

Derivatives with Respect to Metrics and Applications: Subgradient Marching Algorithm

F. Benmansour, G. Carlier, G. Peyré, F. Santambrogio *

January 8, 2009

Abstract

This paper describes the Subgradient Marching algorithm to compute the derivative of the geodesic distance with respect to the metric. The geodesic distance being a concave function of the metric, this algorithm computes an element of the subgradient in $O(N^2 \log(N))$ operations on a discrete grid of N points. It performs a front propagation that computes the subgradient of a discrete geodesic distance. Equipped with this Subgradient Marching, a Riemannian metric can be designed through an optimization process. We show applications to landscape modeling and to traffic congestion. Both applications require the maximization of geodesic distances under convex constraints, and are solved by subgradient descent computed with our Subgradient Marching. We also show application to the inversion of travel time tomography, where the recovered metric is the local minimum of a non-convex variational problem involving geodesic distances.

Keywords: Geodesics, Eikonal equation, subgradient descent, Fast Marching Method, traffic congestion, travel time tomography.

1 Introduction

1.1 Riemannian Metric Design

The shortest path between a pair of points for a given Riemannian metric defines a curve that tends to follow areas where the metric is low. It is an object of primary interest in both pure mathematics and applied fields. For instance, as far as applications are concerned, such minimal paths are used intensively in computer vision and medical image analysis to perform segmentation of objects and extraction of tubular vessels [8]. The metric is designed to be low around the boundary of organs and vessels so that shortest paths follow these salient features.

In some applications, the Riemannian metric is the object of interest, and should be computed from a set of constraints or criteria. Some of these constraints might involve the length of geodesic curves between sets of key points, and these geodesic distances should be maximized or minimized. As shown in this paper, the maximization of geodesic lengths leads to convex problems, whereas the minimization of the distance leads to a non-convex problem. A global (for maximization) or local (for minimization) solution to these metric design problems can be found using a subgradient descent.

* CEREMADE, UMR CNRS 7534, Université Paris-Dauphine, Pl. de Lattre de Tassigny, 75775 Paris Cedex 16, FRANCE {benmansour, carlier, peyre, filippoo}@ceremade.dauphine.fr.

This paper proposes the first algorithm to compute such a subgradient of the geodesic distance with respect to the metric. It can thus be used as a building block for an optimization procedure that computes an optimal metric according to criteria on the length of the geodesic curves.

1.2 Geodesic Distances.

Riemannian metric. An isotropic Riemannian metric ξ on a domain $\Omega \subset \mathbb{R}^d$ defines a weight $\xi(x)$ that penalizes a curve $\gamma(t)$ passing through a point $x = \gamma(t) \in \Omega$. The length of the curve according to ξ is

$$L_\xi(\gamma) = \int_0^T |\gamma'(t)|\xi(\gamma(t))dt. \quad (1.1)$$

This metric ξ defines a geodesic distance $d_\xi(x_0, x)$ that is the minimal length of rectifiable curves joining two points $x_0, x \in \Omega$

$$d_\xi(x_0, x) = \min_{\gamma(0)=x_0, \gamma(1)=x} L_\xi(\gamma). \quad (1.2)$$

The distance map

$$\mathcal{U}^\xi(x) = d_\xi(x_0, x) \quad (1.3)$$

to the starting point x_0 is a function of the metric ξ , where we have drop the dependence with respect to x_0 for simplicity. The mapping $\xi \mapsto \mathcal{U}^\xi(x)$ is the one we wish to maximize or minimize in this paper, where x_0 and x are fixed points.

The geodesic curve γ joining x_1 to x_0 is the solution of an ordinary differential equation that corresponds to a gradient descent of \mathcal{U}^ξ

$$\frac{d\gamma(s)}{ds} = -\text{grad}_{\gamma(s)}\mathcal{U}^\xi \quad \text{and} \quad \gamma(0) = x_1, \quad (1.4)$$

where $\text{grad}_x\mathcal{U}^\xi \in \mathbb{R}^d$ is the usual gradient of the function $x \mapsto \mathcal{U}^\xi(x)$. This should not be confused with the subgradient with respect to the metric defined in the following paragraph.

Geodesic subgradient. The design of a metric through the maximization or minimization of $d_\xi(x_0, x)$ requires to compute the gradient $g = \nabla_\xi\mathcal{U}^\xi(x)$ of the mapping $\xi \mapsto \mathcal{U}^\xi(x)$. For any location $y \in \Omega$, $g(y)$ tells how much the geodesic distance between x_0 and x is sensitive to variations on $\xi(y)$.

In the continuous framework of (1.1) and (1.2), a small perturbation $\xi_\varepsilon = \xi + \varepsilon h$ defines a geodesic distance map $\mathcal{U}^{\xi_\varepsilon}(x)$ between x and x_0 , that can be differentiated with respect to ε at $\varepsilon = 0$

$$\frac{d}{d\varepsilon}\mathcal{U}^{\xi_\varepsilon}(x)\Big|_{\varepsilon=0} = \int_\gamma h d\mathcal{H}^1 = \int_0^1 h(\gamma(t))|\gamma'(t)|dt, \quad (1.5)$$

where the curve γ is the geodesic from x_0 to x according to the metric ξ . If γ is unique, this shows that $\xi \mapsto \mathcal{U}^\xi(x)$ is differentiable at ξ , and that the gradient g is a measure supported along the curve γ . In the case where this geodesic is not unique, this quantity may fail to be differentiable. Yet, the map $\xi \mapsto \mathcal{U}^\xi(x)$ is anyway concave (as an infimum of linear quantities in ξ) and for each geodesic we get an element of the super-differential through Equation (1.5). In the sequel we will often refer to subgradients and subdifferentials for the concave function $\xi \mapsto \mathcal{U}^\xi(x)$ instead of superdifferentials and supergradients, this slight abuse of terminology should not create confusion however.

The extraction of geodesics is quite unstable, especially for metrics such that x and x_1 are connected by many curves of length close to the minimum distance $d_\xi(x_0, x)$. It is thus unclear how to discretize in a robust way the gradient of the geodesic distance directly from the continuous definition (1.5). This paper proposes an alternative method, where g is defined unambiguously as a subgradient of a discretized geodesic distance. Furthermore, this discrete subgradient is computed with a fast Subgradient Marching algorithm.

Figure 1 shows two examples of subgradient computations. Near a degenerate configuration, we can see that the subgradient g might be located around several minimal curves.

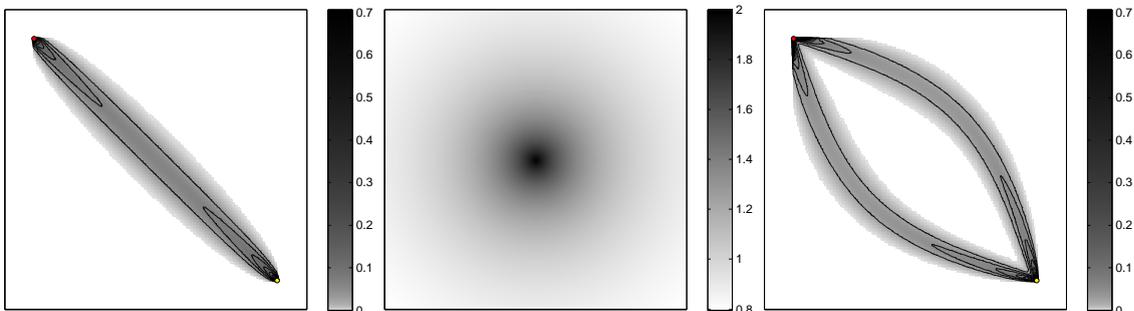


Figure 1: On the left, $\nabla_\xi \mathcal{U}(x_1)$ and some of its iso-levels for $\xi = 1$. In the middle, a non constant metric $\xi(x) = 1/(1.5 - \exp(-\|c - x\|))$, where c is the center of the image. On the right, an element of the superdifferential of the geodesic with respect to the metric shown in the middle figure.

Anisotropic metrics. The geodesic distance and its subgradient can be defined for more complicated Riemannian metric ξ that depends both on the location $\gamma(t)$ of the curve and on its local direction $\gamma'(t)/|\gamma'(t)|$. The algorithm presented in this paper extends to this more general setting, thus allowing to design arbitrary anisotropic Riemannian metric. We decided however to restrict our attention to the isotropic case, that has many practical applications.

1.3 Previous Works and Contributions

Geodesic distance computation. The estimation of distance maps \mathcal{U}^ξ has been intensively studied in numerical analysis and can be approximated on discrete grid of N with the Fast Marching Method of Sethian [13], and Tsitsiklis [14] in $O(N \log(N))$ operations. This algorithm has opened the door to many application in computer vision where the minimal geodesic curves extracts image features, see for instance [13, 8]. Section 2 recalls the basics of the discretization of geodesic distance and Section 2.3 details the front propagation procedure underlying the Fast Marching method.

Geodesic distance optimization. The optimization of \mathcal{U}^ξ with respect to ξ is much less studied. It is however an important problem in some specific fields, such as for landscape design, traffic congestion and seismic imaging. In these applications, the metric ξ is optimized to meet certain criteria, or is recovered by optimization from a few geodesic distance measures.

