The Riemannian Difference of Convex Algorithm in Manopt.jl

Ronny Bergmann

joint work with
O. P. Ferreira, E. M. Santos, and J. C. O. Souza.

Optimization Techniques in Quantum Chemistry

Aachen, June 5, 2024.

NTNU
Norges teknisk-naturvitenskapelige universitet

Difference of Convex

We aim to solve

$$\operatorname*{arg\,min}_{p \in \mathcal{M}} f(p)$$

where

- $\mathcal{M}$ is a Riemannian manifold
- $f\colon \mathcal{M} \to \mathbb{R}$ is a difference of convex function, i. e. of the form

$$f(p) = g(p) - h(p)$$

- $g, h\colon \mathcal{M} \to \overline{\mathbb{R}}$ are convex, lower semicontinuous, and proper

A Riemannian Manifold $\mathcal{M}$

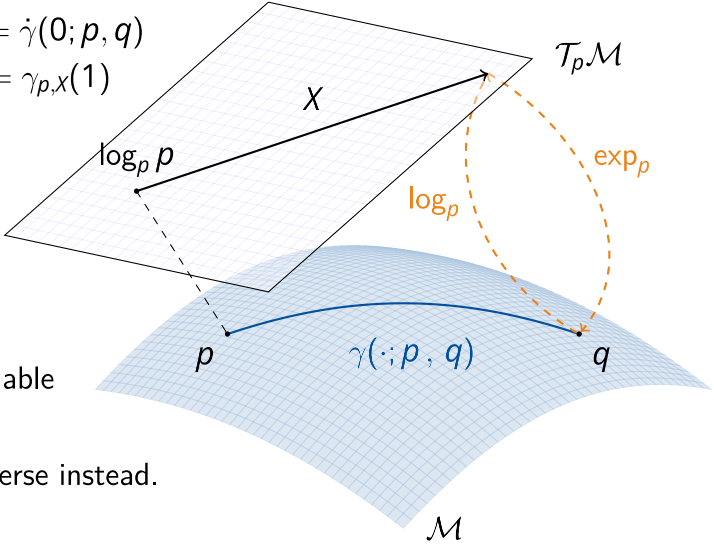
A d-dimensional Riemannian manifold can be informally defined as a set $\mathcal{M}$ covered with a "suitable" collection of charts, that identify subsets of $\mathcal{M}$ with open subsets of $\mathbb{R}^d$ and a continuously varying inner product on the tangent spaces.

[Absil, Mahony, and Sepulchre 2008]

A Riemannian Manifold $\mathcal{M}$

Notation.

- Logarithmic map $\log_p q = \dot\gamma(0; p, q)$
- Exponential map $\exp_p X = \gamma_{p,X}(1)$
- Geodesic $\gamma(\cdot; p\,,\, q)$
- Tangent space $\mathcal{T}_p\mathcal{M}$
- inner product $(\cdot\,,\,\cdot)_p$



Numerics.
$\exp_p$ and $\log_p$ maybe not available efficiently/ in closed form

$\Rightarrow$ use a retraction and its inverse instead.

(Geodesic) Convexity

[Sakai 1996; Udriște 1994]

A set $\mathcal{C} \subset \mathcal{M}$ is called (strongly geodesically) convex
if for all $p, q \in \mathcal{C}$ the geodesic $\gamma(\cdot; p, q)$ is unique and lies in $\mathcal{C}$.

A function $f\colon \mathcal{C} \to \overline{\mathbb{R}}$ is called (geodesically) convex
if for all $p, q \in \mathcal{C}$ the composition $f(\gamma(t; p, q)), t \in [0, 1]$, is convex.

The Riemannian Subdifferential

Let $\mathcal{C}$ be a convex set.

The subdifferential of f at $p \in \mathcal{C}$ is given by [O. Ferreira and Oliveira 2002; Lee 2003; Udriște 1994]

$$\partial_{\mathcal{M}} f(p) := \bigl\{ \xi \in \mathcal{T}_p^*\mathcal{M} \,\big|\, f(q) \geq f(p) + \langle \xi \,, \log_p q \rangle_p \ \text{ for } q \in \mathcal{C} \bigr\},$$

where

- $\mathcal{T}_p^*\mathcal{M}$ is the dual space of $\mathcal{T}_p\mathcal{M}$, also called cotangent space
- $\langle \cdot \,, \cdot \rangle_p$ denotes the duality pairing on $\mathcal{T}_p^*\mathcal{M} \times \mathcal{T}_p\mathcal{M}$

