A FAST ALGORITHM FOR FINDING THE SHORTEST PATH BY SOLVING INITIAL VALUE ODE'S

SHUI-NEE CHOW, JUN LU, HAO-MIN ZHOU

ABSTRACT. We propose a new fast algorithm 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. Comparing to the existing methods, such as the 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 or a 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 [13] 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 [16, 19] for a survey of the results and references therein. However, Canny and Rief [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 [20]. This challenge motivates researchers to develop approximation algorithms. For example, Mitchell proposed an $\mathbf{O}(n \log n/\sqrt{\epsilon})$ complexity algorithm in [18] 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 combinatoric methods can not be applied directly. The commonly known methods are based on the theory of differential equations. For instance, in the planar path evolution approach, we can consider a one-parameter, denoted by t, family of curves:

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

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

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

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

where $W\colon M \to \mathbf{R}$ is a penalty function, κ is the curvature of γ_t and $\mathbf{n}$ is the unit normal vecotor of γ_t. Equation (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 local optimal path. By choosing different penalty functions, this method also works for other path planning problems. We refer to [12, 26] 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 local optimal path. A different viewpoint of the path evolution method is to consider the steady state of (1) which satisfies

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

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,

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

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 [23, 24] or fast sweeping [27, 28]. Similarly, by choosing different speed $F(x)$, front propagation can be extended to solve other path planning problems [22].

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, 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 in 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 by an initial value ODE problem. By the connecting points determined by stationary solutions of the ODE's, we will obtain paths which are local and/or global optimal paths. The number of points in the ODE system may be adjusted while evolving along the ODE's, depending 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, each one corresponds to a path, which may be a local 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 global optimal solution with probabilty arbitrary close to one.

This new method 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 global optimal path. We use the ID method to promote solutions out of the traps of local minima and obtain a global optimal path.
(3) Dimension independent. The strategy can be applied to arbitrary dimension.
(4) Very fast. Since we solve an initial value problem of ODEs, 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. We leave the proof for the main theorem 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 some notations needed in the rest of the paper.

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

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

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

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

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 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 $\gamma\colon [0,1] \to M$ such that

$$\gamma \in (\cap_{i=1}^N \operatorname{Ext} R_i) \cup (\cup_{i=1}^N (R_i))$$

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 simple geometric structures 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-discountinuity is finite. Let γ_{opt} be an optimal solution*

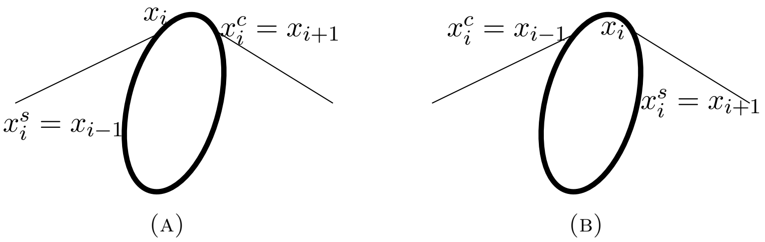


FIGURE 1. Each connecting point on a boundary is connected to the points before and after it by a straight line segment and an arc of the boundary.

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 $I_k, 1 \leq k \leq n$ 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 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 by either straight lines outside of the obstacles or curves 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 $x_0 = X$, $x_{n+1} = Y$. Furthermore, 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.

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

As shown in Figure 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 Figure 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

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

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

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

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 those moving points $\{x_i\}$. This can be obtained by applying *gradient descent* method to $\mathcal{L}$ in H, which leads 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*

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

Moreover, if R_i is C^2, then

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

where $\mathbf{T}$ is the unit tangent 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 arclength 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 $t = 0$:

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

Combining (8) and (9), we get equation (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 $x_1, x_2, \cdots, x_n$ 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 $(x_1, \ldots, x_n)$. 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 $y_i = x_i, 1 \leq i \leq k; y_i = x_{i-2}, i \geq k+3$, 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,

