COMMUN. MATH. SCI. © 2016 International Press
Vol. 14, No. 4, pp. 1169–1180

FAST COMMUNICATION

A NEWTON-LIKE ALGORITHM FOR THE SHORTEST PATH BASED ON THE METHOD OF EVOLVING JUNCTIONS[*]

SHUI-NEE CHOW[†], WUCHEN LI[‡], AND HAOMIN ZHOU[§]

Abstract. We present a fast Newton-like algorithm within the framework of the method of evolving junctions (MEJ) to find the shortest path in a cluttered environment. We demonstrate that the new algorithm converges much faster than the existing methods via numerical examples.

Key words. Shortest path problem, Newton method, intermittent diffusions, method of evolving junctions.

AMS subject classifications. 49J15, 49M15, 60H10.

1. Introduction

Finding the shortest path while avoiding obstacles plays a vital role in many applications, such as robotics, computer graphics, and space exploration. Mathematically, the problem can be formulated as follows. Let (X,d) be a length space, such as $\mathbb{R}^2$ or $\mathbb{R}^3$, where d is the distance defined on X, and let $P_1,\cdots,P_N$ be N open subsets of X representing obstacles with boundaries $\{\partial P_k\}_{k=1}^N$. Given two points $x,y\in X_c=X\setminus\cup_{i=1}^N P_i$, we define the admissible set of paths connecting x and y to be the curves that have no intersection with all the interior of obstacles, i.e.

$$\mathcal{A}(x,y,X_c)=\{\gamma:[0,1]\to X\,|\,\gamma(0)=x,\gamma(1)=y,\gamma\in X_c\},$$

where γ is absolutely continuous. For each admissible path, its length in Euclidean space is

$$J(\gamma(\theta))=\int_0^1|\dot\gamma(\theta)|\mathrm{d}\theta.$$

Then, finding the shortest path can be posed as an optimization problem:

$$\gamma^*=\operatorname{argmin}_{\gamma\in\mathcal{A}(x,y,X_c)}J(\gamma). \tag{1.1}$$

If the obstacles are polygons, the problem can be converted to an optimization problem on a graph, because the optimal path is a union of line segments connecting selected vertices. In this case, the combinatorial methods, such as the well known Dijkstra method, can be applied (see [1, 2, 3, 6, 8, 9, 11]). However, if the obstacles contain non-polygons, such as ones with smooth curved boundaries, the combinatorial methods cannot be applicable. In this case, most methods are based on differential

[*]Received: March 26, 2015; accepted (in revised form): July 3, 2015. Communicated by Jack Xin. This work is partially supported by NSF Awards DMS–1042998, DMS–1419027, and ONR Award N000141310408.

[†]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA(chow@gatech.edu).

[‡]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA (wli83@gatech.edu).

[§]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA (hmzhou@gatech.edu).

equations, such as the Pontryagin maximal principle and the Hamilton–Jacobi equations (see [7, 13, 14, 15, 16, 17, 18, 19]). However, for many problems, especially the ones with a large number of obstacles, these methods may become inefficient.

To overcome computational challenges, a new strategy called the method of evolving junctions (MEJ) has been proposed recently [5]. Compared to the existing methods for non-polygonal obstacles, it has two distinct properties: (1) It finds the shortest path by solving an initial value problem for a finite-dimensional system of ordinary differential equations (ODEs). This is different from the existing methods, which solve infinite-dimensional problems. (2) MEJ uses intermittent diffusion, a stochastic differential equation (SDE) approach for the global optimization, to find the global solution. The existing methods based on the Pontryagin principle can only find local minima, and those based on Hamilton–Jacobi equations find the global solution with (often) prohibitive computation requirements in practice.

In this paper, we further improve the MEJ presented in [5, 10]. In particular, we focus on the shortest path problem in $\mathbb{R}^2$. In short, we use an approximated Newton method to replace the gradient flow in MEJ while retaining the overall MEJ framework, including the SDEs, to help the solution jump out of the traps of local minimizers. Such a replacement significantly reduces the computational time in finding the local minimizers. In addition, the new method provides a fresh viewpoint for the shortest path by posing it as an angle minimization problem, which has not been reported in the past.

This paper is arranged as follows. In Section 2, we briefly review the MEJ proposed in [5]. In Section 3, we state the main contribution of this paper and present the derivation of the approximated Newton method. In Section 4, we give numerical examples to show the efficiency of our method. A short discussion is added in the end of the paper.

2. Method of evolving junctions

We start with a geometric structure, called separable, possessed by all shortest paths.

