Methods and Applications of Analysis.
Vol. 20, No. 4, pp. 353–364, December 2013

003

FAST NUMERICAL METHODS BASED ON SDES FOR SEVERAL PROBLEMS RELATED TO THE SHORTEST PATH*

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

Dedicated to Professor Stan Osher on the occasion of his 70th birthday

Abstract. In this paper, we apply the Evolving Junctions on Obstacle Boundaries (E-JOB) method for the shortest path problem developed in [3] and [4] to three related problems: 1) find the shortest path to move a disk, instead of a point, from one position to another one; 2) calculate the Euclidean distance between two sets in $\mathbf{R}^n$; and 3) compute the shortest path in an environment with obstacles appearing or disappearing at arbitrary times. The methods are based on solving finite dimensional stochastic differential equations with given initial values. And they are robust, efficient and easy to implement. We illustrate the methods by some numerical examples.

Key words. Path planning, shortest path, stochastic differential equations, global optimization, robotics.

AMS subject classifications. 65D18, 65K10, 65C30.

1. Introduction.

Finding the shortest path between two given points X and Y while avoiding all the obstacles $\{P_i\}_{i=1}^N$ in a region is a classical problem that carries different names, such as piano mover's problem, watchman route problem or zookeeper's problem in practice. It has received great attention since decades ago. Various methods based on combinatorics or differential equations have been developed for different scenarios. For polygonal obstacles in $\mathbf{R}^2$, an optimal algorithm based on Dijkstra method with complexity $O(n \log n)$, n being the number of vortices, was obtained by Hershberger and Suri [7]. However, the problem was shown to be NP-Hard in $\mathbf{R}^3$ [1]. For obstacles with smooth boundaries, a method of front propagation is often used. The idea is to propagate a wave front from the starting point X with unit speed. The moment that the front reaches Y is actually the length of the shortest path. This arrival time can be conveniently computed by solving an Eikonal equation using the fast marching [8] or fast sweeping [12] algorithms. The front propagation is in fact a continuous version of Dijkstra method. Its requirement of updating the whole domain is often undesirable in practice. For a survey, see [6, 9].

The method of Evolving Junctions on Obstacle Boundaries(E-JOB) is a novel and fast algorithm developed by the authors in a paper published by Applied and Computational Harmonic Analysis, Volume 35(1), page 165-176 [3] and [4] for the shortest path problem. Unlike the traditional methods using combinatorial or PDE approaches, E-JOB takes a completely different strategy. The key idea of E-JOB is based on a simple structure of the shortest path, i.e. a shortest path consists of only line segments and boundary arcs connected by junctions on the boundaries of obstacles. Figure 1 demonstrates the structure of a shortest path in which the small dots indicate the junctions. To find the shortest path, only those junctions need to be determined. In this way, E-JOB converts the original infinite dimensional problem into a finite dimensional problem of finding points on the boundaries to connect those

*Received January 26, 2013; accepted for publication October 30, 2013. This work is partially supported by NSF Faculty Early Career Development (CAREER) Award DMS-0645266 and DMS-1042998, and ONR Award N000141310408.

†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA ({chow; jlu39; hmzhou}@math.gatech.edu).

line segments and arcs. In other words, the length of the optimal path is merely

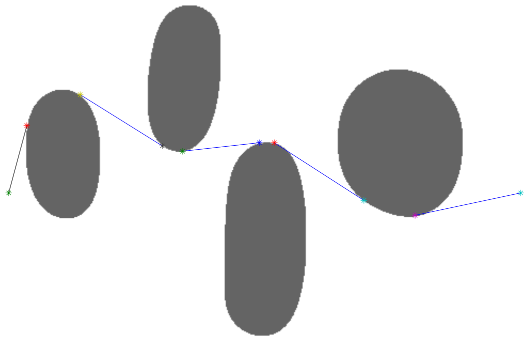


FIG. 1. *Structure of a shortest path*

a function of those junction points. This observation enables us to search for the shortest path in a much smaller set, namely a finite dimensional set K composed by all the paths that consist of line segments and arc connected by the junctions on the boundaries.