This paper tackles directly the problem of optimizing quantities involving the distance function \mathcal{U}^ξ by computing a subgradient $\nabla_\xi \mathcal{U}^\xi(x)$ of the mapping $\xi \mapsto \mathcal{U}^\xi(x)$ for a given

point x . The Subgradient Marching algorithm is described in Section 3. It follows the optimal ordering used by the Fast Marching, making the overall process only $O(N^2 \log(N))$ to compute a subgradient of the maps $\xi \mapsto \mathcal{U}^\xi(x)$ for all the grid points x .

This Subgradient Marching computes an exact subgradient of the discrete geodesic distance, so that it can be used to minimize variational problems involving geodesic distances. We believe it is important to first discretize the problem of interest and perform an exact minimization of the discrete problem. As far as geodesic quantities are involved, discretizing optimality condition of a continuous functional is indeed highly unstable.

Landscape design. Shape design requires the modification of the Riemannian metric defined by the first fundamental form of the surface. Minimization of geodesic length distortion is a well studied criterion to perform surface flattening and shape comparison, see for instance [3].

This paper tackles directly the problem of optimizing a Riemannian metric ξ . The example of landscape design using a fixed amount of resources is studied in Section 4.1. The length of geodesics is maximized under local and global constraint on the metric. This problem has a unique solution that can be found using a subgradient computed with our Subgradient Marching algorithm.

An application to travel time tomography is shown in Section 4.3. A subgradient descent allows one to find a local minimum of a variational energy involving geodesic distances.

Traffic congestion. A continuous generalization of the *Wardrop equilibria* [15], originally proposed in [5], involves the maximization of a concave functional depending on the geodesic distances between landmarks. A subgradient descent approximates this continuous solution and [1] describes an algorithm that makes use of our Subgradient Marching. Section 4.2 recalls basic facts of this congestion approximation and shows some numerical examples.

Seismic imaging. Seismic imaging computes an approximation of the underground from few surfaces measurements [6, 11]. This corresponds to an ill posed inverse problem that is regularized using smoothness prior information about the ground and simplifying assumption about wave propagation. Discarding multiple reflexions, the first arrival time of a pressure wave corresponds to the geodesic distance to the source, for a Riemannian metric ξ that reflects the properties of the underground.

The recovery of ξ from a few measurements $d_\xi(x_i, x_j)$ between sources x_i and sensors x_j corresponds to travel time tomography. A least square recovery of ξ involve the optimization of the geodesic distance. It has been carried over using for instance adjoint state methods [6, 11] that involve many computations of the geodesic map \mathcal{U}^ξ for a varying metric ξ . Our Subgradient Marching algorithm allows to find a local minimum of the regularized least square energy using a descent method.

2 Discrete Geodesic Distances

2.1 Discretization

Eikonal Equation Our approach to minimize geodesic distances first defines a discrete geodesic distance \mathcal{U}^ξ , solution of a discretized partial differential equation. A discrete subgradient $\nabla_\xi \mathcal{U}^\xi(x)$ of the map $\xi \mapsto \mathcal{U}^\xi(x)$ is then defined to solve exactly discrete

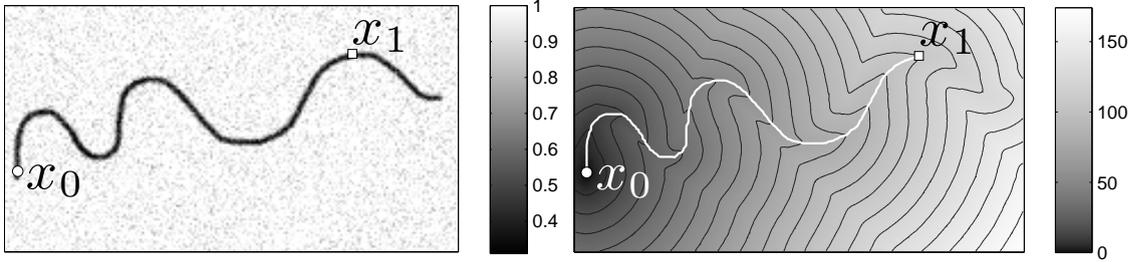


Figure 2: Example of the minimal path computation using the Fast Marching algorithm. On the left: the metric ξ . On the right: The minimal action map \mathcal{U} and the minimal path linking x_1 to x_0 .

variational problems involving geodesic distances. This is a general framework that could be extended to a larger class of non-linear partial differential equations.

The geodesic map $\mathcal{U}^\xi(x)$ defined in (1.3) is the unique viscosity solution of the Eikonal non-linear PDE (see [10])

$$\begin{cases} \|\nabla \mathcal{U}^\xi(x)\| = \xi, \\ \mathcal{U}^\xi(x_0) = 0. \end{cases} \quad (2.1)$$

The computation of $\mathcal{U}^\xi(x)$ thus requires the discretization of (2.1) so that a numerical scheme captures the viscosity solution of the equation.

Upwind Discretization In the following, we describe the computation in 2D and assume that the domain is $\Omega = [0, 1]^2$, although the scheme carries over for an arbitrary domain in any dimension.

We will also drop the dependence on ξ and x_0 of the distance map $\mathcal{U}^\xi = \mathcal{U}$ to ease the notations. The geodesic distance map \mathcal{U}^ξ is discretized on a grid of $N = n \times n$ points, so that $\mathcal{U}_{i,j}$ for $0 \leq i, j < n$ is an approximation of $\mathcal{U}^\xi(ih, jh)$ where the grid step is $h = 1/n$. The metric ξ is also discretized so that $\xi_{i,j} = \xi(ih, jh)$.

Classical finite difference schemes do not capture the viscosity solution of (2.1). Upwind derivative should be used instead

$$\begin{aligned} D_1 \mathcal{U}_{i,j} &:= \max\{(\mathcal{U}_{i,j} - \mathcal{U}_{i-1,j}), (\mathcal{U}_{i,j} - \mathcal{U}_{i+1,j}), 0\}/h, \\ D_2 \mathcal{U}_{i,j} &:= \max\{(\mathcal{U}_{i,j} - \mathcal{U}_{i,j-1}), (\mathcal{U}_{i,j} - \mathcal{U}_{i,j+1}), 0\}/h. \end{aligned}$$

As proposed by Rouy and Tourin [12], the discrete geodesic distance map $\mathcal{U} = (\mathcal{U}_{i,j})_{i,j}$ is found as the solution of the following discrete non-linear equation that discretizes (2.1)

$$D\mathcal{U} = \xi \quad \text{where} \quad D\mathcal{U}_{i,j} = \sqrt{D_1 \mathcal{U}_{i,j}^2 + D_2 \mathcal{U}_{i,j}^2}. \quad (2.2)$$

Rouy and Tourin [12] showed that this discrete geodesic distance \mathcal{U} converges to \mathcal{U}^ξ when h tends to 0.

Figure 2 shows an example of a discrete geodesic distance map \mathcal{U} . The metric ξ has a low value along a black curve, so that the geodesic curves tends to follow this feature. An example of geodesic curve is shown on the right, that is obtained by a numerical integration of the ordinary differential equation (1.4).

2.2 Concavity of the Geodesic Distance

To solve variational problems involving the geodesic distance $d_\xi(x_0, x)$, for $x = (ih, jh)$, one would like to differentiate with respect to ξ the discrete distance map $\mathcal{U}_{i,j}^\xi$, obtained by solving (2.2). Actually, this is not always possible, since the mapping $\xi \mapsto \mathcal{U}_{i,j}^\xi$ is not necessary smooth. The following proposition proves that $\mathcal{U}_{i,j}^\xi$ is a concave function of ξ and this allows for superdifferentiation (the correspondent of subdifferential for concave functions instead of convex).

Proposition 2.1. *For a given point (i, j) , the functional $\xi \mapsto \mathcal{U}_{i,j}^\xi$ is concave.*

Proof. In the following we drop the dependence on (i, j) and note $\mathcal{U}^\xi = \mathcal{U}_{i,j}^\xi$. Thanks to the homogeneity, it is sufficient to prove super-additivity. We want to prove the inequality

$$\mathcal{U}^{\xi_1 + \xi_2} \geq \mathcal{U}^{\xi_1} + \mathcal{U}^{\xi_2}.$$

Thanks to the comparison principle of Lemma 2.2 below, it is sufficient to prove that $\xi_1 + \xi_2 \geq D(\mathcal{U}^{\xi_1} + \mathcal{U}^{\xi_2})$, where the operator D is defined in (2.2). This is easily done if we notice that the operator D is convex (as it is a composition of the function $(s, t) \mapsto \sqrt{s^2 + t^2}$, which is convex and increasing in both s and t , and the operator D_1 and D_2 , which are convex since they are produced as a maximum of linear operators) and 1-homogeneous, and hence it is subadditive, i.e. it satisfies $D(u + v) \leq Du + Dv$. \square

Lemma 2.2. *If $\xi \leq \eta$, then $\mathcal{U}^\xi \leq \mathcal{U}^\eta$.*

Proof. Let us suppose at first a strict inequality $\xi < \eta$. Take a minimum point for $\mathcal{U}^\eta - \mathcal{U}^\xi$ and suppose it is not the fixed point x_0 . Computing D and using sub-additivity we have

$$\eta = D\mathcal{U}^\eta \leq D(\mathcal{U}^\eta - \mathcal{U}^\xi) + D\mathcal{U}^\xi = D(\mathcal{U}^\eta - \mathcal{U}^\xi) + \xi,$$

which gives $D(\mathcal{U}^\eta - \mathcal{U}^\xi) \geq \eta - \xi > 0$. Yet, at minimum points we should have $D(\mathcal{U}^\eta - \mathcal{U}^\xi) = 0$ and this proves that the minimum is realized at x_0 , which implies $\mathcal{U}^\eta - \mathcal{U}^\xi \geq 0$.