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

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

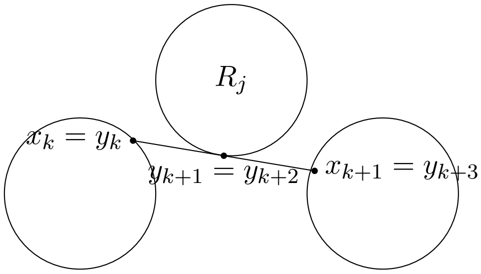


FIGURE 2. New connecting points y_{k+1} and y_{k+2} can be added to the ODE system if the straight line $\overline{x_k x_k^s}$ intersects with another obstacle R_j. And the dimension of H is increased if new points are added.

points decreases, and the number of equations in (6) decreases. The insertion and elimination of points certainly change the number of connecting points and their labels in the gradient descent process. For the presentation convenience, we still denote them by a list $(x_1, \cdots, x_n)$, even though x_i may correspond to different connecting points and n can be different integer values in the evolution steps. The detail of adding or eliminating points is 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 include all local optimal positions which define local minimal 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 $N \geq 1$ 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.

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

If $\mathcal{L}$ is a Morse function (i.e., all cririical 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 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 global 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 poosible paths. There is a large literature on this subject and we refer to [2, 17, 15, 25] for more information.

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

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

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 inverval $[S_i, T_i]$. It is proved that the intermittent diffusion can find the global 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:

$$dx_i = -\nabla J(x_i)dt + \sigma(t)\mathbf{T}dW(t). \tag{12}$$

2.3. **Algorithm.** Now, we are ready to present our algorithm:

(1) Initialization. The initial moving 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 (12) 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 case (a) and (b) respectively as described in step (2).
(4) Repeat (2)-(3) N times to obtain N sample paths and then compare them to obtain the optimal one.

In the algorithm, the moving points x_i are updated only on the boundaries of obstacles, which is represented by a level set framework in our implementation. The boundaries of obstacles are the zero level set of a signed distance function. We achieve this by projecting each step of solving (12) 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 (12) 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 [21], 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

$$\phi(x) = \begin{cases} d(x), & x \text{ is outside } P; \\ -d(x), & x \text{ is inside } P. \end{cases} \tag{13}$$

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

$$\mathbf{n} = \nabla\phi \tag{14}$$

and the curvature at x can be computed by

$$\kappa = \nabla \cdot \nabla \phi. \tag{15}$$

The connecting points move on the boundaries. To ensure it, we use the following lemma to project the updates along the tangent directions to the boundaries as shown in Figure **??**.

Lemma 3. *Let α be a planar curve, $\mathbf{T}$ and $\mathbf{n}$ 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. Denote h the lengths of $\overline{PQ}$, d the length of $\overline{xQ}$, and κ the curvature at x, then*

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

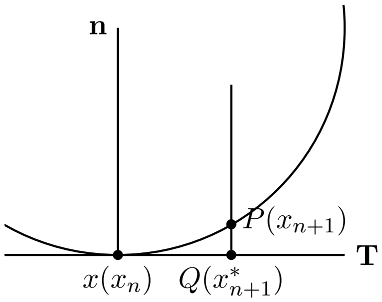


FIGURE 3. The projection from the tangent direction to the boundary used in Lemma 3 and equation 17.

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

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

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

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

□

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

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

Then (6) becomes

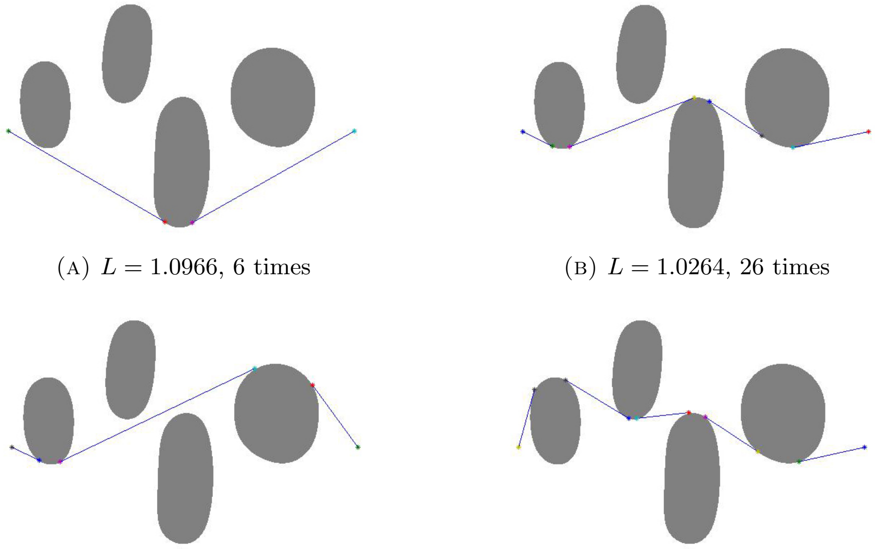
$$\frac{dx}{dt} = f(x)\mathbf{T}. \tag{16}$$