To determine the optimal positions of the junctions, one may use the simplest gradient descent method to decrease the length functional on K. However, gradient descent method alone only finds local optimal solutions. To overcome this challenge, E-JOB applies a recently developed global optimization strategy called Intermittent Diffusion (ID). The resulting algorithm is a stochastic differential equation with initial conditions, which can be computed by many existing numerical schemes. A notable feature of ID is that, unlike other global optimization methods such as Simulated Annealing and Metropolis algorithm, ID is able to obtain a series of local minimizers in each realization, and it has a large (arbitrarily approaching 1 with geometrical convergence rate if enough time is allowed) probability that the global optimal solution is among them. A benefit of this feature is that if only limited time is allowed, E-JOB can still find the best solution within the given time frame. For more details of ID, please see [2]. Another feature of E-JOB is that it uses the level set method [10] to handle complicated geometries of the obstacles. This makes transformations of obstacles straightforward and fast.

In this paper, we consider three different problems that can be solved by certain variations of E-JOB.

The first problem concerns the shortest path for moving a disk between two points. This is a more realistic model in robotics where the moving system, such as the Unmanned Vehicle Systems (UVS), is a rigid body free only to translation. The distance is defined to be the total distance traversed by the center of the moving system. A main difference between moving a disk and moving a point is that some feasible paths for a moving point may become infeasible for a disk due to its fixed size. We propose to overcome the problem by changing the obstacles, i.e. using the Minkowski sum of the obstacles and the disk to form new obstacles, which will be used to replace the original obstacles. Meanwhile, we replace the disk by a point. In this way, we convert the problem of moving a disk into a moving point problem with new obstacles. Then, we apply E-JOB to find the solution. The shortest path for the point mover with the new obstacles is the shortest path for the disk with the original obstacles.

In the second problem, we consider the Euclidean distance between two sets P and Q in $\mathbf{R}^n$, i.e. the shortest path between them. This is an important problem in computer-aided design and computer graphics in which people often want to know whether two objects intersect or how far there are. In [5], the authors focus only on convex sets. Later, another method is proposed to deal with non-convex objects by breaking them into convex components and then apply the algorithms for convex sets to determine the distance between components [11]. The authors also reduced the complexity significantly by eliminating the number of possible pairs of components. In this paper, we demonstrate that E-JOB can be easily modified to solve this problem. The idea is to treat the starting point X and the ending point$Y$ as undetermined junctions and evolved on the boundaries of P and Q respectively. An clear advantage of E-JOB is that it places no restrictions on the obstacles, whether they are convex or non-convex, discrete or having smooth boundaries.

The last problem concerns the shortest path in an environment that the obstacles appear or disappear over time. It is not our intention to investigate the time-varying shortest path problem as described in [9] which studies how the robot moves to reach Y with the minimal cost in a dynamic environment, which includes but not limited to moving, shape changing(shrinking, expanding, etc) or appearing/disappearing obstacles. Instead, we focus on how to recompute the shortest path with appearing or disappearing obstacles assuming that E-JOB runs in real-time. This is an extremely challenging situation for the existing methods. For example, in the method of front propagation, the distance function at all grid points must be recomputed to find the solution once an obstacle appears or disappears. This could be very costly due to the involvement of solving PDEs multiple times. However, E-JOB can treat it efficiently, because many junctions remain unchanged, although some of them appear or disappear as the appearance or disappearance of obstacles. This means the shortest path stays the same mostly regardless of the changing environment. Since all of the remaining junctions already settle down, only local updates of the junctions are required.

The paper is arranged as follows. In section one, we give an introduction to E-JOB. We discuss only the relevant mathematics and algorithm. We refer readers to [3, 4] for more details. Each of the three sections followed describes one problem as stated above. In the last section, we give a conclusion and discuss some future work. For simplicity, all the examples are given in $\mathbf{R}^2$. However, the methods presented in this paper can be extended to higher dimensions without change.

2. An overview of E-JOB. In this section, we briefly introduce E-JOB algorithm proposed in [3] in the most general form. When dealing with polygonal or polyhedron obstacles, a more specified and faster version of E-JOB is available in [4]. We only discuss setup and mathematics relevant to the three particular applications we are going to introduce in this paper.

Consider N obstacles $P_k, 1 \leq k \leq N$ in $\mathbf{R}^2$. Each obstacle P_k can be represented by its signed distance function $\phi_k(x)$,

$$\phi_k(x) = \begin{cases} -d_k(x), & x \in P_k; \\ d_k(x), & x \in P_k^c \end{cases} \tag{1}$$

where $d_k(x)$ is the distance between x and the boundary of P_k, denoted by ∂P_k.