To handle the case $\xi \leq \eta$ without a strict inequality, just replace η by $(1 + \varepsilon)\eta$ and notice that the application $\eta \mapsto \mathcal{U}^\eta$ is continuous. \square

2.3 Fast Marching Propagation

The Fast Marching algorithm, introduced by Sethian in [13] and Tsitsiklis in [14], allows to solve (2.2) in $O(N \log(N))$ operations using an optimal ordering of the grid points. This greatly reduces the numerical complexity with respect to iterative methods, because grid points are only visited once.

We recall the basic ideas underlying this algorithm, because our Subgradient Marching computation of $\nabla_\xi \mathcal{U}^\xi(x)$ makes use of the same ordering process.

The values of \mathcal{U} may be regarded as the arrival times of wavefronts propagating from the source point x_0 with velocity $1/\xi$. The central idea behind the Fast Marching method is to visit grid points in an order consistent with the way wavefronts propagates.

In the course of the algorithm, the state of a grid point (i, j) passes successively from *Far* (no estimate of $\mathcal{U}_{i,j}$ is available) to *Trial* (an estimate of $\mathcal{U}_{i,j}$ is available, but it might not be the solution of (2.1)) to *Known* (the value of $\mathcal{U}_{i,j}$ is fixed and solves (2.1)). The set of *Trial* points forms an interface between *Known* points (initially the point x_0 alone) and the *Far* points. The Fast Marching algorithm progressively propagates this front of *Trial* points so that all grid points are visited, see Fig. 3.

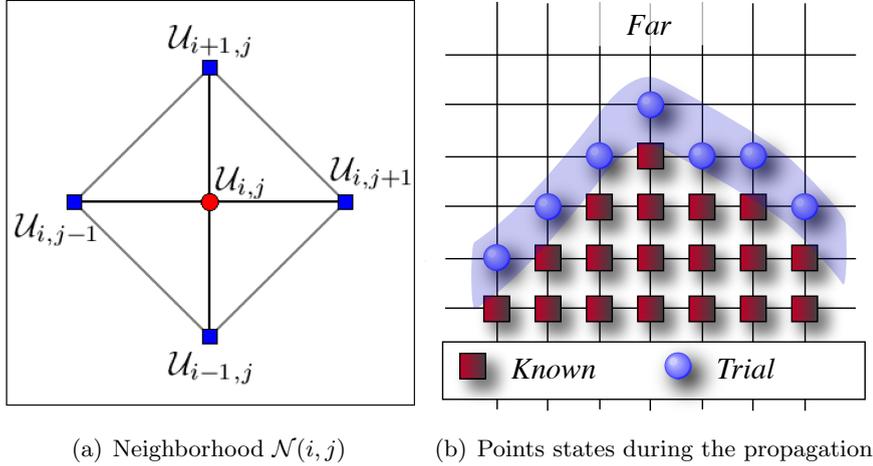


Figure 3: Fast Marching propagation on a regular grid.

At each iteration of the algorithm, a point (i, j) is tagged as *Known* so that $\mathcal{U}_{i,j}$ is the solution of (2.1). The value of \mathcal{U} at the neighboring points $(i', j') \in \mathcal{N}(i, j) = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$ such that (i', j') is not yet *Known* are updated by solving (2.2), using only the values of \mathcal{U} that are *Known*.

The decision of moving the state of a point from *Trial* to *Known* is made by selecting the *Trial* point with minimum value of \mathcal{U} . It can be shown that updating the value of \mathcal{U} by solving (2.2) can only increase the value of $\mathcal{U}_{(i',j')}$ for a trial point (i', j') , so that the values of *Known* points are ensured to solve (2.1). A heap data structure allows one to locate this minimum point in at most $\log(N)$ operations, so that the overall complexity of the algorithm is $O(N \log(N))$ operations. This is similar to Dijkstra's algorithm for computing shortest paths on graphs [9].

Table 1 Fast Marching algorithm for solving (2.1).

Initialization: Set $\mathcal{U}_{x_0} = 0$ and for all $(i, j) \neq x_0, \mathcal{U}_{i,j} = +\infty$.

Tag all $(i, j) \neq x_0$ as *Far*. Tag x_0 as *Trial*.

While the set of *Trial* points is not empty, repeat:

1. Find (i, j) the *Trial* point with minimum value of $\mathcal{U}_{i,j}$.
 2. Tag (i, j) as *Known*.
 3. For all $(i', j') \in \mathcal{N}(i, j)$ that are *Trial* or *Far*, do:
 - Tag (i', j') as *Trial*.
 - Update the value of $\mathcal{U}_{i',j'} = u$, by solving (2.2).
-

2.4 Update of the Geodesic Distance Map

Each step of the Fast Marching requires the resolution of (2.2) to update the value of $u = \mathcal{U}_{i,j}$ for a small set of points (i, j) . One thus needs to compute the solution u of

$$\max\{(u - \mathcal{U}_{i-1,j}), (u - \mathcal{U}_{i+1,j}), 0\}^2 + \max\{(u - \mathcal{U}_{i,j-1}), (u - \mathcal{U}_{i,j+1}), 0\}^2 = (h\xi_{i,j})^2. \quad (2.3)$$

This computation deserves special attention because our Subgradient Marching requires to compute the derivative of the obtained distance u .

This resolution (2.3) can be decomposed by considering independently each of the four triangles, that are composed, for any combination of signs $(a, b) = (\pm 1, \pm 1)$, of the three points $\{(i, j), (i + a, j), (i, j + b)\}$.

Depending on whether the neighboring point are *Known* or not, three possibilities can occur.

- In the case where $\mathcal{U}_{i+a,j} \neq +\infty$ and $\mathcal{U}_{i,j+b} \neq +\infty$, Equation (2.3), restricted to such a triangle, defines a solution $u_{a,b}$ which is the maximum real solution of the quadratic equation

$$(u_{a,b} - \mathcal{U}_{i+a,j})^2 + (u_{a,b} - \mathcal{U}_{i,j+b})^2 = (h\xi_{i,j})^2, \quad (2.4)$$

where by convention, $u_{a,b} = \infty$ if either $\mathcal{U}_{i+a,j} = +\infty$ or $\mathcal{U}_{i,j+b} = +\infty$.

- In the case where $\mathcal{U}_{i+a,j} \neq +\infty$ and $\mathcal{U}_{i,j+b} = +\infty$, then one defines

$$u_{a,b} = \mathcal{U}_{i+a,j} + h\xi_{i,j}. \quad (2.5)$$

- In the case where $\mathcal{U}_{i+a,j} = +\infty$ and $\mathcal{U}_{i,j+b} \neq +\infty$, then one defines

$$u_{a,b} = \mathcal{U}_{i,j+b} + h\xi_{i,j}. \quad (2.6)$$

The solution u of (2.3) is then obtained as

$$u = \min_{a=\pm 1, b=\pm 1} u_{a,b}. \quad (2.7)$$

3 Subgradient Marching Algorithm

3.1 Dependency Graph

A point $(i', j') = (i + a^*, j)$ or $(i', j') = (i, j + b^*)$ is a parent of (i, j) if $\mathcal{U}_{i',j'} \neq +\infty$ and $u = u_{a^*,b^*}$ where u is defined in (2.7). In the case where two different values of signs $(a, b) \neq (a', b')$ satisfy $u = u_{a,b} = u_{a',b'}$, one of the two sets of neighbors is chosen arbitrarily (say, w.r.t. to an order convention on the points of the grid, which is fixed a priori and is used in case of ex-aequo in the first phase of the FM algorithm as well).

When a point (i, j) is tagged as *Known* by the algorithm, its value $\mathcal{U}_{i,j}$ solves the discrete Eikonal equation (2.2), so that this values only depends on the value of its parent nodes, that are necessarily *Known*.

The Fast Marching algorithm described in Section 2.3 computes, during the propagation, a parent relationship $(i, j) \rightarrow (i + a^*, j)$ and/or $(i, j) \rightarrow (i, j + b^*)$, since each point excepted x_0 has exactly one or two parents. This defines a directed graph structure without cycle, that stores the dependencies induced by the resolution of the discrete Eikonal equation. Figure 4 shows an example of such a graph of dependencies for the metric $\xi = 1$, together with the sub-graph of the dependencies of a single point x_1 .