Definition 2.1. *A path $\gamma\colon [0,1]\to X_c$ is* separable *if there exists a finite number of points $\{x_1,x_2,\cdots,x_n\}$ with $x_i\in\partial P_{k_i}$, $k_i\le N$, such that γ concatenates line segments and partial curves on the boundaries of the obstacles, i.e.*

$$\gamma=\gamma_0(x,x_1)\cdot\gamma_c(x_1,x_2)\cdot\gamma_0(x_2,x_3)\cdot\gamma_c(x_3,x_4)\cdots\gamma_0(x_n,y), \tag{2.1}$$

where $\gamma_0(x_{i-1},x_i)$ is the line segment connecting x_{i-1} and x_i and $\gamma_c(x_{i-1},x_i)$ is the geodesic on the boundary ∂P_{k_i} between the two points.

A simple example is shown in Figure 2.1, and we call x_i a junction.

Theorem 2.2. *Let ∂P_k be a finite combination of convex and concave curves (surfaces). Then, γ^* is separable. Moreover, each line segment $\overline{x_{i-1}x_i}$ is tangent to the obstacle ∂P_{k_i}.*

Therefore, the length of the shortest path is a function depending on the junctions, i.e. $\{x_1,\dots,x_n\}$,

$$J(x_1,\dots,x_n)=\sum_{i=1}^{n}J(x_{i-1},x_i),$$

where $J(x_{i-1},x_i)$ represents the distance connecting (x_{i-1},x_i):

$$J(x_{i-1},x_i)=\begin{cases}\|x_{i-1}-x_i\|, & \text{if i is odd;}\\ \mathrm{dist}_c(x_{i-1},x_i), & \text{if i is even,}\end{cases}$$

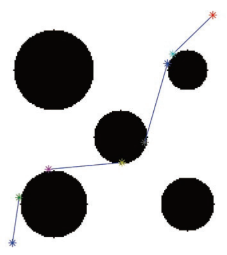


Fig. 2.1: *Each connecting point between a line segment and a boundary of the obstacles is a junction.*

in which $\|\cdot\|$ is the Euclidean norm.

Based on this theorem, MEJ restricts the search space to the set of all admissible paths with separable structures, a finite-dimensional subset of $\mathcal{A}(x,y,X_c)$. More precisely, MEJ finds the shortest path by solving the following optimization problem.

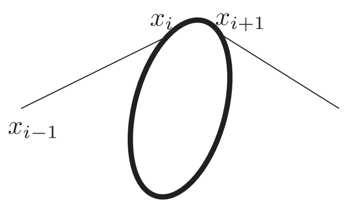


Fig. 2.2: *Each junction on a boundary is connected to the points before and after it by a straight line segment and an arc of the boundary.*

$$\min_{x_1,\cdots,x_n} J(x_1,\cdots,x_n). \tag{2.2}$$

Remark 2.1. The dimension of variables, the number of junctions, can be changed during the searching. This is very different from the existing methods.

To find the global solution of (2.2), MEJ uses intermittent diffusion (ID), an SDE-based global-optimization method developed in [4]. More precisely, it solves

$$d\hat{x} = -\nabla J(\hat{x})dt + \sigma(t)dW_t, \tag{2.3}$$

where $\hat{x} = \{x_1,\cdots,x_n\}$ represents the junctions, W_t the standard Brownian motion in $\mathbb{R}^N$, and $\sigma(t)$ a piecewise constant function

$$\sigma(t) = \sum_{j=1}^{m} \sigma_j \chi_{[S_j,T_j]}(t), \tag{2.4}$$

with $0 = S_1 < T_1 < \cdots < S_m < T_m < S_{m+1} = T$ and $\chi_{[S_j,T_j]}$ being the characteristic function of interval $[S_j,T_j]$.

If $\sigma(t) = 0$, equation (2.3) becomes a gradient-descent flow which converges to a local minimizer; if $\sigma(t) > 0$, the path has a certain (positive) probability, controlled by $\sigma(t)$, to jump out of the local traps and therefore to reach the global solution.

3. The Newton-like algorithm

In this section, we present a new approximated Newton method on the shortest path problem in $\mathbb{R}^2$ by finding the line segments tangent to the obstacles directly to replace the gradient flow in (2.3) when $\sigma(t)=0$.