This can be computed by evolving the points in the tangent space followed by a projection to the boundary. More precisely, as shown in Figure **??**, 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),

$$\|x_{n+1} - x^*_{n+1}\| = \frac{1}{2}|\kappa|\|x_n - x^*_{n+1}\|^2 = \frac{1}{2}|\kappa|(f(x_n)\Delta t)^2. \tag{17}$$



(A) $L = 1.0966$, 6 times

(B) $L = 1.0264$, 26 times

(C) $L = 1.1320$, 9 times

(D) $L = 1.1501$, 9 times

FIGURE 4. Example 1, the algorithm finds 4 different shortest path in one trial. Among them, the global minimizer (B) is observed 26 time, far more than other local minimizers.

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

$$x_{n+1} - x^*_{n+1} = -\operatorname{sign}(\phi(x_n))\frac{|k|(f(x_n)\Delta t)^2}{2}\mathbf{n}. \tag{18}$$

We remark that the project from the tangent space to the boundary can be accomplished by 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 by using different projection methods.

To discretize (12), let x^*_{n+1} be the point along the tangent direction such that

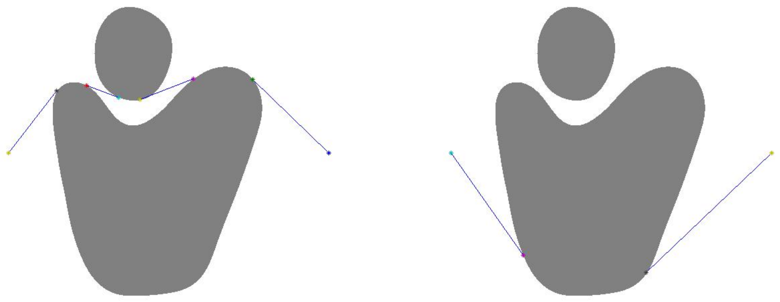
$$x^*_{n+1} - x_n = f(x_n)\Delta t\mathbf{T} + \sigma\sqrt{\Delta t}\xi\mathbf{T}, \tag{19}$$

where $\xi \in N(0,1)$ is a standard normal random variable. Projection point x_{n+1} is then obtained by the same projection as in (18).

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 (16) 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 Figure 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 local minimial paths as shown in Figure 4. Among them (4b) is the global minimal path. And it is observed 26 times, which is far more than the other local optimal solutions.



(A) $L = 1.1874$, 46 times (B) $L = 1.3393$, 4 times

FIGURE 5. Example 2, the algorithm finds 2 different shortest path in one trial. Among them, the global minimizer (A) is observed 46 time.

Example 2. In this example, there are two obstacles which form a tunnel in Figure 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 Figure (5a). The algorithm finds the global minimizer 46 times, which is much more frequent than visiting the other local minima.