3.2 Recursive Subdifferentiation

Proposition 2.1 proved that for a fixed point (i, j) and a fixed source x_0 the functional $\xi \mapsto \mathcal{U}_{i,j}^\xi$ is concave. For a metric $\xi > 0$, one can thus consider an element $\nabla_\xi \mathcal{U}_{i,j}$ of the subdifferential of this functional.

The values of $\mathcal{U}_{i,j}$ at a point (i, j) depend only on the values of its parents $(i + a^*, j)$ and/or $(i, j + b^*)$ through quadratic or linear equations (2.4), (2.5) or (2.6). The subdifferential $\nabla_\xi \mathcal{U}_{i,j}$ thus also depends on the subdifferentials $\nabla_\xi \mathcal{U}_{i+a^*,j}$ and $\nabla_\xi \mathcal{U}_{i,j+b^*}$.

One has to consider several cases, depending on the number of parents of (i, j) .

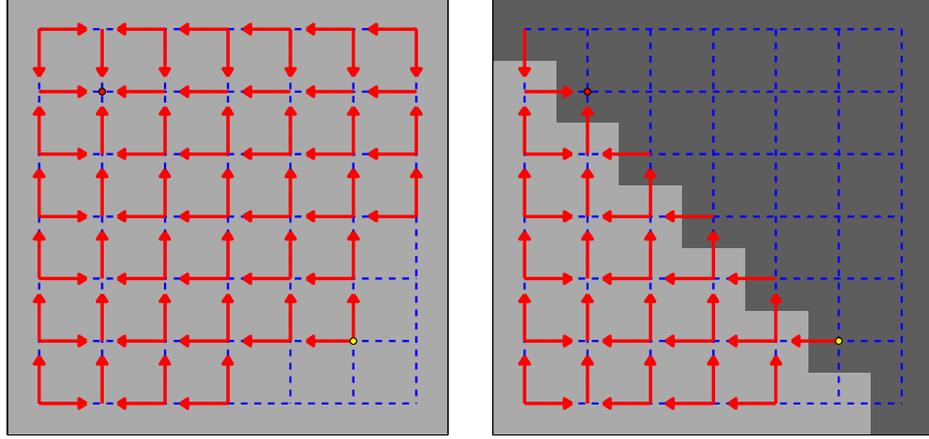


Figure 4: Graphs of dependencies of the Fast Marching computations. Left: uniform metric, right: varying metric.

- For the special case $(i, j) = x_0$, the value of $\mathcal{U}_{i,j}$ is zero and does not depend on ξ . Thus, $\nabla_{\xi}\mathcal{U}_{i,j} = \mathbf{0}$, the null vector.
- If (i, j) has two parents, differentiating (2.4) with respect to ξ leads to

$$\alpha(\nabla_{\xi}\mathcal{U}_{i,j} - \nabla_{\xi}\mathcal{U}_{i+a^*,j}) + \beta(\nabla_{\xi}\mathcal{U}_{i,j} - \nabla_{\xi}\mathcal{U}_{i,j+b^*}) = h^2\xi_{i,j}$$

where $\alpha = \mathcal{U}_{i,j} - \mathcal{U}_{i+a^*,j} \in \mathbb{R}$ and $\beta = \mathcal{U}_{i,j} - \mathcal{U}_{i,j+b^*} \in \mathbb{R}$. Since $\xi > 0$, one has $\alpha + \beta > 0$. The sub-gradient at the point (i, j) is thus the vector $\nabla_{\xi}\mathcal{U}_{i,j}$ defined as

$$\nabla_{\xi}\mathcal{U}_{i,j} = \frac{1}{\alpha + \beta} (h\xi_{i,j}\delta_{i,j} + \alpha\nabla_{\xi}\mathcal{U}_{i+a^*,j} + \beta\nabla_{\xi}\mathcal{U}_{i,j+b^*}), \quad (3.1)$$

where $\delta_{i,j}$ is the Dirac vector such that $\delta_{i,j}(i', j') = 1$ if $(i, j) = (i', j')$ and $\delta_{i,j}(i', j') = 0$ otherwise.

- If only $(i + a^*, j)$ is a parent of (i, j) , differentiating (2.5) with respect to ξ leads to

$$\nabla_{\xi}\mathcal{U}_{i,j} = \nabla_{\xi}\mathcal{U}_{i+a^*,j} + h\delta_{i,j}. \quad (3.2)$$

- If only $(i, j + b^*)$ is a parent of (i, j) , differentiating (2.6) with respect to ξ leads to

$$\nabla_{\xi}\mathcal{U}_{i,j} = \nabla_{\xi}\mathcal{U}_{i,j+b^*} + h\delta_{i,j}. \quad (3.3)$$

Applying these rules during the Fast Marching propagation allows one to compute the value of the subgradient $\nabla_{\xi}\mathcal{U}_{i,j}$ at all grid points (i, j) . The corresponding Subgradient Marching algorithm is detailed in Table 2.

Table 2 Subgradient Marching algorithm.

Initialization: Set $\mathcal{U}_{x_0} = 0$ and for all $(i, j) \neq x_0, \mathcal{U}_{i,j} = +\infty$.

Set $\nabla_{\xi} \mathcal{U}_{x_0} = \mathbf{0}$ the null vector. Tag all $(i, j) \neq x_0$ as *Far*. Tag x_0 as *Trial*.
While the set of *Trial* points is not empty, repeat:

1. Find (i, j) the *Trial* point with minimum value of $\mathcal{U}_{i,j}$.
 2. Tag (i, j) as *Known*.
 3. For all $(i', j') \in \mathcal{N}(i, j)$ that are *Trial* or *Far*, do:
 - Tag (i', j') as *Trial*.
 - Update the value of $\mathcal{U}_{i',j'}$, using either (2.4), (2.5) or (2.6).
 - Update the value of $\nabla_{\xi} \mathcal{U}_{i',j'}$, using either (3.1), (3.2) or (3.3).
-

The following theorem ensures the validity of the Subgradient Marching algorithm.

Theorem 3.1. *For $\xi > 0$, a given starting point x_0 and a given (i, j) , the vector $\nabla_{\xi} \mathcal{U}_{i,j}$ computed with Subgradient Marching belongs to the subdifferential set of the functional $\xi \mapsto \mathcal{U}_{i,j}^{\xi}$.*

Proof. Consider the set of metrics $\xi \in (\mathbb{R}_+)^N$ which give distinct values to the action map at every point, i.e. such that $\mathcal{U}_{i,j} \neq \mathcal{U}_{i',j'}$ for $(i, j) \neq (i', j')$. In this case the expression of $\mathcal{U}_{i,j}$ is given by the recursive algebraic formula (3.1), (3.2) or (3.3) involving the values of the parents. Each parent (both in the case of one parent and in the case of two parents) is defined with no ambiguity and the same tree of parental dependence would stay valid even if ξ was changed by small perturbations. It means that there exists a neighborhood of ξ such that for all other metrics $\tilde{\xi}$ in such a neighborhood all the parental relations between points are the same. Hence, for $\tilde{\xi}$ in this neighborhood the value of $\mathcal{U}_{i,j}^{\tilde{\xi}}$ is given by the same algebraic and recursive expression. The vector $\nabla_{\xi} \mathcal{U}_{i,j}$ is exactly the differential of this expression.

Now we suppose that ξ is such that there are possible ex-aequo entries in the vector $u(i, j) = \mathcal{U}_{i,j}^{\xi}$. One can slightly perturb this function by a sequence of function u_n satisfying the same strict inequalities satisfied by u (i.e., $u(i, j) < u(i', j') \Rightarrow u_n(i, j) < u_n(i', j')$) and replacing equalities by inequalities according to the order convention $\tilde{<}$ on the points of the grid: this means

$$u(i, j) = u(i', j'), (i, j) \tilde{<} (i', j') \quad \Rightarrow \quad u_n(i, j) < u_n(i', j').$$

This is possible by small perturbations, so that $u_n(i, j) \rightarrow u(i, j)$ for each (i, j) . Moreover in this way the parental relation is left unchanged for ξ_n and for ξ . Then define ξ_n by $\xi_n = Du_n$ according to (2.2). Obviously we have $\xi_n \rightarrow \xi$.

Moreover, for every (i, j) and every n the vectors $\nabla_{\xi} \mathcal{U}_{i,j}^{\xi_n}$ belongs to the superdifferential of the map $\xi \mapsto \mathcal{U}_{i,j}^{\xi}$ (since the function is concave and this vector is the gradient). Since the graph of the superdifferential is closed, the limit of this sequence of vectors must belong to the superdifferential at ξ . This limit actually exists and is given by $\nabla_{\xi} \mathcal{U}_{i,j}^{\xi}$, because of the continuity of the formulas that we used to compute all of these vectors. This is possible because the approximation ξ_n was chosen in order not to change the parental relations. \square

Each vector $\nabla_{\xi} \mathcal{U}_{i,j}$ stores at most N non-zero coefficients, so that the overall computation takes $O(N^2 \log(N))$ operations and has a space complexity of $O(N^2)$.

Figure 1 shows two examples of subgradients $\nabla_{\xi}\mathcal{U}_{i,j}$ computed with the Subgradient Marching algorithm. For the metric $\xi = 1$, the subgradient is concentrated closely along the geodesic, which is a straight line. The second example shows a configuration for which the subgradient is located around two geodesic curves.