A path is a curve γ in $\mathbf{R}^2$, which is a continuous map:

$$\gamma\colon [0,1] \to \mathbf{R}^2.$$

and it is said to be feasible for a point mover if γ doesn't intersect with any obstacle, i.e.

$$\phi_k(\gamma(t)) > 0, \quad t \in [0,1], \quad 1 \leq k \leq N.$$

A path γ is rectifiable if its Euclidean length is finite and let $L(\gamma)$ be the length of it. Denote F the set of all feasible and rectifiable paths such that $\gamma(0) = X$, $\gamma(1) = Y$. The shortest path connecting X and Y we are interested in is

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

In [3], we developed an SDE based strategy to find the shortest path, using the following observation,

THEOREM 1. *Let the boundaries 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*

1. *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.*
2. *For a given straight line component of γ_{opt}, it is tangent to the boundary of the corresponding obstacles provided the boundary of the obstacles is C^1 in a neighborhood of the connecting point.*

In other words, for the optimal path γ_{opt}, there exists a sequence of junction points $(x_0^*, x_1^*, \ldots, x_n^*, x_{n+1}^*)$ with $x_0^* = X, x_{n+1}^* = Y$, such that each x_i^* is connected to its two neighbors either by a line segment or a curve that is part of the boundary of an obstacle. Now, we consider all feasible paths sharing the similar structure, i.e. a path γ determined by a sequence of junctions $(x_0, x_1, \cdots, x_n, x_{n+1})$ (We stress that the number of junctions n may vary in the algorithm). We denote x_i^c the neighboring point that is connected to x_i by a curve and x_i^s the point connected to x_i by straight line segment (The superscripts are self-explanatory). We assume $x_0^c = x_0$ and $x_{n+1}^c = x_{n+1}$. For any two points x, y on the boundary of P_k, let $d_k(x,y)$ be the distance between them along the boundary. This distance function can be computed efficiently by a traverse of the boundary, in other words by straigtening the boundary. See numerical implementation section in [3] for details.

For each x_i, let k_i be the obstacle index that x_i belongs to and

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

The length of γ is then

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

The resulting problem becomes a finite dimensional optimization problem. Hence gradient descent with Intermittent Diffusion for global optimization can be applied.

More precisely, one can solve the following stochastic differential equation to determine the optimal positions of junctions x_i,

$$dx_i = -\nabla J(x_i)dt + \sigma(t)\mathbf{T}dW(t). \tag{2}$$

Here $\mathbf{T}$ is the unit tangent in counter-clockwise direction, $W(t)$ is the standard brownian motion and $\sigma(t)$ is a step function representing the magnitude of the noise, i.e.,

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

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 $[S_i, T_i]$. Intuitively, in the time interval that $\sigma(t)$ is 0, x_i converges to a local minimizer following the negative gradient flow, while on the contrary x_i jumps out of a local trap and has a chance to converge to a different local minimizer in the interval that $\sigma(t)$ is positive. This particular choice of σ enables us to obtain not only the global optimizer but also a series of local minimizers. In our implementation, both the interval $[S_i, T_i]$ and the magnitude of σ_i are chosen randomly. For more details of intermittent diffusion, we refer readers to [2].

Another crucial part of E-JOB is how to handle vanished or newly generated junction points. New junctions are generated when a straight line segment of the path intersects with an obstacle during the evolution. In this case, E-JOB simply insert the intersection points into the sequence of junctions. On the other hand, when two straight components $\overline{x_i z}$, $\overline{x_j z}$ share a common junction z, the path can be shortened by connecting x_i, x_j directly. We hence remove z from the sequence of junctions.

We excerpt the algorithm from [3] as follows.

1. Initialization. The initial junctions consist of all the intersection points of the straight line $\overline{XY}$ with the boundaries of the obstacles.
2. Update each junction x_i by computing the stochastic differential equation (2) 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 junction, add or remove points according to the following cases:
    - (a) Adding junctions. 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 junctions.
    - (b) Eliminating junctions. If $x_i = x_i^c$, then we remove x_i and x_i^c from the set of junctions. Then add the intersection points if $\overline{x_i^s x_j^s}$, where $x_j = x_i^c$, intersects with the obstacles.
3. Update each junction x_i by the gradient flow (2) with $\sigma(t) = 0$ until a convergence criterion is satisfied. And record the path connecting x_i at the final states. In each time step of updating the junctions, 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 sort them to obtain the optimal one.