In order to explain our method more clearly, we introduce an arc-length parameter θ to represent junctions. Let $x_i=x(\theta_i)$, $x_i^s=x(\theta_i^s)$, $x_i^c=x(\theta_i^c)$, where θ_i, θ_i^s, θ_i^c are arc-length parameters on the corresponding boundaries, super index s indicates the junction connected to x_i by a straight line, and c denotes the junction connected to x_i by a boundary arc (see Figure 3.1 for an illustration). With these notations, the length of the curve containing one straight line segment and the boundary arc $\gamma_0(x_i^s,x_i)\cdot\gamma_c(x_i,x_i^c)$ becomes

$$J_i(\theta)=\|x(\theta_i)-x(\theta_i^s)\|+d(\theta_i,\theta_i^c),$$

where $d(\theta_i,\theta_i^c)=\min\{d^+(\theta_i,\theta_i^c),d^-(\theta_i,\theta_i^c)\}$, with d^+, d^- representing the counter-clockwise and clockwise distance on the obstacle boundary between $x(\theta_i^c)$ and $x(\theta_i)$, respectively, as illustrated in Figure 3.1.

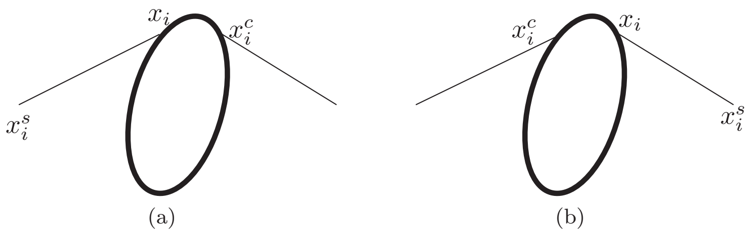


Fig. 3.1: *Two different scenarios for each junction on a boundary that is connected to the points before and after it by a straight line segment and an arc of the boundary.*

Thus, the optimization problem (2.2) becomes

$$\min_{\theta_1,\cdots,\theta_n} J(\theta)=\frac{1}{2}\sum_{i=1}^{n} J_i(\theta),$$

where $\theta=(\theta_1,\cdots,\theta_n)$, and the intermittent diffusion (2.3) is

$$d\theta=-\nabla J(\theta)dt+\sigma(t)dW_t, \tag{3.1}$$

in which we have

$$\frac{\partial J}{\partial\theta_i}=\frac{x(\theta_i)-x(\theta_i^s)}{\|x(\theta_i)-x(\theta_i^s)\|}\cdot\dot{x}(\theta_i)+\mathrm{sign}(d^+(\theta_i,\theta_i^c)-d^-(\theta_i,\theta_i^c)),$$

where $\dot{x}(\theta_i)=\frac{dx(\theta_i)}{d\theta_i}$.

The main idea of this paper is that, instead of using the gradient flow to find the local minimizer of (2.2), we apply the Newton method to solve $\nabla J(\theta)=0$ directly. This is equivalent to solving the tangent condition in Theorem 2.2, as stated in the next theorem. Here, we denote $J_i^{(k)}(\theta)=\frac{\partial^k J_i}{\partial\theta_i^k}(\theta)$, $k=2,3$.

Theorem 3.1. *If θ^* is the local minimizer of* (2.2), *then the following statements are equivalent:*

(i) The line segment is tangent to the obstacle;

(ii) The second-order derivative $J_i^{(2)}(\theta^*)=0$ for $i=1,\ldots,n$.

Moreover, the third-order derivative satisfies

$$|J_i^{(3)}(\theta^*)|=|\kappa(\theta_i^*)|^2$$

and

$$\frac{\partial J_i^{(2)}}{\partial\theta_j}(\theta^*)=0$$

for any $i,j=1,\cdots,n$ and $i\neq j$.

Proof. First, we show that solving $\nabla J(\theta)=0$ implies that the line connecting the junctions is tangent to the obstacles, since

$$\frac{\partial J}{\partial\theta_i}=g(\theta_i,\theta_i^s)+\operatorname{sign}(d^+(\theta_i,\theta_i^c)-d^-(\theta_i,\theta_i^c))=0, \tag{3.2}$$

where

$$g(\theta_i,\theta_i^s)=\frac{x(\theta_i)-x(\theta_i^s)}{\|x(\theta_i)-x(\theta_i^s)\|}\cdot\dot{x}(\theta_i). \tag{3.3}$$

Hence $\nabla J(\theta)=0$ is to solve $g(\theta_i,\theta_i^s)^2=1$ for each i. Moreover, since θ is the arc-length parameter, $\dot{x}(\theta_i)$ is a unit vector, which implies that $g(\theta_i,\theta_i^s)$ is the inner product of two unit vectors. Then, $g(\theta_i,\theta_i^s)^2=1$ means $x(\theta_i)-x(\theta_i^s)$ is parallel to tangent vector $\dot{x}(\theta_i)$, which implies the tangent property.