4 Applications

This section describes some applications of the Subgradient Marching algorithm.

4.1 Landscape Design

The design of a landscape in a domain $\Omega \subset \mathbb{R}^d$ corresponds to the optimization of a metric $\xi(x)$ that describes locally the difficulty of passing through some point $x \in \Omega$. The criterion to optimize should take into account the geodesic distance $d_{\xi}(x_s, x_t)$ between landmark points $\{x_s\}_{k=0}^{p-1}$ as well as additional constraint on ξ .

Constrained distance maximization. A natural condition, originally investigated in [4], is that the landmarks should be maximally distant one from each others, according to the metric ξ . This corresponds to the maximization of

$$\mathcal{E}(\xi) = \sum_{s,t} w_{s,t} d_{\xi}(x_s, x_t), \quad (4.1)$$

where $w_{s,t} \geq 0$ are weights describing the interaction between the landmarks. This criterion models agents located at the points $\{x_s\}_s$ and that are free to modify the landscape to defend themselves optimally from the other agents.

The optimization of \mathcal{E} should be done under additional constraints on the set of admissible metrics, to avoid degenerate solutions. We consider here a local constraint

$$\forall i, j, \quad 0 < \underline{\xi}_{i,j} \leq \xi_{i,j} \leq \bar{\xi}_{i,j}, \quad (4.2)$$

that accounts for the maximal concentration of material allowed. We also consider a global constraint

$$\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \xi_{i,j} \leq \lambda, \quad (4.3)$$

where $|\Omega|$ is the number of grid points in Ω , that accounts for the total amount of ground material available. The constant λ satisfies necessarily

$$\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \underline{\xi}_{i,j} \leq \lambda \leq \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \bar{\xi}_{i,j}.$$

We note that the maximization of \mathcal{E} under the pointwise constraint (4.2) alone would be saturated everywhere $\xi_{i,j} = \bar{\xi}_{i,j}$.

Taking into account that the constraint (4.3) is obviously saturated, the landscape design problem is written as

$$\xi^* = \operatorname{argmax}_{\xi \in \mathcal{C}} \mathcal{E}(\xi) \quad \text{where} \quad \mathcal{C} = \left\{ \xi ; \underline{\xi}_{i,j} \leq \xi_{i,j} \leq \bar{\xi}_{i,j}, \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \xi_{i,j} = \lambda \right\}, \quad (4.4)$$

and we note that this maximum might be non-unique. A continuous formulation of the problem (4.4) is studied in [4] that proves existence of optimal solutions.

Projected subgradient descent. The problem (4.4) is the maximization of a non-smooth concave functional under convex constraints. It can be solved using a subgradient ascent, given an initial metric $\xi^{(0)}$,

$$\xi^{(k+1)} = \Pi_{\mathcal{C}}(\xi^{(k)} + \rho_k \nabla_{\xi^{(k)}} \mathcal{E}), \quad (4.5)$$

where $\rho_k > 0$ is a sequence of gradient step size and $\Pi_{\mathcal{C}}$ is the orthogonal projection on the set of constraints \mathcal{C} .

A subgradient of \mathcal{E} is obtained by combining a subgradient of the distance map $\mathcal{U}_s^\xi(x) = d_\xi(x_s, x)$ to each landmark x_s

$$\nabla_\xi \mathcal{E} = \sum_{s,t} w_{s,t} \nabla_\xi \mathcal{U}_s^\xi(x_t). \quad (4.6)$$

Each subgradient vector $\nabla_\xi \mathcal{U}_s^\xi(x_t)$ is computed by the subgradient descent algorithm, Table 2, starting the front propagation from the point x_s .

Thanks to the following proposition, the projection $\Pi_{\mathcal{C}}$ is easily computed. A simple dichotomy is used to find the value of α that satisfies (4.8)

Proposition 4.1. For $\underline{\xi} \leq \lambda \leq \bar{\xi}$, one has

$$\Pi_{\mathcal{C}}(\xi) = \mathcal{P}_{\underline{\xi}}^{\bar{\xi}}(\xi + \alpha), \quad \text{where} \quad \mathcal{P}_{\underline{\xi}}^{\bar{\xi}}(\xi)_{i,j} = \max(\min(\xi_{i,j}, \bar{\xi}_{i,j}), \underline{\xi}_{i,j}) \quad (4.7)$$

where $\alpha \in \mathbb{R}$ is such that

$$\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \Pi_{\mathcal{C}}(\xi + \alpha)_{i,j} = \lambda. \quad (4.8)$$

Proof. The projected metric $\Pi(\xi)$ satisfies

$$\Pi_{\mathcal{C}}(\xi) = \min_{\tilde{\xi} \in \mathcal{C}} \|\tilde{\xi} - \xi\|^2. \quad (4.9)$$

For a given Lagrange multiplier $\alpha \in \mathbb{R}$, this corresponds to minimize

$$\min_{\underline{\xi} \leq \tilde{\xi} \leq \bar{\xi}} \|\tilde{\xi} - \xi\|^2 - 2\alpha \sum_{(i,j) \in \Omega} \tilde{\xi}_{i,j}.$$

By developing the square terms, one sees that this amounts to project $\xi + \alpha$ on the set of constraints $\underline{\xi} \leq \tilde{\xi} \leq \bar{\xi}$. This is obtained by truncation with the operator $\mathcal{P}_{\underline{\xi}}^{\bar{\xi}}$, as written in (4.7).

Since the function

$$\alpha \mapsto \sum_{(i,j) \in \Omega} \mathcal{P}_{\underline{\xi}}^{\bar{\xi}}(\xi + \alpha)_{i,j}$$

is an increasing continuous function, there exists $\alpha \in \mathbb{R}$ satisfying conditions (4.8). \square

The following theorem ensures the convergence of the projected gradient descent.

Theorem 4.2. For $\rho_k = 1/k$, the sequence $(\xi^{(k)})_k$ converges to a maximizer ξ^* of (4.4).

Proof. As stated for instance in [2], the convergence of a subgradient descent is ensured if

$$\sum_k \rho_k = +\infty \quad \text{and} \quad \sum_k \rho_k^2 < +\infty.$$

and if the sequence $(\nabla_{\xi^{(k)}} \mathcal{E})_k$ stay bounded. Since for each landmarks (x_s, x_t) , the mapping $\xi \mapsto \mathcal{U}_s^\xi(x_t)$ is concave and 1-homogeneous, it is Lipschitz continuous and hence its subgradients are bounded. \square

Numerical examples. We first consider $p = 2$ agents located at two points x_0, x_1 in the corners of a square domain, as shows in Figure 5. The constraints are set to $\underline{\xi} = 0.1$, $\bar{\xi} = 1$ and $\lambda = 0.2|\Omega|$. The domain Ω is sampled on a square grid of 100×100 points. Since the landmarks are close to the boundary of Ω , hills appear between each x_i and the boundary. This phenomena is explained by the fact that is less costly to *build* these short hills and it makes a bypass behind the defender more difficult.

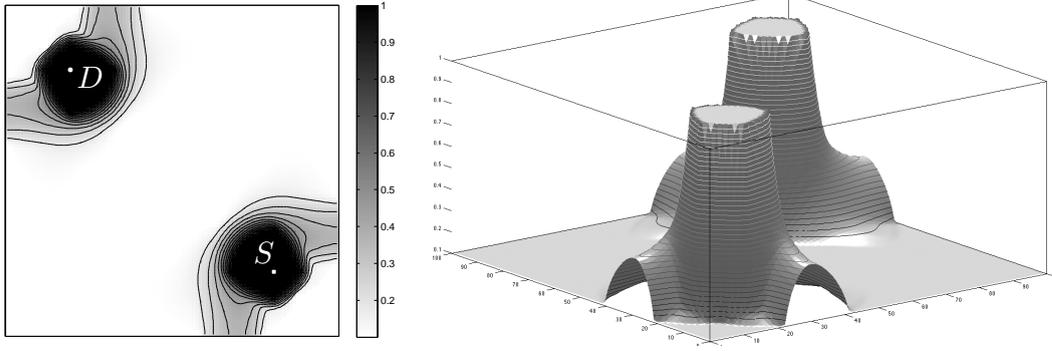


Figure 5: 2D and 3D display of the optimal metric ξ^* .

Figure 6, top, shows the decay of the error between the iterates $\xi^{(k)}$ and the optimal metric ξ^* . Figure 6, bottom, shows the increase of the energy, which is not strictly monotonic because of the non-smooth nature of the problem to maximize.