Equation (2) can be discretized by many well established schemes. In [3], Euler scheme is used, in which the noisy term $dW(t)$ is discretized by

$$dW(t) \sim \sqrt{\Delta t}\xi,$$

where $\xi \sim N(0, 1)$ is standard normal random variable and Δt is the step size. The tangent direction $\mathbf{T}(x)$ is computed by rotating the normal $\mathbf{n}(x)$ which is readily obtained via the level set function

$$\mathbf{n} = \nabla \phi_k(x).$$

We note the updated junction is projected onto the boundary of P_k.

3. The shortest path for moving a disk. In this section, we consider the shortest path between two points for a disk D with radius r. Given the size of the mover, the feasible path for a point may become infeasible for the disk. To cope the scenario, we take the idea of replacing the obstacles, denoted their region by P, by their expanded versions to account for the size of the disk. One way to achieve this is to form the Minkowski sums $\bar{P} = P \oplus (-D)$. In the level set framework, this can be accomplished easily. Suppose $\phi_k(x)$ is the signed distance function that represents the obstacle, then $\phi_k(x) - r$ represents the enlarged obstacle by length r as desired. In this case, a path is said to be feasible if the disk centered at $\gamma(t)$ doesn't intersect with any obstacle, i.e.

$$\phi_k(\gamma(t)) > r, \quad t \in [0,1], \quad 1 \leq k \leq N.$$

Let us define $\bar{\phi_k} = \phi_k - r$, then it is easy to see that

Theorem 2. *A path $\gamma(t)$ for the disk D is feasible if and only if*

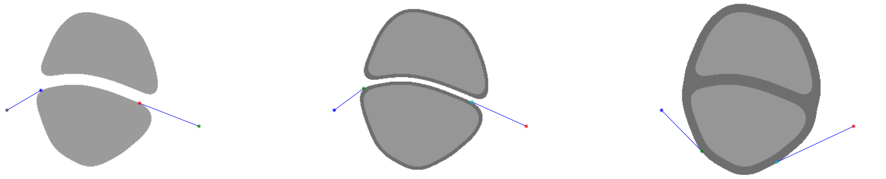
$$\bar{\phi_k}(\gamma(t)) > 0, \quad t \in [0,1], \quad 1 \leq k \leq N.$$

The length of the path is naturally taken to be the distance that the disk center traverses. Thus the shortest path for a disk is equivalent to the shortest path for a point in the new environment where the level set function ϕ_k for the original obstacles are replaced by $\bar{\phi_k}$. With this setup, the path can be readily computed by E-JOB. The following is an example for moving a disk.

Example. In Figure (2), we illustrate how radius of the disk affects the shortest path. The left picture shows the shortest path for a point mover obtained by E-JOB. It goes through the tunnel. The middle picture is the path for a disk with radius $r = 0.02$. The obstacles are flattened. The darker area is the new region in the enlarged obstacles represented. The tunnel is narrowed. Since the radius of the disk is relatively small, the tunnel is large enough for the disk to pass through. However, when the radius increases up to 0.05, the tunnel is stuck which forces the shortest path to go "outside" of the obstacle (shown on the right of Figure 2).

Remark 1. The above approach can be equally applied to non-circular movers, since any shape of movers can be covered by a disk. The tradeoff depends on the shape and whether rotations are allowed in the plan. For a detailed treatment of non-circular movers, we refer readers to [9].

4. The shortest path between two regions. We consider in this section the shortest path problem when the starting and ending points X, Y become two regions P and Q, in other words, the distance between two Euclidean regions. If P and Q are treated as two new obstacles, the problem differs little to the original shortest path problem in the framework of E-JOB. In fact, the length of the two segments adjacent to junction x_i, $J(x_i)$, remains the same while x_0 and x_{n+1} now become two



(a) $L = 0.6266$, $r = 0$ (b) $L = 0.6347$, $r = 0.02$ (c) $L = 0.6952$, $r = 0.05$

FIG. 2. *Example: shortest path with disk movers: Left: a point mover, the shortest path is through the tunnel. Middle: a disk with radius $r = 0.02$. The tunnel is still large enough for the disk to pass through. Right: A disk with radius $r = 0.05$, which is too large to move through the tunnel. The shortest path must go outside.*

new undetermined junctions instead of being fixed. Therefore, the length functional is

$$L(\gamma) = \mathcal{L}(x_0, x_1, \ldots, x_n, x_{n+1}) = \frac{1}{2}\sum_{i=0}^{n+1} J(x_i).$$

More precisely, we have the following theorem similar to theorem 1.