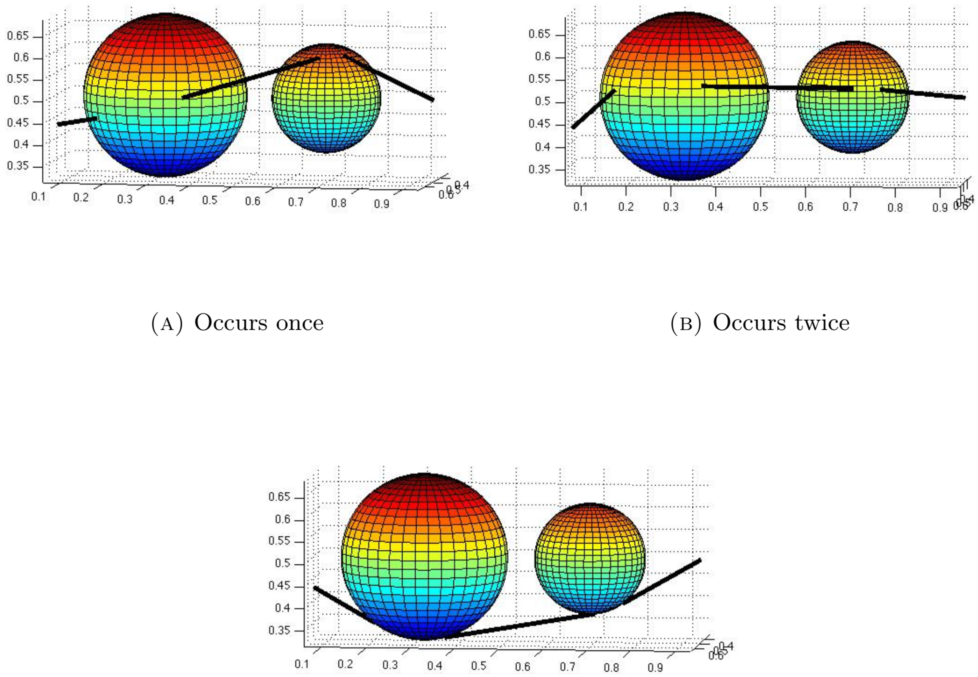
Although our algorithm are presented for 2-D cases, but the idea can be extended to 3 or higher dimension with minor modifications. The main change is that the general implementation in higher dimensions needs the shortest length between two moving points on the surface which can be computed by the fast marching method [14] 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 Figure 6 and (6c) 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. The theorem shows that the shortest path possesses a nice structure, i.e, the optimal path is 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.

We assume the boundaries of all obstacles to be C^2 in this section for technical reasons.



(A) Occurs once

(B) Occurs twice

(C) Occurs 27 times

Figure 6. Example 3, the algorithm finds 3 different shortest path in one trial. Among them, the global minimizer (C) is observed 27 time, which the other two local minimizers are only observed 1 (A) or 2 (B) times.

4.1. **A new metric.** Define a continuous function $f\colon \mathbf{R} \to \mathbf{R}$ as follows:

$$g(x) = \begin{cases} W, \quad x < -\epsilon, \\ \text{smooth and decreasing}, \quad -\epsilon \leq x < 0 \\ 1, \quad x >= 0. \end{cases} \tag{20}$$

Here W is a large number which is determined later.

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

$$g_p(X, Y) = g(d(p, \Gamma))^2 \langle X, Y \rangle .$$

Here, Γ is the union of the boundaries of all obstacles, $d(p, \Gamma)$ is the signed distance between p and Γ, and $\langle X, Y \rangle$ 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 the new metric and the old metric(the Euclidean metric) respectively. For example, if $\gamma\colon [0,1] \to M$ is C^2,

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

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

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

Since M is a smooth Riemannian manifold, α is a geodesic in M [11]. Therefore, α is straight line segment outside the obstacles.

At each point x on the boundary of P_i, we can find a corresponding 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 $P_i^{2\epsilon} \subset P_i^{\epsilon} \subset P_i$. We have the following lemma

Lemma 4. *There exists ϵ such that ∂P_i^{ϵ} is simple.*

Proof. Let $r\colon [0,1] \to M$ be an arc length parametrization of ∂P_i^{ϵ}. Assume that ∂P_i^{ϵ} is not simple. Then for every k, there exist t_k, s_k such that

$$r(t_k) + \frac{1}{k}\mathbf{n}(t_k) = r(s_k) + \frac{1}{k}\mathbf{n}(s_k).$$

Since $t_k \in [0,1]$, it has a accumulating point t. Let $t_{n_k} \to t$. But

$$\|r(t_{n_k}) - r(s_{n_k})\| = \frac{1}{k}\|\mathbf{n}(s_{n_k}) - \mathbf{n}(t_{n_k})\| \leq \frac{2}{k},$$

and $r(t)$ is simple, we obtain $s_{n_k} \to t$. Therefore,

$$\frac{r(t_{n_k}) - r(s_{n_k})}{t_{n_k} - s_{n_k}} = -\frac{1}{k}\frac{\mathbf{n}(t_{n_k}) - \mathbf{n}(s_{n_k})}{t_{n_k} - s_{n_k}}$$