The Fenchel Conjugate

The Fenchel conjugate of a function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ is given by

$$f^*(\xi) := \sup_{x\in\mathbb{R}^n} \langle \xi, x\rangle - f(x) = \sup_{x\in\mathbb{R}^n} \begin{pmatrix} \xi \\ -1 \end{pmatrix}^{\mathsf{T}} \begin{pmatrix} x \\ f(x) \end{pmatrix}$$

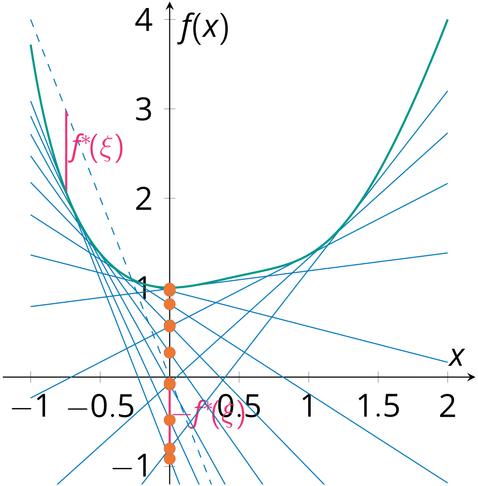
- given $\xi \in \mathbb{R}^n$: maximize the distance between $\xi^{\mathsf{T}}\cdot$ and f
- can also be written in the epigraph

The Fenchel biconjugate reads

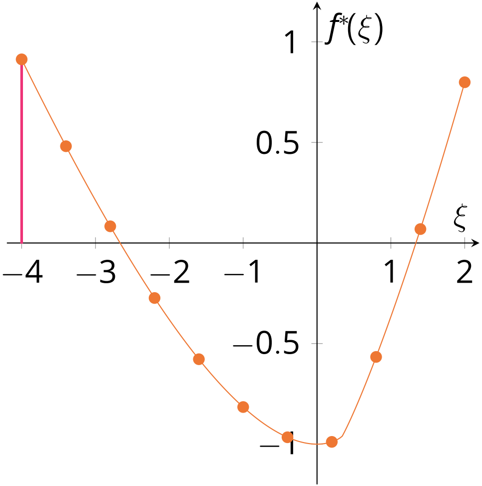
$$f^{**}(x) = (f^*)^*(x) = \sup_{\xi\in\mathbb{R}^n} \langle \xi, x\rangle - f^*(\xi).$$

Illustration of the Fenchel Conjugate

The function f



The Fenchel conjugate f^*



The Riemannian Difference of Convex Algorithm

The Euclidean DCA

Idea 1. At x_k, approximate $h(x)$ by its affine minorization

$$h_k(x) := h(x^{(k)}) + \langle x - x^{(k)}, y^{(k)} \rangle \text{ for some } y^{(k)} \in \partial h(x^k)$$

$\Rightarrow$ iteratively minimize $g(x) - h_k(x) = g(x) - h(x^{(k)}) - \langle x - x^{(k)}, y^{(k)} \rangle$

Idea 2. Using duality theory finding a new $y^{(k)} \in \partial h(x^{(k)})$ is equivalent to

$$y^{(k)} \in \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \Bigl\{ h^*(y) - g^*(y^{(k-1)}) - \langle y - y^{(k-1)}, x^{(k)} \rangle \Bigr\}$$

Idea 3. Reformulate 2 using a proximal map $\Rightarrow$ DCPPA
on manifolds this was done in [Almeida, Neto, Oliveira, and Souza 2020; Souza and Oliveira 2015]

In the Euclidean case, all three models are equivalent.

A Fenchel Duality on a Hadamard Manifold

Let

- $\mathcal{T}\mathcal{M} = \dot{\bigcup}_p T_p\mathcal{M}$ denote the tangent bundle
- analogously $T^*\mathcal{M}$ denotes the cotangent bundle
- $\mathcal{M}$ be a Hadamard manifold (non-positive sectional curvature).