To show the equivalence of (i) and (ii), we need to prove that (i) implies (ii). Without loss of generality, let us assume $\operatorname{sign}(d^+(\theta_i,\theta_i^c)-d^-(\theta_i,\theta_i^c))=-1$. Since $g(\theta_i,\theta_i^s)\leq 1$, solving (3.2) is equivalent to finding the maximizer θ^* of

$$\max_{\theta_i,\theta_i^s} g(\theta_i,\theta_i^s).$$

Then, it must satisfy

$$g_{\theta_i}(\theta_i^*,\theta_i^{s*})=0.$$

Since $J_i^{(2)}(\theta)=g_{\theta_i}(\theta_i,\theta_i^s)$, we have $J_i^{(2)}(\theta^*)=0$.

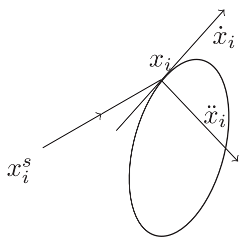
(ii) implies (i): A direct computation gives the second-order derivative of J:

$$J_i^{(2)}(\theta)=\frac{1-g(\theta_i,\theta_i^s)^2+(x(\theta_i)-x(\theta_i^s))\cdot\ddot{x}(\theta_i)}{\|x(\theta_i)-x(\theta_i^s)\|}. \tag{3.4}$$

If $J_i^{(2)}(\theta^*)=0$ for all $i=1,\ldots,n$, we have

$$1-g(\theta_i^*,\theta_i^{s*})^2+(x(\theta_i^*)-x(\theta_i^{s*}))\cdot\ddot{x}(\theta_i^*)=0.$$

Notice that $1-g(\theta_i^*,\theta_i^{s*})^2\geq 0$. Moreover, since $\overline{x(\theta_i^*)x(\theta_i^{s*})}$ first intersects obstacle P_{k_i} at point $x(\theta_i^*)$, which implies angle between vector $x(\theta_i^*)-x(\theta_i^{s*})$, and since $\ddot{x}(\theta_i^*)$ is not larger than $\frac{\pi}{2}$, $(x(\theta_i^*)-x(\theta_i^{s*}))\cdot\ddot{x}(\theta_i^*)\geq 0$. Hence, the solution satisfies $g(\theta_i^*,\theta_i^s)^2=1$, which implies the tangent property.



The third derivative is

$$\begin{aligned}J_i^{(3)}(\theta) = &\left(\frac{1}{\|x(\theta_i)-x(\theta_i^s)\|}\right)^{(1)} \cdot J_i^{(2)}(\theta)\\ &+\frac{1}{\|x(\theta_i)-x(\theta_i^s)\|}\cdot[2\dot{x}(\theta_i)\cdot\ddot{x}(\theta_i)-2g(\theta_i)\cdot J_i^{(2)}(\theta)\\ &+(x(\theta_i)-x(\theta_i^s))\cdot\dddot{x}(\theta_i)].\end{aligned}$$

Considering that θ_i is an arc-length parameter, we have $\dot{x}(\theta_i)\cdot\ddot{x}(\theta_i)=0$. Combining it with $J_i^{(2)}(\theta^*)=0$, we can show

$$J_i^{(3)}(\theta^*) = \frac{(x(\theta_i^*)-x(\theta_i^{*s}))\cdot\dddot{x}(\theta_i^*)}{\|x(\theta_i^*)-x(\theta_i^{*s})\|}.$$

By the tangent property, (3) can be formulated as

$$|J_i^{(3)}(\theta^*)| = |\dot{x}(\theta_i^*)\cdot\dddot{x}(\theta_i^*)|.$$

Since

$$\begin{aligned}F(\theta_i) = &\int_0^{\theta_i}\dot{x}(u)\dddot{x}(u)du\\ = &\dot{x}(0)\ddot{x}(0)-\int_0^{\theta_i}\ddot{x}(u)^2du\end{aligned}$$

and $|\kappa(u)|=|\ddot{x}(u)|$,

$$\dot{x}(\theta_i)\cdot\dddot{x}(\theta_i) = \frac{dF(\theta_i)}{d\theta_i} = -\kappa^2(\theta_i),$$

which implies $|J_i^{(3)}(\theta^*)|=\kappa^2(\theta_i^*)$.