THEOREM 3. *Let the boundaries 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*

1. *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. In other words, γ_{opt} admits representation*

$$(x_0, x_1, \cdots, x_n, x_{n+1}) \tag{3}$$

   where x_i are junctions on the boundaries.
2. *For any $1 \le i \le n$, $\overline{x_i x_i^s}$ is tangent to ∂P_{k_i} given ∂P_{k_i} is C^1 in a neighborhood of x_i.*
3. *$\overline{x_0 x_0^s}$ and $\overline{x_{n+1} x_{n+1}^s}$ are perpendicular to ∂P_{k_0} and $\partial P_{k_{n+1}}$, respectively given that ∂P_{k_0} and $\partial P_{k_{n+1}}$ are C^1 in a neighborhood of x_0 and x_{n+1}.*

Proof. The first and second parts follows exactly as in the proof in [3]. For part (3), notice that the length functional can be written as

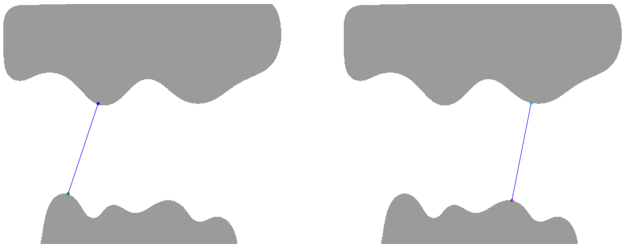
$$\mathcal{L}(x_0, \cdots, x_{n+1}) = \frac{1}{2}\sum_{i=0}^{n+1} J(x_i), \quad x_i \in \partial P_{k_i}. \tag{4}$$

Therefore by setting the derivative of $\mathcal{L}$ w.r.t x_0 to 0 (in the one dimensional manifold ∂P_{k_0}), we have

$$\frac{\partial \mathcal{L}}{\partial x_0} = \frac{\partial \|x_0 - x_1\|}{\partial x_0} = \frac{x_0 - x_1}{||x_0 - x_1||} \cdot \mathbf{T} = 0. \tag{5}$$

Here $\mathbf{T}$ is the tangent at x_0 on ∂P_{k_0}. This implies $x_0 - x_1$ is perpendicular to $\mathbf{T}$. □

REMARK 2. In part (2) of the above theorem, if x_i is at a singularity point of ∂P_{k_i}, the tangent is not defined. However, locally $\overline{x_i x_i^s}$ is still the shortest line segment connecting the two obstacles. Similar result is true for part (3) and Theorem (1).



(a) $L = 0.2840$ (b) $L = 0.2979$

FIG. 3. *Shortest path between two sets. The left one is the global optimal solution.*

Unlike the initialization as stated in the algorithm in Section 2, it is more challenging to find a good initialization here. The reason is that the starting point x_0 and ending point x_{n+1} are variables themselves. When X and Y are fixed, the initialization which consists of the intersections of the line segment $\overline{XY}$ and all the obstacles is effective and efficient in the sense that it leads to the globally shortest path more quickly. However, when X and Y varies, there is no particular criterion to tell which initialization is better than the other. As a result, the initialization we choose is simply randomly select x_0 and x_{n+1} on the boundaries of P and Q respectively. For the same reason, it usually takes more time to converge to the global optimizer. We illustrate it by the following example.