Definition [Silva Louzeiro, RB, and Herzog 2022]

Let $f\colon \mathcal{M} \to \overline{\mathbb{R}}$.
The Fenchel conjugate of f is the function $f^*\colon \mathcal{T}^*\mathcal{M} \to \overline{\mathbb{R}}$ defined by

$$f^*(p,\xi) \coloneqq \sup_{q\in\mathcal{M}} \Bigl\{ \langle \xi, \log_p q \rangle - f(q) \Bigr\}, \qquad (p,\xi) \in \mathcal{T}^*\mathcal{M}.$$

The Dual Difference of Convex Problem

Given the Difference of Convex problem

$$\operatorname*{arg\,min}_{p \in \mathcal{M}} g(p) - h(p)$$

and the Fenchel duals g^* and h^*,
we can state the dual difference of convex problem as [RB, O. P. Ferreira, Santos, and Souza 2024]

$$\operatorname*{arg\,min}_{(p,\xi) \in T^*\mathcal{M}} \; h^*(p, \xi) - g^*(p, \xi).$$

On $\mathcal{M} = \mathbb{R}^n$ this indeed simplifies to the classical dual problem.

Theorem. [RB, O. P. Ferreira, Santos, and Souza 2024]

$$\inf_{(q,X) \in \mathcal{T}^*\mathcal{M}} \Big\{ h^*(q, X) - g^*(q, X) \Big\} = \inf_{p \in \mathcal{M}} \{ g(p) - h(p) \}.$$

The Dual Difference of Convex Problem

The primal and dual Difference of Convex problem

$$\operatorname*{arg\,min}_{p\in\mathcal{M}} g(p) - h(p) \qquad \text{and} \qquad \operatorname*{arg\,min}_{(p,\xi)\in T^*\mathcal{M}} h^*(p,\xi) - g^*(p,\xi)$$

are equivalent in the following sense.

Theorem. [RB, O. P. Ferreira, Santos, and Souza 2024]

If p^* is a solution of the primal problem, then $(p^*,\xi^*) \in T^*\mathcal{M}$ is a solution for the dual problem for all $\xi^* \in \partial_{\mathcal{M}} h(p^*) \cap \partial_{\mathcal{M}} g(p^*)$.

If $(p^*,\xi^*) \in T^*\mathcal{M}$ is a solution of the dual problem for some $\xi^* \in \partial_{\mathcal{M}} h(p^*) \cap \partial_{\mathcal{M}} g(p^*)$, then p^* is a solution of the primal problem.

NTNU

Derivation of the Riemannian DCA

We consider the first order Taylor approximation of h at some point $p^{(k)}$:
With $\xi \in \partial h(p^{(k)})$ we set

$$h_k(p) := h(p^{(k)}) + \langle \xi \,, \log_{p^{(k)}} p \rangle_{p^{(k)}}$$

Using musical isomorphisms we identify $X = \xi^\sharp \in T_p\mathcal{M}$,
where we call X a subgradient. Locally h_k minorizes h, i. e.

$$h_k(q) \leq h(q) \quad \text{locally around} \quad p^{(k)}$$

$\Rightarrow$ Use $-h_k(p)$ as upper bound for $-h(p)$ in $f = g - h$.

Note. On $\mathbb{R}^n$ the function h_k is linear.
On a manifold h_k is nonlinear and not even necessarily convex, even on a Hadamard manifold.

The Riemannian DC Algorithm

[RB, O. P. Ferreira, Santos, and Souza 2024]

Input: An initial point $p^{(0)} \in \operatorname{dom}(g)$, g and $\partial_{\mathcal{M}} h$

1: Set $k = 0$.
2: **while** not converged **do**
3: Take $X^{(k)} \in \partial_{\mathcal{M}} h(p^{(k)})$
4: Compute the next iterate $p^{(k+1)}$ as

$$p^{(k+1)} \in \operatorname*{arg\,min}_{p \in \mathcal{M}} g(p) - \bigl(X^{(k)} , \log_{p^{(k)}} p \bigr)_{p^{(k)}}. \qquad (*)$$

5: Set $k \leftarrow k + 1$
6: **end while**

Note. In general the subproblem $(*)$ can not be solved in closed form. But an approximate solution yields a good candidate.

For example: Given g, $p^{(k)}$, and $X^{(k)}$ and $\operatorname{grad} g \Rightarrow$ Gradient descent.

Convergence of the Riemannian DCA

Let $\{p^{(k)}\}_{k\in\mathbb{N}}$ and $\{X^{(k)}\}_{k\in\mathbb{N}}$ be the iterates and subgradients of the RDCA.

Theorem. [RB, O. P. Ferreira, Santos, and Souza 2024]

If $\bar{p}$ is a cluster point of $\{p^{(k)}\}_{k\in\mathbb{N}}$, then $\bar{p} \in \operatorname{dom}(g)$ and there exists a cluster point $\bar{X}$ of $\{X^{(k)}\}_{k\in\mathbb{N}}$ s. t. $\bar{X} \in \partial g(\bar{p}) \cap \partial h(\bar{p})$.