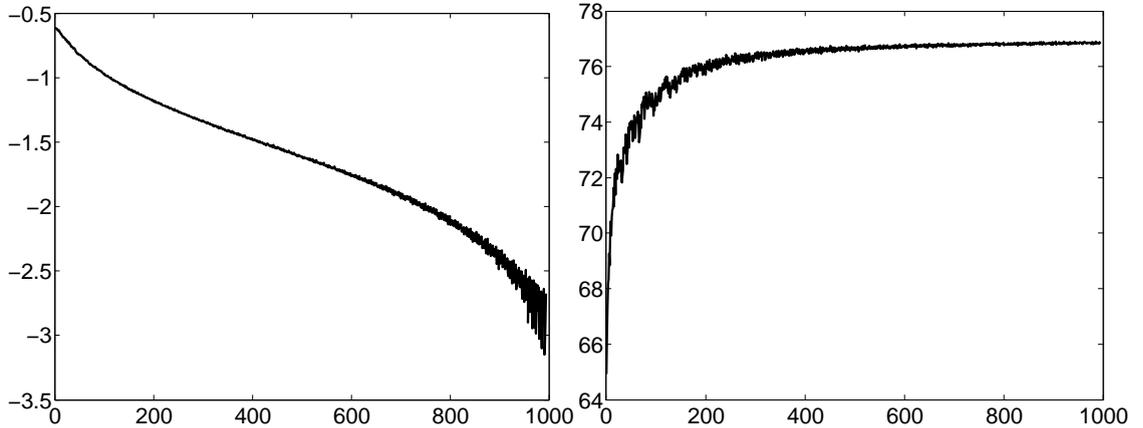


Figure 6: Left: decrease of the error $\log_{10}(\|\xi^{(k)} - \xi^*\|/\|\xi^*\|)$, right: increase of the energy $\mathcal{E}(\xi^{(k)})$.

Figure 7, left and middle, shows the influence of the total mass parameter λ . Decreasing the value of λ causes the optimal metric ξ^* to be more concentrated around the landmark positions on elongated regions. Decreasing the value of $\underline{\xi}$ toward 0 causes these regions on become circular, see Figure 7, right. One can note that the Gradient Marching algorithm is able to compute a subgradient of \mathcal{E} as soon as $\xi > 0$, but the convergence tends to become slower when $\underline{\xi}$ approaches 0.

One can indeed prove that if $\underline{\xi}_{i,j} = 0$ and $\bar{\xi}_{i,j} = c$, in a continuous setting, the unique optimal metric ξ^* is given by $\xi_{i,j} = c$ for (i, j) in two balls around x_0 and x_1 and $\xi_{i,j} = 0$

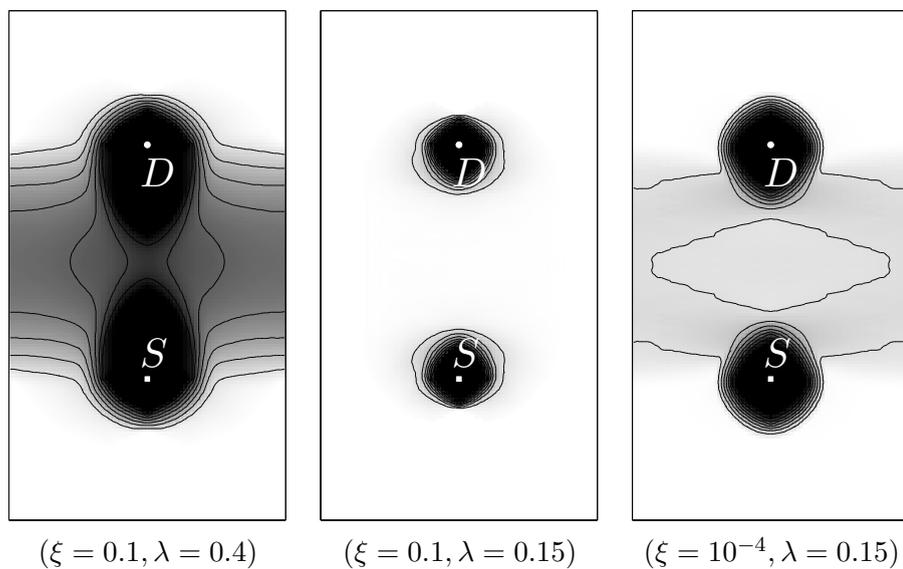


Figure 7: Dependence on parameters λ and $\underline{\xi}$ of the optimal metric ξ . In all examples $\bar{\xi} = 1$.

elsewhere, if λ is small enough so that the two balls fit inside Ω .

Figure 8, left, shows an example of spatially varying constraints. To prevent the agent located in x_1 to modify the metric, we enforce

$$\forall (i, j) \in \Omega_1, \quad \underline{\xi}_{i,j} = \bar{\xi}_{i,j} = 0.1,$$

where Ω_1 is a region surrounding x_1 , whereas outside Ω_1 we set

$$\forall (i, j) \notin \Omega_1, \quad \underline{\xi}_{i,j} = 0.1, \quad \bar{\xi}_{i,j} = 1.$$

The metric is thus only optimized in $\Omega \setminus \Omega_1$, as shown on Figure 5, left.

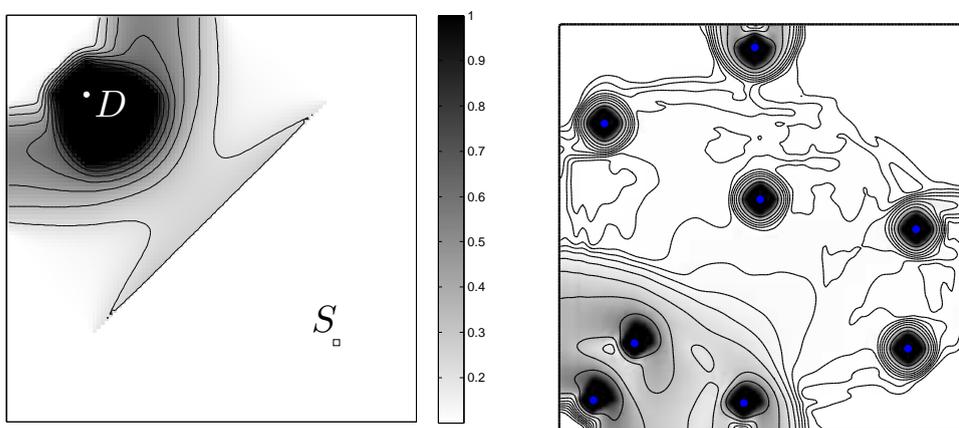


Figure 8: Left: spatially varying constraint $\bar{\xi}_{i,j}$ with $P = 2$ landmarks, right: constant constraint with $P = 8$ landmarks.

Figure 8, right, shows an example of optimal metric ξ^* computed with $P = 8$ landmarks. The weights between the landmarks are set to $w_{s,t} = 1$ and $\underline{\xi}_{i,j} = 0.1, \bar{\xi}_{i,j} = 6$.

Figure 9 shows the iterations of the algorithm for a domain Ω with a hole.

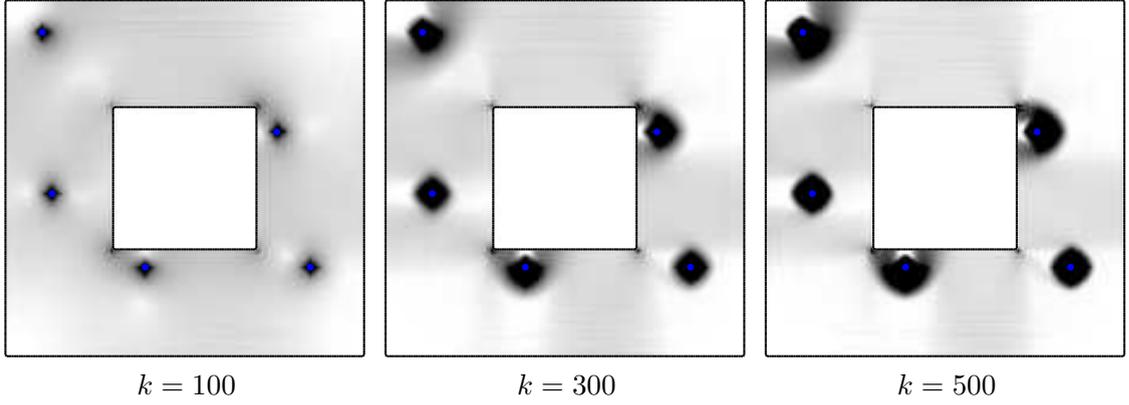


Figure 9: Iterations $\xi^{(k)}$ computed for a domain Ω with a hole and with $P = 5$ landmarks.

Extension of the model. It is possible to modify the energy \mathcal{E} defined in (4.1) to mix differently the distances between the points $\{x_s\}_s$. One can for instance maximize

$$\mathcal{E}_{\min}(\xi) = \sum_s \min_{t \neq s} d_\xi(x_s, x_t).$$

This functional is a minimum of concave functions, and hence \mathcal{E}_{\min} is a concave function. The maximization of the energy \mathcal{E}_{\min} forces each landmark to be maximally distant from its closest neighbors.