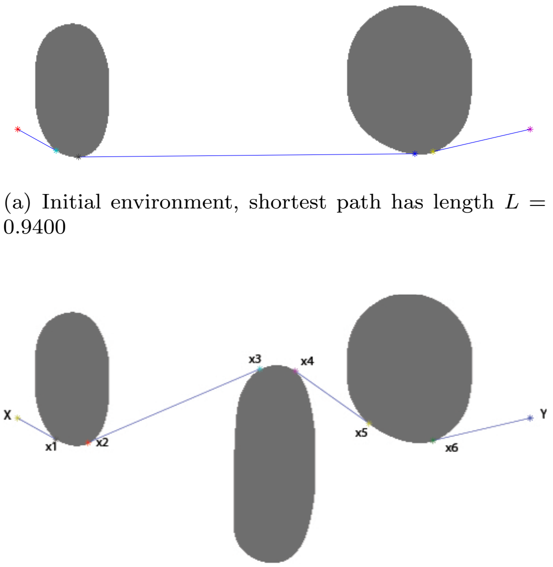
EXAMPLE. We compute the distance between two islands whose boundaries have a couple of bumps. It is not hard to see there are at least 16 local minimizers. We show two of them in Figure 3. Among them, the left one is the globally shortest path.

5. The shortest path with obstacles appearing or disappearing. This section concerns the shortest path between two fixed points in a dynamic environment. More specifically, we consider the scenario in which obstacles appear or disappear over time. We stress that it is not the usual time-varying shortest path problem as discussed, for example, in [9]. The main difference is that the functional (length of the path) we want to minimize doesn't involve time because the obstacles are not in motion. This is still a common scenario in practice, especially in robotics. And E-JOB provides an extremely low-cost solution as compared to other existing algorithms, such as the PDE based methods. In fact, as we demonstrate in the examples, it often requires only local updates to obtain the shortest path by E-JOB with appearing or disappearing obstacles.

For simplicity, we only consider the scenario in which one obstacle pops out or disappear. The case in which multiple obstacles appear or disappear can be handled in the same manner. Suppose we have N obstacles initially and the shortest path between X and Y is obtained by E-JOB with the following junctions,

$$\gamma = (x_0, x_1, \ldots, x_n, x_{n+1}).$$

Assume that a new obstacle, denoted by P_{N+1}, appears. Without loss of generality, let us assume that P_{N+1} only intersects with line segment $\overline{x_k x_{k+1}}$ for some k and



(a) Initial environment, shortest path has length $L = 0.9400$

(b) One obstacle appears, globally minimizer has length $L = 0.9992$

Fig. 4. *Shortest path with obstacle appearing*

denote the intersections by

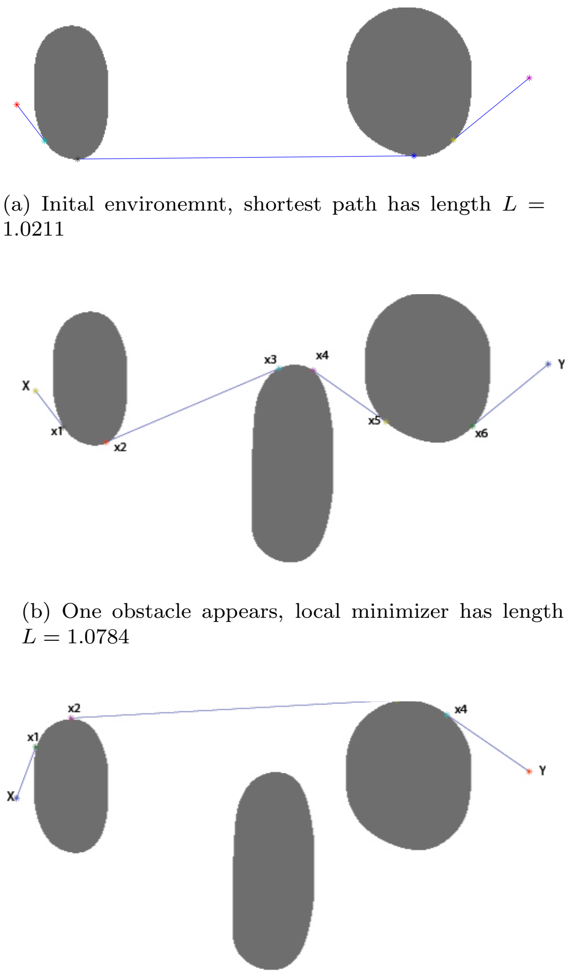
$$(x_k, x_{n+2}, x_{n+3}, \ldots, x_{n+l}, x_{k+1}). \tag{6}$$

The new set contains $n+l+1$ junctions in total. One can apply E-JOB to obtain the globally optimal solution if needed. However, if a local minimizer is needed, only $l+1$ junctions need to be evolved. The junctions not in (6) are already settled down. Moreover, from our experience and the examples showing in this section, since $x_0, \ldots, x_{n+1}$ is part of the globally optimal path, the local minimizer obtained by evolving $x_{n+2}, \ldots, x_{n+l}$ is very likely to be the global minimizer, although this is not necessarily always the case.