Let $k \to \infty$, we get

$$r'(t) = 0.$$

This is a contradiction to the fact that $r'(t) = 1$. □

Proposition 5. *There exists W in (20) 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

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

S is compact since S is equivalent to

$$S = \{\, (p,q) \in \partial P_i^{\epsilon} \times \partial P_i^{\epsilon} \mid d(\overline{pq}, P_i^{2\epsilon}) \leq 0 \,\}$$

and d is continous. Moreover, 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 F, denote its length under the new metric by L. Select W such that $lW > L$. 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 $lW > L$. This implies that α is outside $P_i^{2\epsilon}$ everywhere since α is straight in P_i^{ϵ}. □

Let's restrict α on the interval $[S,T]$ where $\alpha(S), \alpha(T) \in \partial P_i$ and $\mathrm{Img}\,\alpha \subset P_i$ for $t \in [S,T]$. Denote $R = \alpha(S), S = \alpha(T)$. See the figure below.

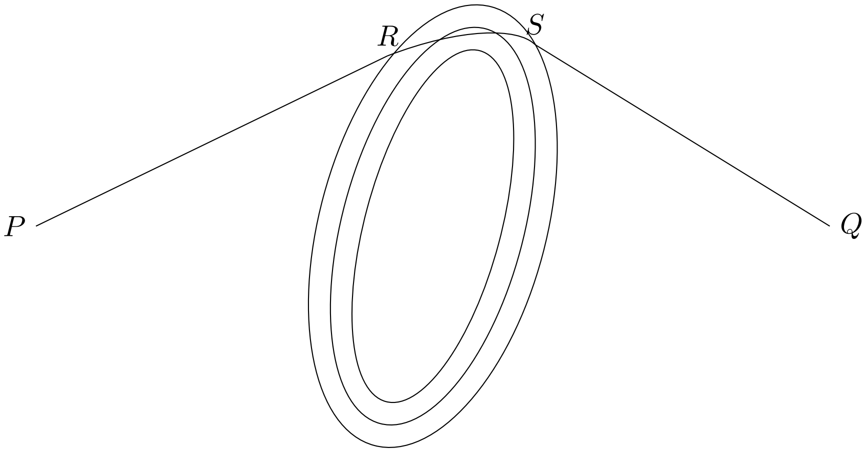


FIGURE 7

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.

Proposition 6. *$\tilde{\alpha}(t)$ 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.

Proof. If not, for any $t \in [0,1]$, let

$$I(t) = \{\, s \in [0,1] \mid (\tilde{\alpha}(s) - \tilde{\beta}(t)) \cdot \tilde{\beta}'(t) = 0 \,\}$$

and $t_0 = \inf I(t), t_1 = \sup I(t)$. Since $\tilde{\alpha}(1) = S$ is not on the normal line, then $t_1 < 1$. Moreover, it is easy to see that $t_0, t_1 \in I(t)$ by the continuity of $(\tilde{\alpha}(s) - \tilde{\beta}(t)) \cdot \tilde{\beta}'(t)$. Now $\tilde{\alpha}(s), s \in [t_0, t_1]$ must be entirely on the normal line otherwise it would not be the geodesic. Now we can choose Δt so small such that $\overline{\tilde{\alpha}(t_0)\tilde{\alpha}(t_1 + \Delta t)}$ is in the band. But $\overline{\tilde{\alpha}(t_0)\tilde{\alpha}(t_1 + \Delta t)}$ has smaller length, which is a contradiction!. □

Proposition 7.

$$\mathbb{L}_{\text{old}}(\tilde{\alpha}(t)) \geq (1 - 4\epsilon\kappa)\,\mathbb{L}_{\text{old}}(\tilde{\beta}(t)) \tag{21}$$

where κ is the maximal curvature.