In the end, we show that $\frac{\partial J_i^{(2)}}{\partial\theta_j}(\theta^*)=0$ for $j\neq i$. Since $J_i^{(2)}(\theta)$ depends on θ_i^s and θ_i, we only need to show $\frac{\partial J_i^{(2)}}{\partial\theta_i^s}(\theta^*)=0$. By direct computations, we have

$$\frac{\partial J_i^{(2)}}{\partial\theta_i^s}(\theta^*) = -\frac{2g_{\theta_i^s}(\theta_i^*,\theta_i^{s*})g(\theta_i^*,\theta_i^{s*})}{\|x(\theta_i^*)-x(\theta_i^{s*})\|}.$$

Since

$$g_{\theta_i^s}(\theta_i^*,\theta_i^{s*})=0,$$

$\frac{\partial J^{(2)}}{\partial\theta_i^s}(\theta^*)=0$, which finishes the proof. $\square$

Now, we are ready to present our method. We want to solve the tangent condition $\nabla J(\theta)=0$ directly through the Newton method. By Theorem 3.1, it can be found that $\nabla J(\theta)=0$ is a degenerate system, i.e. its Jacobian matrix becomes 0 at θ^*. Hence, we can solve the system

$$J^{(2)}(\theta)=(J_1^{(2)}(\theta),\cdots,J_n^{(2)}(\theta))=0$$

instead. We use an approximated Jacobian matrix of $J^{(2)}(\theta)$ given by

$$H(\theta)=\mathrm{diag}(J_i^{(3)}(\theta)),$$

and it leads to the following iterations:

$$\theta^{k+1}=\theta^k-H(\theta^k)^{-1}J^{(2)}(\theta^k). \tag{3.5}$$

We must point out that, in this iterative scheme, we consider obstacles with non-zero curvature boundaries. Hence, H is an invertible diagonal matrix and that the iteration can be carried out. On the other hand, when boundaries are straight lines or curves with curvature close to 0, we can simply adjust $H(\theta)$, such as letting $H(\theta)=\mathrm{diag}(J_i^{(3)}(\theta)+\lambda_i I)$, with λ_i being a selected scalar, to continue the iteration.

This iteration is an "approximated" Newton method. Only at the minimizer θ^* is the Jacobian matrix of $J^{(2)}(\theta)$ exactly as $H(\theta)$. Otherwise, $H(\theta)$ is an approximation to the Jacobian. We use this formulation because its computation complexity is as low as the gradient-descent algorithm. However, its convergence is superlinear, which is faster than the gradient-descent algorithm.

THEOREM 3.2. *Let $J^{(2)}(\theta)\colon \mathbb{R}^N\to\mathbb{R}^N$ be smooth, with no zero-curvature points for the boundaries of all obstacles in $\mathbb{R}^2$. Then, there exists $\epsilon>0$ such that, if the iteration* (3.5) *starts at $\|\theta^0-\theta^*\|<\epsilon$, θ^k converges to θ^* superlinearly.*

Proof. The proof of the theorem follows the standard procedure for Newton-like methods, which is often divided into two steps. Firstly, we use the fixed-point theorem to show that there exists a sufficiently small ϵ such that, if $\|\theta^0-\theta^*\|<\epsilon$, θ^k converges to θ^*. In other words, let us consider a map $l:\mathbb{R}^N\to\mathbb{R}^N$, i.e.

$$l(\theta)=\theta-H(\theta)^{-1}J^{(2)}(\theta).$$

We want to find a small neighbor of θ^*, $B(\theta^*,\epsilon)=\{\theta\,|\,\|\theta-\theta^*\|\le\epsilon\}$, such that $\sup_{\theta\in B(\theta^*,\epsilon)}\|\mathbf{D}l(\theta)\|<1$, where $\mathbf{D}$ is a Jacobian operator. To show this, by using the fact that $H(\theta)=\mathrm{diag}(h_i(\theta))$ is diagonal, we directly calculate

$$\mathbf{D}l(\theta)=I-H(\theta)^{-1}\mathbf{D}J^{(2)}(\theta)+M,$$

where M is an $n\times n$ matrix with $M_{ij}=\frac{\partial}{\partial\theta_j}(\frac{1}{h_i(\theta)})J_i^{(2)}(\theta)$. Substitute $H(\theta^*)=\mathbf{D}J^{(2)}(\theta^*)$ and $J^{(2)}(\theta^*)=0$ into the above equation, and we have $\mathbf{D}l(\theta^*)=0$. By the continuity of $\mathbf{D}l(\theta)$, there exists $\epsilon>0$ such that $\sup_{\theta\in B(\theta^*,\epsilon)}\|\mathbf{D}l(\theta)\|<1$. Hence θ^k is convergent.