The case in which an obstacle disappears can be handled in a similar manner. Assume obstacle P_k disappears at certain time. If there were no junctions on the boundary of P_k, the shortest path remains the same. If x_i and x_{i+1} were on P_k, we shorten the length by directly connecting x_i^s and x_{i+1}^s with a line segment and removing junctions x_i and x_{i+1}. If line segment $\overline{x_i^s x_{i+1}^s}$ intersects with any existing obstacles, we add the intersections as new junctions.

Examples. We give four examples with two showing obstacle appearing and two for obstacle disappearing . In the first two examples (Figure 4A, 5A), there are two obstacles initially. At certain time, a new obstacle (in the middle) pops out and changes the shortest path accordingly. In the first example, the local minimizer (4B) obtained by evolving new junctions x_3, x_4 is the global optimizer. Notice that x_1, x_6 remain the same. On the other hand, the second example demonstrates the contrary situation. Local updates in Figure (5B) only leads to a local minimizer. All the junctions x need to be computed in order to obtain the globally shortest path (5C).

The case in which the middle obstacle disappears are represented by the same



(a) Inital environemnt, shortest path has length $L = 1.0211$

(b) One obstacle appears, local minimizer has length $L = 1.0784$

(c) One obstacle appears, globally minimizer has length $L = 1.0593$

FIG. 5. *shortest path with obstacle appearing*

figures with reversed orders. More specifically, Figure (4B,5C) are new initial environments and Figure (4A,5A) are the globally shortest path respectively after the middle obstacle disappears.

6. Conclusion. In this paper, we apply E-JOB that we developed previously to three problems from practice: moving a disk, computing the distance between sets, and re-calculated the shortest path with appearing or disappearing obstacles. They can be solved by E-JOB with minimal modifications, so they enjoy the advantages of E-JOB, which we summarize here:

1. It can deal with any shape of obstacles by incorporating the level set framework. This is especially convenient when handling dynamical environment with obstacles moving or changing geometry and topology.
2. It's dimension independent. It has been successfully applied to environment with 3-D polyhedral obstacles [4], which is a NP-hard problem in the classical

combinatorial framework.

3. It can find the globally optimal path. In addition, it obtains a series of locally optimal paths in the process. This makes it especially useful when a time limit is imposed.

As evident in our third application, E-JOB can successfully handle appearing or disappearing obstacles. This shed the light on how to use E-JOB to tackle the classical time-varying shortest path problems with dynamical environment with moving obstacles. It is a much harder problem with different considerations, such as temporal dependent cost functions associated with the optimal conditions. This is beyond the scope of this paper and we will report our findings in the future.

REFERENCES

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

[2] S.-N. Chow, T.-S. Yang, and H. Zhou, *Global Optimizations by Intermittent Diffusion*, Chaos, CNN, Memristors and Beyond, 2013.

[3] S.-N. Chow, J. Lu, and H.-M. Zhou, *Finding the shortest path by evolving junctions on obstacle boundaries (e-job): An initial value ode's approach*, Applied and Computational Harmonic Analysis, 35:1 (2013), pp. 165–176.

[4] S.-N. Chow, J. Lu, and H.-M. Zhou, *Shortest path amid 3-d polyhedral obstacles*, submitted to SIAM multiscale modeling and simulation, available at `http://www.math.ucla.edu/applied/cam`, #13-54, 2013.

[5] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, *A fast procedure for computing the distance between complex objects in three-dimensional space*, Robotics and Automation, IEEE Journal of, 4:2 (1988), pp. 193–203.

[6] J. E. Goodman and J. O'Rourke, *Handbook of discrete and computational geometry*, Chapman & Hall/CRC, 2004.

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

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

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

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

[11] S. Quinlan, *Efficient distance computation between non-convex objects*, in "Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on", pp. 3324–3329. IEEE, 1994.

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