Proof. Proposition 6 implies

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

and

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

for sufficiently small ϵ. Consequently,

$$\mathbb{L}_{\text{old}}(\tilde{\alpha}(t)) \geq \mathbb{L}_{\text{old}}(\tilde{\beta}(t)) - 4\epsilon\kappa\,\mathbb{L}_{\text{old}}(\tilde{\beta}(t)).$$ □

Proposition 8. *Suppose $\tilde{\alpha}_1(t), t \in [0,1]$ and $\tilde{\alpha}_2(t), t \in [0,1]$ are arc length reparametrizations of two segments of α inside the same obstacle, and $\tilde{\beta}_1$, $\tilde{\beta}_2$ are two curves defined as above, i.e.*

$$\tilde{\alpha}_1(t) = \tilde{\beta}_1(t) + \epsilon_1(t)\mathbf{n}_1(t),$$
$$\tilde{\alpha}_2(t) = \tilde{\beta}_2(t) + \epsilon_2(t)\mathbf{n}_2(t),$$

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

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

where $\epsilon_1 = 4\epsilon\kappa$

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

$$\mathbb{L}_{\mathrm{old}}(\tilde{\alpha}_1(t)) \geq \mathbb{L}_{\mathrm{old}}(\tilde{\beta}_1(t)) - 4\epsilon\kappa\,\mathbb{L}_{\mathrm{old}}(\tilde{\beta}_1(t))$$
$$\mathbb{L}_{\mathrm{old}}(\tilde{\alpha}_2(t)) \geq \mathbb{L}_{\mathrm{old}}(\tilde{\beta}_2(t)) - 4\epsilon\kappa\,\mathbb{L}_{\mathrm{old}}(\tilde{\beta}_2(t)).$$

By the definition of α, we have

$$\mathbb{L}_{\mathrm{old}}(\tilde{\beta}_2) - \mathbb{L}_{\mathrm{old}}(\tilde{\beta}_1) > (\mathbb{L}_{\mathrm{old}}(\tilde{\beta}_2) + \mathbb{L}_{\mathrm{old}}(\tilde{\beta}_1))(1-\epsilon_1)$$

Therefore,

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

Proposition 9. *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.*

$$\gamma_{opt}(t) = \mathrm{argmin}_{\gamma \in F}\,\mathbb{L}_{\mathrm{old}}(\gamma).$$

Then There exists a constant C such that

$$|\mathbb{L}_{\mathrm{old}}(\gamma_{opt}) - \mathbb{L}_{\mathrm{old}}(\beta)| \leq 2C\epsilon\kappa L.$$

Proof. Summing the inequality in (21), we get

$$\mathbb{L}_{\mathrm{old}}(\alpha) \geq \mathbb{L}_{\mathrm{old}}(\beta) - 4C\epsilon\kappa L,$$

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

$$\mathbb{L}_{\mathrm{old}}(\beta) \geq \mathbb{L}_{\mathrm{old}}(\gamma_{opt}) = \mathbb{L}_{\mathrm{new}}(\gamma_{opt}) \geq \mathbb{L}_{\mathrm{new}}(\alpha) \geq \mathbb{L}_{\mathrm{old}}(\alpha) \geq \mathbb{L}_{\mathrm{old}}(\beta) - 4C\epsilon\kappa L. \qquad \square$$

Proof of the structure theorem

From the discussion above, $\mathbb{L}_{\mathrm{old}}(\beta)$ and $\mathbb{L}_{\mathrm{old}}(\gamma_{opt})$ can be arbitrarily close. Hence we only need to minimize $\mathbb{L}_{\mathrm{old}}(\beta)$. By the construction of β, there are intervals $\{I_k\}_{k\in\Lambda_1}$ and $\{J_k\}_{k\in\Lambda_2}$ such that $\beta(t)|_{t\in I_k}$ is line segment and $\beta(t)|_{t\in J_k}$ is on the boundary of one obstacle P. Therefore, there exist an interval B_k such that $X(s)|_{s\in B_k}$ represents the same curve. Here X is the arc-length parametrization of the boundary of P. Let $\{t_k\}$ be the set of ending points of all B_ks. For any k, denote X_k the arc-length parametrization of the boundary of the obstacle that t_k belongs to and t_k^c, t_k^s the parameter that t_k is connected to by boundary and line segment respectively, then the length of β can be written as

$$2\mathcal{L}(\beta) = \sum_k \|X_k(t_k) - X_{k_s}(t_k^s)\| + \min(|t_k - t_k^c|, 1 - |t_k - t_k^c|).$$