Secondly, we show that the convergence rate is superlinear. To show this, let $e_k=\theta^*-\theta^k$, we need to show $\lim_{k\to\infty}\|e_{k+1}\|/\|e_k\|=0$. Since θ_k converges to θ^*, we only consider the bounded region $B(\theta^*,\epsilon)$. On one hand, by the Taylor expansion of $J^{(2)}(\theta)$,

$$0=J^{(2)}(\theta^*)=J^{(2)}(\theta^k+e_k)=J^{(2)}(\theta^k)+\mathbf{D}J^{(2)}(\theta^k)e_k+O(\|e_k\|^2).$$

Hence

$$e_k + \mathbf{D}J^{(2)}(\theta^k)^{-1}J^{(2)}(\theta^k) = O(\|e_k\|^2). \tag{3.6}$$

On the other hand, substitute e_{k+1}, e_k and (3.6) into equation (3.5):

$$\begin{aligned} e_{k+1} =& \theta^* - \theta^{k+1} = \theta^* - [\theta^k - H(\theta^k)^{-1}J^{(2)}(\theta^k)] \\ =& e_k + \mathbf{D}J^{(2)}(\theta^k)^{-1}J^{(2)}(\theta^k) + [H(\theta^k)^{-1} - \mathbf{D}J^{(2)}(\theta^k)^{-1}]J^{(2)}(\theta^k) \\ =& O(\|e_k\|^2) + [H(\theta^k)^{-1} - \mathbf{D}J^{(2)}(\theta^k)^{-1}]J^{(2)}(\theta^k). \end{aligned} \tag{3.7}$$

We need to consider $\|[H(\theta^k)^{-1} - \mathbf{D}J^{(2)}(\theta^k)^{-1}]J^{(2)}(\theta^k)\|$ in terms of e_k. Here, we notice that $\mathbf{D}J^{(2)}(\theta^*)$ is invertible and $J^{(2)}(\theta)$ is smooth. Then by the inverse function theorem, $\mathbf{D}J^{(2)}(\theta)^{-1}$ exists and is a smooth function when $\theta \in B(\theta^*, \epsilon)$. Moreover, we apply the Taylor expansion to the function $\|H(\theta)^{-1} - \mathbf{D}J^{(2)}(\theta)^{-1}\|$:

$$\begin{aligned} \|H(\theta^k)^{-1} - \mathbf{D}J^{(2)}(\theta^k)^{-1}\| \le& \|H(\theta^*)^{-1} - \mathbf{D}J^{(2)}(\theta^*)^{-1}\| + C\|\theta^k - \theta^*\| \\ =& C\|\theta^k - \theta^*\| = C\|e_k\|, \end{aligned}$$

where

$$C = \sup_{\theta \in B(\theta^*, \epsilon)} \|\mathbf{D}(\|H(\theta)^{-1} - \mathbf{D}J^{(2)}(\theta^*)^{-1}\|)\|.$$

Combining all results into (3.7), we obtain

$$\begin{aligned} \|e_{k+1}\| \le& \|H(\theta^k)^{-1} - \mathbf{D}J^{(2)}(\theta^k)^{-1}\| \|J^{(2)}(\theta^k)\| + O(\|e_k\|^2) \\ \le& C\|e_k\| \|J^{(2)}(\theta^k)\| + O(\|e_k^2\|). \end{aligned}$$

Hence

$$\frac{\|e_{k+1}\|}{\|e_k\|} \le C\|J^{(2)}(\theta^k)\| + O(\|e_k\|).$$

Since θ^k converges to θ^*, we have $\lim_{k\to\infty} \|J^{(2)}(\theta^k)\| = \|J^{(2)}(\theta^*)\| = 0$ as well as $\lim_{k\to\infty} \|e_k\| = 0$. This implies $\lim_{k\to\infty} \|e_{k+1}\|/\|e_k\| = 0$, indicating that θ^k converges to θ^* superlinearly. ∎

3.1. Algorithm. With all the components discussed above, we are ready to state our algorithm.

Method of Evolving Junctions

Input: Number of intermittent diffusion intervals m.

Output: The optimal set γ^* for the junctions.

```
1.  Initialization. Find the initial path $\gamma^{(0)} = (\theta_1, \cdots, \theta_n)$;
2.  Select the duration of diffusion $\Delta T_l$, $l \le m$;
3.  Select diffusion coefficients $\sigma_l$, $l \le m$;
4.  for $l = 1:m$
5.      $\gamma^{(l)} = \gamma^{(0)}$;
6.      for $j = 1:\Delta T_l$
7.          Find $\nabla J(\gamma^{(l)})$.
```