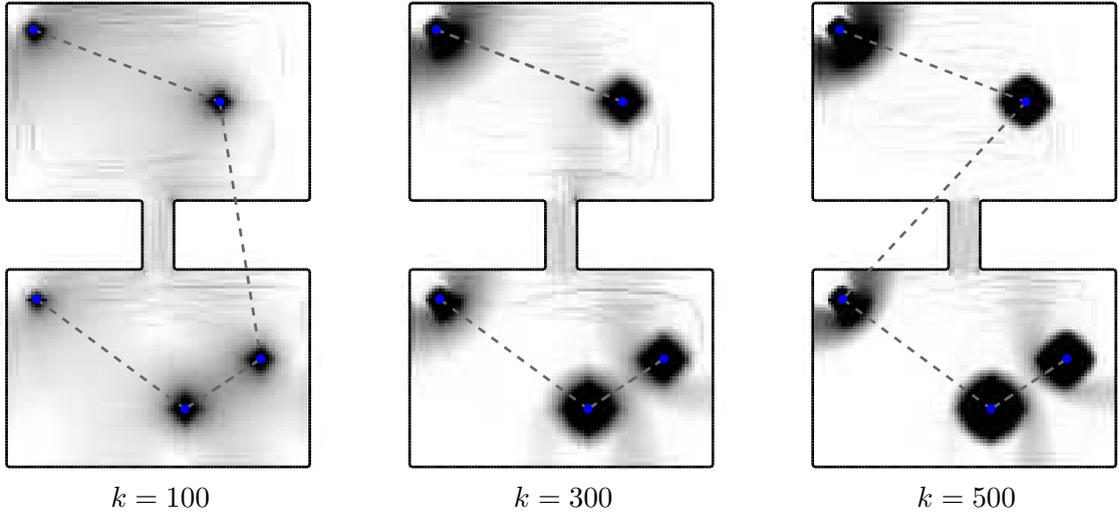


Figure 10: Iteration $\xi^{(k)}$ computed during the subgradient descent. The dashed line corresponds to the connexion $s \rightarrow t(s)$ of nearest neighbors points.

The subgradient of \mathcal{E}_{\min} is computed as

$$\nabla_\xi \mathcal{E} = \sum_s \nabla_\xi \mathcal{U}_s^\xi(x_{t(s)}).$$

where, for each landmark x_s , $x_{t(s)}$ is the closest landmark according to the metric ξ

$$t(s) = \underset{t \neq s}{\operatorname{argmin}} d_\xi(x_s, x_t).$$

A projected gradient descent similar to (4.5) converges to a maximum of \mathcal{E}_{\min} .

Figure 10, left and center, shows how the metric $\xi^{(k)}$ evolves during the iterations of a projected gradient descent. The graph connecting each x_s to its nearest neighbor $x_{t(s)}$ is overlaid. The points x_s are clustered on two sides of the domain, so that during the first iterations, the graph connects points on each side of the domain. At convergence, the metric is optimized so that points located in two different sides of the domain are also relatively close one from each other. This is why the graph also connects points located on two different sides.

4.2 Traffic Congestion Equilibria

The simulation of a static traffic congestion is the computation of a Riemannian metric ξ^* so that agents travel along geodesics between sources $\{x_s\}_{s=0}^{p-1}$ and targets $\{y_t\}_{t=0}^{q-1}$.

This metric ξ^* is an equilibrium when iterating between choosing geodesic paths for the agents, and updating the metric to penalize areas where many agents are travelling. Originally formulated on a graph, this *Wardrop equilibrium* is extended to a continuous setting as the solution of a variational problem involving geodesic distances, see [5]. A discretization of this continuous formulation reads

$$\xi^* = \underset{\xi \geq 0}{\operatorname{argmax}} \mathcal{E}_c(\xi) = \sum_{s,t} \gamma_{s,t} d_\xi(x_s, y_t) - \sum_{(i,j) \in \Omega} H^*(\xi_{i,j}) \quad (4.10)$$

where $\gamma_{s,t}$ is the strength of the traffic between source s and target t , and $H^*(x) = x^3/3$. See [1] for more details about this variational formulation and how H^* relates the metric ξ and the traffic intensity.

The variational problem (4.10) can be solved efficiently with Subgradient Marching descent

$$\xi^{(k+1)} = \max(0, \xi^{(k)} + \rho_k \delta_k),$$

where $\rho_k = 1/k$ are decreasing gradient step size and the subgradient \mathcal{E}_c is

$$\delta_k = -H^{*'}(\xi^{(k)}) + \sum_{s,t} \gamma_{s,t} \nabla_{\xi^{(k)}} \mathcal{U}_s(y_t), \quad (4.11)$$

where $H^{*'}$ is applied to each component $\xi_{i,j}^{(k)}$ of $\xi^{(k)}$. The subgradient $\nabla_{\xi^{(k)}} \mathcal{U}_s(x_t)$ at $\xi^{(k)}$ of the mapping $\xi \mapsto d_\xi(x_s, y_t)$ is computed with the Subgradient Marching algorithm, Table 2, starting the front propagation at the point x_s .

Numerical example. Figure 11 shows an example of congested metric with a complex domain Ω and multiple sources and targets. In a symmetric configuration of two sources S_1 and S_2 , and two targets T_1 and T_2 ; we consider a river where there is no traffic and a bridge linking the two sides of the river. We chose the traffic weights such that $\gamma_{1,1} + \gamma_{1,2} = 2(\gamma_{2,1} + \gamma_{2,2})$ and $\frac{\gamma_{2,2}}{\gamma_{2,1}} = \frac{\gamma_{1,1}}{\gamma_{1,2}} = 2$. The traffic intensity going out from S_1 is twice S_2 's. One can note the two hollows on each side of the river appearing because of the inter-sides and intra-sides crossed traffics.

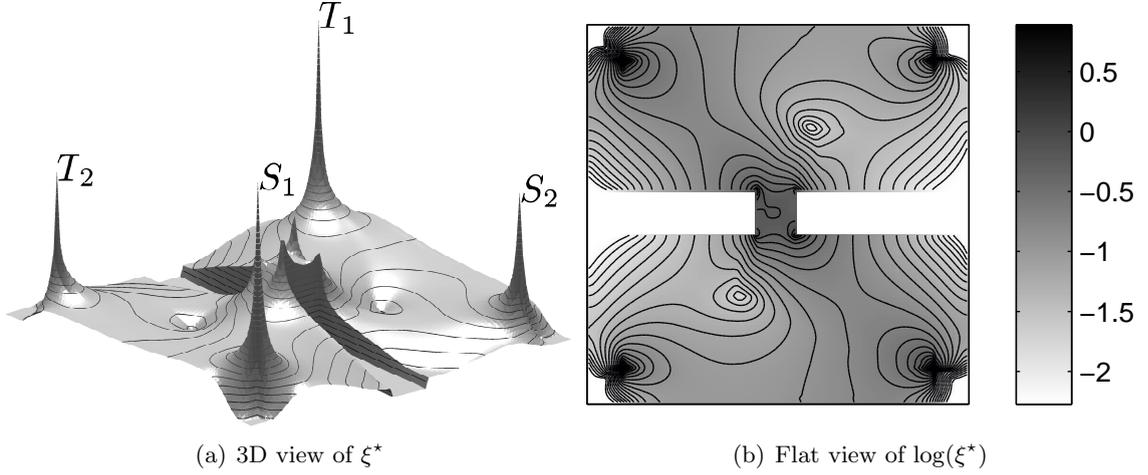


Figure 11: Two sources and two targets, with a river and a bridge on a symmetric configuration and an asymmetric traffic weights.

4.3 Travel Time Tomography

An simple model of seismic data acquisition assumes that geodesic distances are collected between sources x_s and sensors y_t

$$\forall s = 0, \dots, p-1, \forall t = 0, \dots, q-1, \quad d_{s,t} = d_{\xi^0}(x_s, y_t),$$

where ξ^0 is an unknown Riemannian metric that reflects the properties of the underground.

Geodesic tomography inversion. Recovering an approximation ξ^* of ξ^0 from the measures $d_{s,t}$ is an ill posed inverse problem. Such an approximation can be obtained by minimizing a functional that computes a tradeoff between fitting the geodesic measurements and imposing smoothness of the metric. Assuming a uniform smoothness of the metric leads to a Sobolev regularization

$$\min_{\xi \in \mathcal{C}} \mathcal{E}_{\text{tom}}(\xi) = \sum_{s,t} (d_{\xi}(x_s, x_t) - d_{s,t})^2 + \frac{\mu}{2} \sum_{(i,j) \in \Omega} \|\text{grad}_{i,j} \xi\|^2 \quad (4.12)$$

where the operator grad is a finite difference discretization of the 2D gradient

$$\text{grad}_{i,j} \xi = (\xi_{i+1,j} - \xi_{i,j}, \xi_{i,j+1} - \xi_{i,j}),$$

with Neumann condition on the boundary $\partial\Omega$ of the domain. The parameter μ controls the strength of the regularization and should be adapted to the number pq of measurements and the smoothness of ξ^0 . The condition $\xi \in \mathcal{C}$ allows one to incorporate additional constraints to enhance the quality of the recovery.