It is easy to see that for each k, there are only two terms containing t_k, i.e.

$$\mathcal{L}(t_k) = \|X_k(t_k) - X_{k_s}(t_k^s)\| + \min(|t_k - t_k^c|, 1 - |t_k - t_k^c|).$$

For any k, t_k minimizes $\mathcal{L}(t_k)$ only if

$$\frac{d\mathcal{L}(t_k)}{dt_k} = \frac{(X_k(t_k) - X_{k_s}(t_k^s)) \cdot X_k'(t_k)}{\|X_k(t_k) - X_{k_s}(t_k^s)\|} + \text{sign}(\frac{1}{2} - |t_k - t_k^c|) = 0$$

But

$$\|(X_k(t_k) - X_{k_s}(t_k^s)) \cdot X_k'(t_k)\| \leq \|X_k(t_k) - X_{k_s}(t_k^s)\|$$

which implies that $X_k(t_k) - X_{k_s}(t_k^s)$ is parallel to $X_k'(t_k)$, i.e. $X_k(t_k) - X_{k_s}(t_k^s)$ is the tangent to P_k. Therefore, if β minimizes $\mathcal{L}$, then all the straight part of β should be tangent to a obstacle. By the inequality above, we know that this is also the solution to the original problem. □

References

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

[2] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *Journal of Optimization Theory and Applications*, 47(1):1–16, 1985.

[3] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Birkhauser, 2008.

[4] D. Burago, Y. Burago, S. Ivanov, and American Mathematical Society. *A course in metric geometry*. American Mathematical Society Providence, 2001.

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

[6] D. Z. Chen. On the all-pairs euclidean short path problem. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 292–301. Society for Industrial and Applied Mathematics, 1995.

[7] D. Z. Chen, G. Das, and M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. In *In Proc. 8th Canad. Conf. Comput. Geom.* Citeseer, 1996.

[8] S.-N. Chow and J. K. Hale. *Methods of bifurcation theory*, volume 138. Springer-Verlag, 1982.

[9] S.-N. Chow, T.-S. Yang, and H. Zhou. Global Optimizations by Intermittent Diffusion. *National Science Council Tunghai University Endowment Fund for Academic Advancement Mathematics Research Promotion Center*, page 121, 2009.

[10] K. Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 56–65. ACM, 1987.

[11] M.P. do Carmo. *Riemannian geometry*. Birkhauser, 1992.

[12] R. Goroshin, Q. Huynh, and H. Zhou. Approximate solutions to several visibility optimization problems. *Communications in Mathematical Sciences*, 9(2):535–550, 2011.

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

[14] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, 95(15):8431, 1998.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671, 1983.

[16] S. M. LaValle. *Planning algorithms*. Cambridge Univ Pr, 2006.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087, 1953.

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

[19] J. S. B. Mitchell. Shortest path and networks. *Handbook of Discrete and computational geometry*, pages 755–778, 1997.

[20] J. S. B. Mitchell and M. Sharir. New results on shortest paths in three dimensions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 124–133. ACM, 2004.

[21] S. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces.* Springer Verlag, 2003.

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

[23] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591, 1996.

[24] J. A. Sethian. *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*, volume 1999. Cambridge University Press Cambridge:, 1996.

[25] H. Szu and R. Hartley. Fast simulated annealing* 1. *Physics letters A*, 122(3-4):157–162, 1987.

[26] Y. H. R. Tsai, L. T. Cheng, S. Osher, P. Burchard, and G. Sapiro. Visibility and its dynamics in a PDE based implicit framework* 1. *Jo issn=0178-4617*, 1992.

[27] Y .H. R. Tsai, L. T. Cheng, S. Osher, and H. K. Zhao. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM journal on numerical analysis*, pages 673–694, 2004.

[28] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of computation*, 74(250):603–628, 2005.

School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A., {chow, jlu39, hmzhou}@math.gatech.edu.