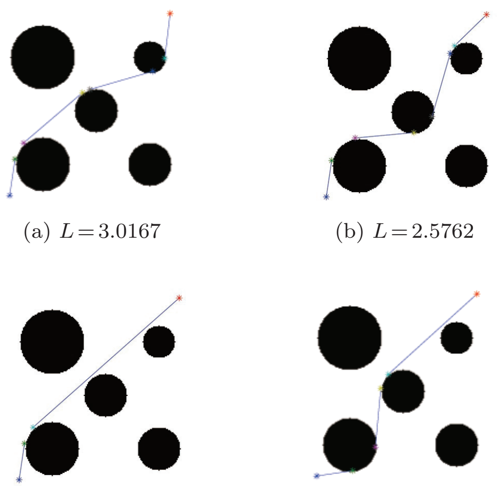
```
8.              Update $\gamma^{(l)}$ according to (3.1) with $\sigma(t)=\sigma_l$;
9.              Remove junctions from or add junctions to $\gamma^{(l)}$ when necessary;
10.         end
11.         while $\|\nabla J(\gamma^{(l)})\|>\epsilon$
12.             Update $\gamma^{(l)}$ according to (3.5) with $\sigma(t)=0$;
13.         end
14.     end
15.     Compare $J(\gamma^{(l)})$, $l\le m$ and set $\gamma_{opt}=\text{argmin}_{l\le m} J(\gamma^{(l)})$;
```

In our implementation, we choose the parametrization direction, either clockwise or counterclockwise, according to the initial condition θ^0. For instance, if $\text{sign}(d^+(\theta_i^0,\theta_i^{0c})-d^-(\theta_i^0,\theta_i^{0c}))=1$, we parametrize P_{k_i} clockwise.

4. Numerical experiment

In this section, we use two numerical examples to show the effectiveness of the new algorithm.

Example 1: In this case, the obstacles are 5 disks with centers (1,1), (1.5,1.5), (0.5,0.5), (1.5,0.5), (0.5,1.5) and radii 0.2, 0.2, 0.3, 0.25, 0.15, respectively. The starting and ending points are $X=(1.8,0.2)$, $Y=(0.1,1.7)$. We use $m=20$ for ID defined in formula (2.4). Figure 4.1 shows the four shortest paths found by the algorithm. They are all local minimizers except the global one shown in Figure 4.1(C).



(a) $L=3.0167$ (b) $L=2.5762$

(c) $L=2.3575$ (d) $L=2.5045$

Fig. 4.1: *The algorithm finds 4 different shortest path, where (C) is the global minimizer.*

It is worth pointing out that, by using a 2013 Macbook Air with CPU core i5, 1.8G HZ, RAM 4GB, our method needs only 1.75 seconds to finish one simulation, while the method in [5] spends 485.777 seconds. Here, both methods obtain 10 minimizers. On average, it takes 0.175 seconds for our method to reach a minimizer, and the gradient-descent method needs 48.58 seconds to do it. This indicates that the computation time is reduced by more than 200 times. A further look at the experiments finds that it often takes about 10 iterations for our Newton-like method to converge to a minimizer, which is much fewer steps, with a similar complexity for each step, than that of the gradient-descent method. This agrees with the superlinear convergence proved in Theorem 3.2. On the other hand, our algorithm still enjoys the advantages of the ID strategy that allows us to find the global minimizer. Moreover, our experiment shows that the larger the value of m, the larger the probability of obtaining a global minimizer. To demonstrate this, we compute 10 independent simulations with $m=20$ for each simulation. We find the global minimizer plotted in Figure 4.1(C) in 5 out of the 10 simulations. If we take $m=50$, we observe Figure 4.1(C) in 7 out of the 10 simulations. This indicates that the larger the value of m, the larger the probability of reaching one of the global minimizers.

Example 2: We consider the same environment as the one used in [5]: four obstacles with arbitrary shape, as shown in Figure 4.2. The starting point is $X=(0.5,0.002)$, and the ending point is $Y=(0.5,0.98)$. Different from the previous example, for which we have analytical parameterizations for the obstacles, here we compute the necessary quantities, such as the curvature and the principle norm, numerically by the level-set method [12]. To compare the result reported in [5], we take $m=2$ and focus on the computation time to obtain one minimizer. Our method needs 3.719 seconds, while the method in [5] takes 215.840 seconds. In fact, the minimizer is a global one.

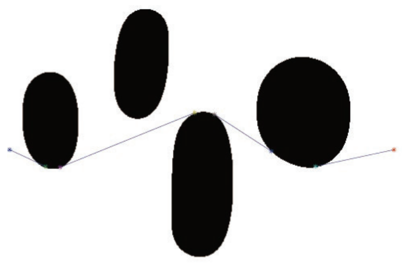


Fig. 4.2: *General obstacles.*

5. Further discussion

In general, the shortest path problem is considered as an infinite-dimensional constrained optimal control problem, since any feasible path is an absolute continuous function. Through the “separable” geometric structure possessed by the optimal path, we present an efficient method for finding the global optimal solution via a finite-dimensional system of differential equations. Moreover, the Newton-like algorithm provides a different but interesting viewpoint for the shortest path problem. Instead of looking at the length function formulated by all junctions as we did in this paper, we can study it from the tangent property between two adjoint junctions directly. To be more precise, the shortest path problem in $\mathbb{R}^2$ can be re-formulated so as to minimize the angle u_i between line $\overline{x_i^s x_i}$ and the tangent of the obstacle at x_i, as shown in Figure 5.1. If we use the arc-length parametrization as we introduced in Section 3, finding the junctions for the shortest path is equivalent to the following angle-optimization problem:

$$\min_{\theta_i,\theta_i^s} u_i = <\dot{x}(\theta_i), x(\theta_i^s)-x(\theta_i)> \quad i=1,\dots,n, \tag{5.1}$$

where $<,>$ represents the acute angle between vectors. Using the monotonicity of the

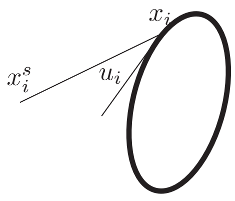


Fig. 5.1: *Representing junctions though angle u_i.*

sine function in $[0,\frac{\pi}{2}]$, (5.1) can be posed as

$$\min_{\theta_i,\theta_i^s} \sin^2 u_i = 1 - g(\theta_i,\theta_i^s)^2, \tag{5.2}$$

where g is the function defined in (3.3). Since the minimum of $\sin^2 u_i$ is 0, this is to say the minimizer θ^* satisfying

$$g(\theta_i,\theta_i^s) = \pm 1,$$

which is the same as (3.2) showing $\frac{\partial J}{\partial \theta_i} = 0$. In this sense, the equation $J^{(2)}(\theta) = 0$ solved by our Newton algorithm is the equation satisfied by the minimizer of (5.2). In other words, the above optimization (5.2) only involves a junction pair, and our method is a Newton method to solve such junction pairs directly. One may use other methods to solve the optimization problem as well.

The significance of this viewpoint is that the angle u_i only depends on the junction pair on the two obstacles, so this formulation only needs local, instead of global, information of the optimal path, which is different from our general understanding of the shortest path problem. This may provide new routes to construct more efficient computation methods, which is among the future research considerations.

The framework of MEJ can be extended to the shortest path problem in $\mathbb{R}^3$, as discussed in [5]. However, the proposed Newton-like method must undergo a significant change if one wants to apply to the 3-D case. The main challenges are listed below.

1 It takes two parameters to describe a junction on the surface in 3-D. This leads to a much more complex formula than Equation (7) for the derivative of J, which may no long be degenerate. So, the second part of Theorem 3 may not hold anymore. Without that, the formulations we used to design the algorithm are not applicable either.

2 The proposed Newton-like algorithm is heavily based on the fact that the second-order derivatives of J form a diagonal matrix. It is not clear whether this is true in 3-D.

With the aforementioned difficulties, one has to re-derive the Newton method, and this is among the tasks of our future investigations.

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, Society for Industrial and Applied Mathematics, 283–292, 2009.

[2] R.K. Ahuja, K. Mehlhorn, J. Orlin, and R.E. Tarjan, *Faster algorithms for the shortest path problem*, Journal of the ACM (JACM), 37(2), 213–223, 1990.

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

[4] S.-N. Chow, T.-S. Yang, and H. Zhou, *Global Optimizations by Intermittent Diffusion*, International Journal of Bifurcation and Chaos, 466–479, 2013.

[5] S.-N. Chow, J. Lu, and H. Zhou, *Fast numerical methods based on sdes for several problems related to the shortest path*, Meth. Appl. Anal., 20(4), 353–364, 2013.

[6] D.Z. Ghent, *On the all-pairs euclidean short path problem*, in Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms, SIAM, 76, 292, 1995.

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

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

[9] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.

[10] J. Lu, Y. Diaz-Mercado, M. Egerstedt, H. Zhou, and S.-N. Chow, *Shortest paths through 3-dimensional cluttered environments*, in Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, 6579–6585, 2014.

[11] J.S.B. Mitchell, *Shortest paths and networks*, in Handbook of Discrete and Computational Geometry, CRC Press, Inc., 445–466, 1997.

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

[13] 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.

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

[15] J.A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 1999, Cambridge University Press Cambridge, 1996.

[16] J.A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, 3, Cambridge University Press, 1999.

[17] Y.-H.R. Tsai, L.-T. Cheng, S. Osher, P. Burchard, and G. Sapiro, *Visibility and its dynamics in a PDE based implicit framework*, J. Comput. Phys., 199(1), 260–290, 2004.

[18] 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, 41(2), 673–694, 2003.

[19] H. Zhao, *A fast sweeping method for eikonal equations*, Mathematics of Computation, 74(250), 603–627, 2005.