In the following, we use the local and global constraints \mathcal{C} defined in (4.4), where

$$\underline{\xi} = \min_{(i,j) \in \Omega} \xi_{i,j}^0, \quad \bar{\xi} = \max_{(i,j) \in \Omega} \xi_{i,j}^0, \quad \text{and} \quad \lambda = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \xi_{i,j}^0.$$

Subgradient descent recovery. The minimization problem (4.12) is non-convex, but a local minimizer ξ^* can be computed using a projected gradient descent. Starting from an initial metric $\xi^{(0)}$, the iteration reads

$$\xi^{(k+1)} = \Pi_{\mathcal{C}}(\xi^{(k)} - \rho_k(\delta_k - \mu\Delta\xi^{(k)})) \quad (4.13)$$

where $\rho_k = 1/k$ is a sequence of decreasing step sizes, $\Pi_{\mathcal{C}}$ is the orthogonal projection on \mathcal{C} computed as detailed in Theorem 4.1, $\Delta = -\text{grad}^* \circ \text{grad}$ is the Laplacian, and

$$\delta_k = \sum_{s,t} (d_{\xi}(x_s, x_t) - d_{s,t}) \nabla_{\xi^{(k)}} \mathcal{U}_s(y_t).$$

The subgradient $\nabla_{\xi^{(k)}} \mathcal{U}_s(y_t)$ at $\xi = \xi^{(k)}$ of the mapping $\xi \mapsto d_{\xi}(x_s, y_t)$ is computed with the Subgradient Marching algorithm, Table 2, starting the propagation from the point x_s .

The subgradient descent (4.13) converges to a local minimum ξ^* of the problem (4.12). This solution ξ^* is close to the metric ξ^0 to recover if ξ^0 is smooth enough, if the contrast $\bar{\xi}/\underline{\xi}$ is not too large, if the number of measurements pq is large enough and if the initialization $\xi^{(0)}$ is not too far from ξ^0 .

Numerical examples. Figure 12 shows two examples of smooth metrics ξ^0 recovered from travel time tomography measurements. In each case, we set $\bar{\xi}/\underline{\xi} = 1.3$ and we use $\xi_{i,j}^{(0)} = \lambda$ as an initial flat metric.

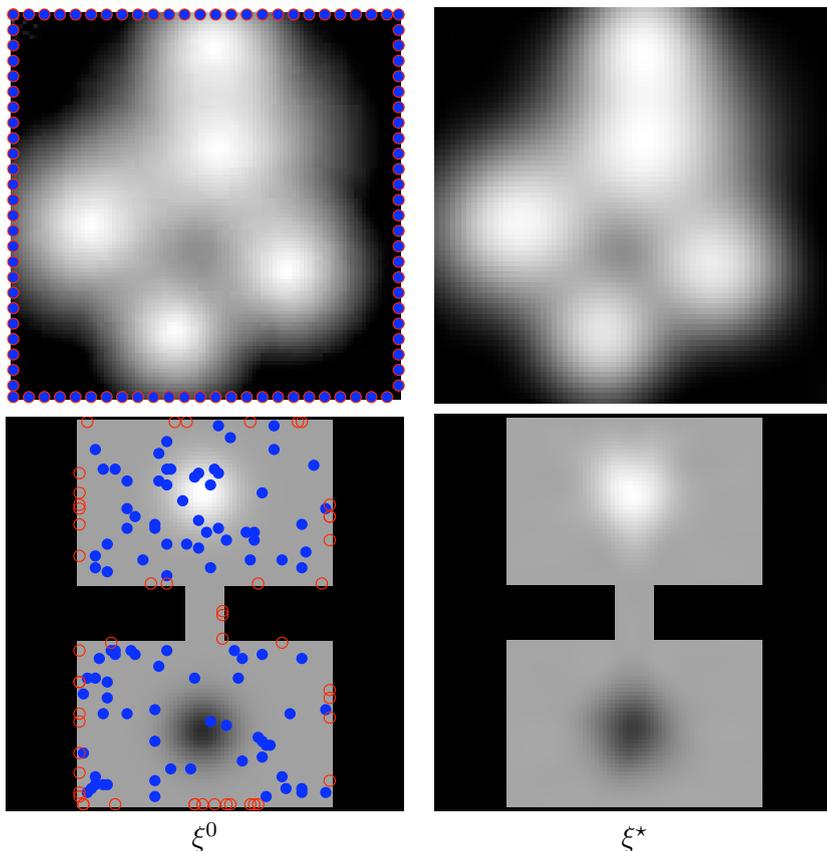


Figure 12: Examples of travel time tomography recovery.

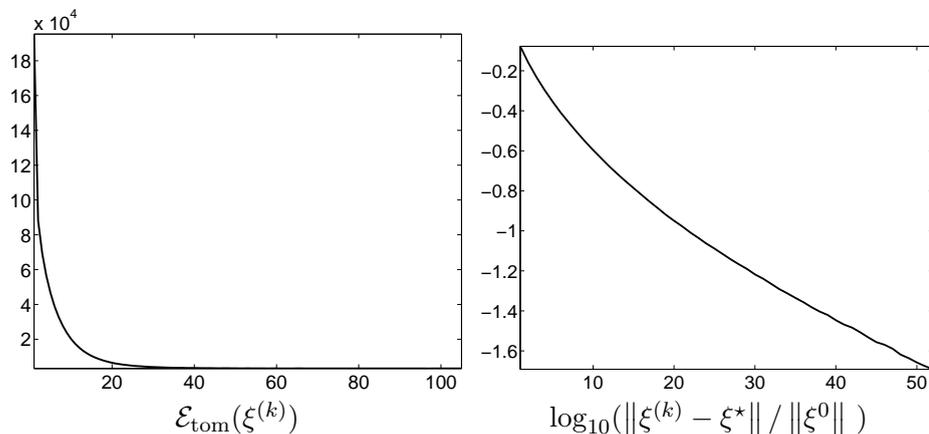


Figure 13: Decay of the energy and the error for the first example shown in Figure 12.

For the first example, we use $p = q = 100$ points distributed evenly on the boundary of a square Ω , discretized at $N = 150 \times 150$ points, and set $x_i = y_i$ so that sensor and sources are the same. For the second example, we use $p = 50$ source distributed evenly on the boundary of a complicated domain Ω , and $q = 150$ sensors distributed randomly within the domain. We enforce the smoothness of the solution by setting a large enough regularization parameter $\mu = 0.1$. Figure 13 shows the decay of the energy \mathcal{E}_{tom} and the error $\|\xi^{(k)} - \xi^*\|$.

The recovery error $\|\xi^0 - \xi^*\| / \|\xi^0\|$ is 2.5×10^{-2} for the first example and 7×10^{-3} for the second example. Both examples shows that for moderately complicated tomography problems (smooth medium and low contrast), a good approximation can be obtained by Subgradient Marching descent. These synthetic examples are however quite simple, and a detailed analysis of the method and the properties of the minimizer ξ^* is desirable.

Conclusion

We have presented a new Subgradient Marching algorithm to compute the derivative of a discrete geodesic distance with respect to the metric. Up to our knowledge, this is the first time that a consistent numerical tool has been introduced to solve variational problems that take into account geodesic distances between points. Three representative applications illustrate the practical use of Subgradient Marching. Landscape design and traffic congestion lead to the maximization of a concave functional, and can be solved efficiently with a projected gradient descent. Recovery of geodesic inverse problems such as travel time tomography is obtained by computing a local minimizer of a non-convex problem.

References

- [1] F. Benmansour, G. Carlier, G. Peyré, F. Santambrogio, Numerical Approximation of Continuous Traffic Congestion Equilibria, preprint.
- [2] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal and C. Sagastizábal, *Numerical Optimization*, 2nd Edition, Springer-Verlag, Heidelberg, 2006.

- [3] A. M. Bronstein, M. M. Bronstein, R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS)*, **103**(5):1168–1172, 2006.
- [4] G. Buttazzo, A. Davini, I. Fragalà and F. Maciá, Optimal Riemannian distances preventing mass transfer, *J. Reine Angew. Math.*, 575: 157–171, 2004.
- [5] G. Carlier, C. Jimenez , F. Santambrogio, Optimal transportation with traffic congestion and Wardrop equilibria, *SIAM Journal on Control and Opt.* **47** (3): 1330–1350, 2008.
- [6] M. Cavalca and P. Lailly, Accounting for the definition domain of the forward map in traveltime tomography—application to the inversion of prismatic reflections, *Inverse problems*, **23** (1), p. 139 (2007)
- [7] M. G. Crandall and P.-L. Lions, Two approximations of solutions of Hamilton-Jacobi equations, *Math. Comp.*, **43**: 1–19, 1984.
- [8] L.D. Cohen and R. Kimmel. Global Minimum for Active Contour Models: A Minimal Path Approach. *International Journal of Computer Vision*, **24**(1):57–78 , 1997.
- [9] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, **1**:269–271, 1959.
- [10] P.-L. Lions. Generalized Solutions of Hamilton-Jacobi Equations, *Pitman*. 1982.
- [11] S. Leung and J. Qian, An Adjoint State Method For Three-dimensional Transmission Traveltime Tomography Using First-Arrivals *Commun. Math. Sci.* **4** (1), 249-266, 2006.
- [12] E. Rouy and A. Tourin. A viscosity solution approach to shape from shading. *SIAM Journal on Numerical Analysis*, **29**:867–884, 1992.
- [13] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [14] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, **40**:1528–1538, 1995.
- [15] J.G. Wardrop, Some theoretical aspects of road traffic research, *Proc. Inst. Civ. Eng.*, **2**, no.2, 325-378, 1952.