}(t)\right). \qquad \square$$

The constructed path $\alpha(t)$ may intersect with P_i multiple or even infinite times. However, its total length can be controlled. More specifically, we have the following

Lemma 6. *Suppose $\tilde{\alpha}_1(t)$, $t \in [0, 1]$ and $\tilde{\alpha}_2(t)$, $t \in [0, 1]$ are arc length parametrizations of two segments of α inside the same obstacle, and $\tilde{\beta}_1$, $\tilde{\beta}_2$ are two curves defined as in (22), i.e.*

$$\begin{aligned}\tilde{\alpha}_1(t) &= \tilde{\beta}_1(t) + \epsilon_1(t)\mathbf{n}_1(t), \\ \tilde{\alpha}_2(t) &= \tilde{\beta}_2(t) + \epsilon_2(t)\mathbf{n}_2(t).\end{aligned}$$

Then either $\operatorname{Img}(\tilde{\beta}_1) \subset \operatorname{Img}(\tilde{\beta}_2)$ or $\operatorname{Img}(\tilde{\beta}_1) \cap \operatorname{Img}(\tilde{\beta}_2) = \emptyset$. Moreover, if the former is true, then

$$\mathbb{L}_{\text{old}}(\tilde{\beta}_1) < \frac{\epsilon_1}{2 - \epsilon_1}\mathbb{L}_{\text{old}}(\tilde{\beta}_2).$$

where $\epsilon_1 = 4\epsilon\kappa$. In other words, the length decreases exponentially.

Proof. The first part of the claim is obvious. For the second part, notice

$$\begin{aligned}\mathbb{L}_{\text{old}}\left(\tilde{\alpha}_1(t)\right) &\geqslant \mathbb{L}_{\text{old}}\left(\tilde{\beta}_1(t)\right) - 4\epsilon\kappa\mathbb{L}_{\text{old}}\left(\tilde{\beta}_1(t)\right), \\ \mathbb{L}_{\text{old}}\left(\tilde{\alpha}_2(t)\right) &\geqslant \mathbb{L}_{\text{old}}\left(\tilde{\beta}_2(t)\right) - 4\epsilon\kappa\mathbb{L}_{\text{old}}\left(\tilde{\beta}_2(t)\right).\end{aligned}$$

By the definition of α, we have

$$\mathbb{L}_{\text{old}}(\tilde{\beta}_2) - \mathbb{L}_{\text{old}}(\tilde{\beta}_1) > \left(\mathbb{L}_{\text{old}}(\tilde{\beta}_2) + \mathbb{L}_{\text{old}}(\tilde{\beta}_1)\right)(1 - \epsilon_1).$$

Therefore,

$$\mathbb{L}_{\text{old}}(\tilde{\beta}_1) < \frac{\epsilon_1}{2 - \epsilon_1}\mathbb{L}_{\text{old}}(\tilde{\beta}_2). \qquad \square$$

Proof of the structure theorem. Let β be the curve consisting of the straight part of α and all the $\tilde{\beta}(t)$ parts as above, i.e. portions of the boundaries of P_i and γ_{opt} the shortest path of our original problem, i.e.

$$\gamma_{opt}(t) = \operatorname*{argmin}_{\gamma \in F} \mathbb{L}_{\text{old}}(\gamma).$$

First we show that $\mathbb{L}_{\text{old}}(\beta)$ and $\mathbb{L}_{\text{old}}(\gamma_{opt})$ can be arbitrarily close. In fact, by summing the inequality in (22), we get

$$\mathbb{L}_{\text{old}}(\alpha) \geqslant \mathbb{L}_{\text{old}}(\beta) - 4C\epsilon\kappa L$$

where L is the total perimeters of all the obstacles and $C = \sum_0^\infty (\frac{\epsilon_1}{2-\epsilon_1})^i$. This implies

$$\mathbb{L}_{\text{old}}(\beta) \geqslant \mathbb{L}_{\text{old}}(\gamma_{opt}) = \mathbb{L}_{\text{new}}(\gamma_{opt}) \geqslant \mathbb{L}_{\text{new}}(\alpha) \geqslant \mathbb{L}_{\text{old}}(\alpha) \geqslant \mathbb{L}_{\text{old}}(\beta) - 4C\epsilon\kappa L.$$

In other words,

$$\left|\mathbb{L}_{\text{old}}(\gamma_{opt}) - \mathbb{L}_{\text{old}}(\beta)\right| \leqslant 2C\epsilon\kappa L.$$

As a result, we only need to minimize $\mathbb{L}_{\text{old}}(\beta)$. By the construction of β, it can be represented by a series of points $(x_0, x_1, \ldots, x_{n+1})$ and

$$2\mathcal{L}(\beta) = \sum_{i=1}^{n} J(x_i).$$

For any i, x_i minimizes $J(x_i)$ only if

$$\nabla J(x_i) = \left(\frac{x_i - x_i^s}{\|x_i - x_i^s\|} \cdot \mathbf{T}\right)\mathbf{T} + \operatorname{sign}\left(d^+\left(x_i, x_i^c\right) - d^-\left(x_i, x_i^c\right)\right)\mathbf{T} = 0.$$

But

$$\left\| \frac{x_i - x_i^s}{\|x_i - x_i^s\|} \cdot \mathbf{T} \right\| \leqslant 1$$

which implies that $x_i - x_i^s$ is parallel to $\mathbf{T}$, i.e. $x_i - x_i^s$ is tangent to P_k. Therefore, if β minimizes $\mathcal{L}$, then all the straight parts of β should be tangent to a obstacle. By the inequality above, we know that this is also the solution to the original problem. $\square$

5. Conclusion

In this paper, we proposed a novel algorithm Evolving Junctions on Obstacle Boundaries (E-JOB) to find the shortest path that connects two given points while avoiding obstacles. Based on the simple structure of the shortest path as proposed in the previous sections, E-JOB computes the shortest path by solving an initial value problem of stochastic ODE's. Compared to existing strategies for this classical problem, our method has several compelling advantages. First of all, E-JOB is able to handle any shape of obstacles. In particular, it can be applied to polyhedrons which has been proven to be NP-hard. Secondly, E-JOB converges much faster than its PDE based counterparts since we only need to solve initial value ODE's. Lastly, the globally optimal path is guaranteed by the introduction of intermittent diffusion, a global optimization strategy.

E-JOB can also be applied to many other problems with some minor changes of the algorithm. For example, finding the shortest path when the mover is a disk instead of a point, computing the distance between two sets in $\mathbf{R}^n$, finding the shortest path when obstacles disappear or pop out, just to name a few. Moreover, the algorithm can be integrated into robotic navigation system which is a problem we are currently working on.

References

[1] P.K. Agarwal, R. Sharathkumar, H. Yu, Approximate euclidean shortest paths amid convex obstacles, in: Proceedings of the Twentieth Annual ACM–SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2009, pp. 283–292.

[2] F. Aluffi-Pentini, V. Parisi, F. Zirilli, Global optimization and stochastic differential equations, J. Optim. Theory Appl. 47 (1) (1985) 1–16.

[3] L. Ambrosio, N. Gigli, G. Savaré, Gradient Flows: In Metric Spaces and in the Space of Probability Measures, Birkhäuser, 2008.

[4] D. Burago, Y. Burago, S. Ivanov, A Course in Metric Geometry, American Mathematical Society, Providence, 2001.

[5] J. Canny, J. Reif, New lower bound techniques for robot motion planning problems, in: 28th Annual Symposium on Foundations of Computer Science, IEEE, 1987, pp. 49–60.

[6] D.Z. Chen, On the all-pairs Euclidean short path problem, in: Proceedings of the Sixth Annual ACM–SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 1995, pp. 292–301.

[7] D.Z. Chen, G. Das, M. Smid, Lower bounds for computing geometric spanners and approximate shortest paths, in: Proceedings of the 8th Canadian Conference on Computational Geometry , Citeseer, 1996.

[8] S.-N. Chow, J.K. Hale, Methods of Bifurcation Theory, vol. 138, Springer-Verlag, 1982.

[9] S.-N. Chow, T.-S. Yang, H. Zhou, Global optimizations by intermittent diffusion, in: Andrew Adamatzky, Guanrong Chen (Eds.), Chaos, CNN, Memristors and Beyond, World Scientific Publishing Company, 2013.

[10] K. Clarkson, Approximation algorithms for shortest path motion planning, in: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, ACM, 1987, pp. 56–65.

[11] R. Dias, N.L. Garcia, A.Z. Zambom, Monte Carlo algorithm for trajectory optimization based on Markovian readings, Comput. Optim. Appl. 51 (1) (2012) 305–321.

[12] M.P. do Carmo, Riemannian Geometry, Birkhäuser, 1992.

[13] R. Goroshin, Q. Huynh, H. Zhou, Approximate solutions to several visibility optimization problems, Commun. Math. Sci. 9 (2) (2011) 535–550.

[14] J. Hershberger, S. Suri, An optimal algorithm for Euclidean shortest paths in the plane, SIAM J. Comput. 28 (6) (1999) 2215–2256.

[15] R. Kimmel, J.A. Sethian, Computing geodesic paths on manifolds, Proc. Natl. Acad. Sci. USA 95 (15) (1998) 8431.

[16] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671.

[17] S.M. LaValle, Planning Algorithms, Cambridge Univ. Press, 2006.

[18] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al., Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087.

[19] J.S.B. Mitchell, L1 shortest paths among polygonal obstacles in the plane, Algorithmica 8 (1) (1992) 55–88.

[20] J.S.B. Mitchell, Shortest path and networks, in: Handbook of Discrete and Computational Geometry, 1997, pp. 755–778.

[21] J.S.B. Mitchell, M. Sharir, New results on shortest paths in three dimensions, in: Proceedings of the Twentieth Annual Symposium on Computational Geometry, ACM, 2004, pp. 124–133.

[22] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, 2003.

[23] M. Sabry Hassouna, A.E. Abdel-Hakim, A.A. Farag, PDE-based robust robotic navigation, Image Vision Comput. 27 (1–2) (2009) 10–18.

[24] A. Sedeno-Noda, C. Gonzalez-Martin, An efficient label setting/correcting shortest path algorithm, Comput. Optim. Appl. 51 (1) (2012) 437–455.

[25] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. USA 93 (4) (1996) 1591.

[26] J.A. Sethian, Level Set Methods and Fast Marching Methods – Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science, 2nd ed., Cambridge University Press, Cambridge, 1999.

[27] H. Szu, R. Hartley, Fast simulated annealing, Phys. Lett. A 122 (3–4) (1987) 157–162.

[28] Y.H.R. Tsai, L.T. Cheng, S. Osher, P. Burchard, G. Sapiro, Visibility and its dynamics in a PDE based implicit framework, J. Computational Physics 199 (2004) 260–290.

[29] Y.H.R. Tsai, L.T. Cheng, S. Osher, H.K. Zhao, Fast sweeping algorithms for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2) (2003) 673–694.

[30] M. Vasile, M. Locatelli, A hybrid multiagent approach for global trajectory optimization, J. Global Optim. 44 (4) (2009) 461–479.

[31] H. Zhao, A fast sweeping method for eikonal equations, Math. Comp. 74 (250) (2005) 603–628.