$\Rightarrow$ Every cluster point of $\{p^{(k)}\}_{k\in\mathbb{N}}$, if any, is a critical point of f.

Proposition. [RB, O. P. Ferreira, Santos, and Souza 2024]

Let g be σ-strongly (geodesically) convex. Then

$$f(p^{(k+1)}) \leq f(p^{(k)}) - \frac{\sigma}{2} d^2(p^{(k)}, p^{(k+1)})$$

and $\displaystyle\sum_{k=0}^{\infty} d^2(p^{(k)}, p^{(k+1)}) < \infty$, so in particular $\displaystyle\lim_{k\to\infty} d(p^{(k)}, p^{(k+1)}) = 0$.

Software

Manifolds.jl & Manopt.jl – Why Julia?

Goals.

- abstract definition of manifolds

$\Rightarrow$ implement abstract solvers on a generic manifold

- well-documented and well-tested
- fast.

$\Rightarrow$ "Run your favourite solver on your favourite manifold".

Why Julia?

julialang.org

- high-level language, properly typed
- multiple dispatch (cf. `f(x)`, `f(x::Number)`, `f(x::Int)`)
- just-in-time compilation, solves two-language problem
  $\Rightarrow$ "nice to write" and as fast as C/C++
- I like the community

ManifoldsBase.jl

NTNU

[Axen, Baran, RB, and Rzecki 2023]

Goal. Provide an interface to implement and use Riemannian manifolds.

Interface `AbstractManifold` to model manifolds

Functions like `exp(M, p, X)`, `log(M, p, X)` or `retract(M, p, X, method)`.

Decorators for implicit or explicit specification of an embedding, a metric, or a group,

Efficiency by providing in-place variants like `exp!(M, q, p, X)`

Manifolds.jl

Goal. Provide a library of Riemannian manifolds, that is efficiently implemented and well-documented

[Axen, Baran, RB, and Rzecki 2023]

Meta. generic implementations for $\mathcal{M}^{n\times m}$, $\mathcal{M}_1 \times \mathcal{M}_2$, vector- and tangent-bundles, esp. $T_p\mathcal{M}$, or Lie groups

Library. Implemented functions for

- Circle, Sphere, Torus, Hyperbolic, Projective Spaces, Hamiltonian
- (generalized, symplectic) Stiefel, Rotations
- (generalized, symplectic) Grassmann, fixed rank matrices
- Symmetric Positive Definite matrices, with fixed determinant
- (several) Multinomial, (skew-)symmetric, and symplectic matrices
- Tucker & Oblique manifold, Kendall's Shape space
- probability simplex, orthogonal and unitary matrices, Rotations, …

Concrete Manifold Examples.

Before first run `] add Manifolds` to install the package.

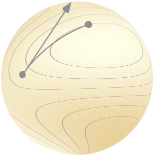
Load packages with `using Manifolds` and

- Euclidean space `M1 = ℝ^3` and 2-sphere `M2 = Sphere(2)`
- their product manifold `M3 = M1 × M2`
- A signal of rotations `M4 = SpecialOrthogonal(3)^10`
- SPDs `M5 = SymmetricPositiveDefinite(3)` (affine invariant metric)
- a different metric `M6 = MetricManifold(M5, LogCholeskyMetric())`

Then for any of these

- Generate a point `p=rand(M)` and a vector `X = rand(M; vector_at=p)`
- and for example `exp(M, p, X)`, or in-place `exp!(M, q, p, X)`

Manopt.jl



Goal. Provide optimization algorithms on Riemannian manifolds.

Features. Given a `Problem p` and a `SolverState s`,
implement `initialize_solver!(p, s)` and `step_solver!(p, s, i)`
$\Rightarrow$ an algorithm in the `Manopt.jl` interface

Highlevel interfaces like `gradient_descent(M, f, grad_f)`
on any manifold `M` from `Manifolds.jl`.

All provide debug output, recording, cache & counting capabilities,
as well as a library of step sizes and stopping criteria.

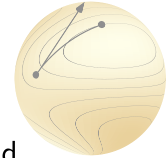
Manopt family.

manoptjl.org [RB 2022]

manopt.org [Boumal, Mishra, Absil, and Sepulchre 2014]

pymanopt.org [Townsend, Koep, and Weichwald 2016]

List of Algorithms in Manopt.jl



Derivatve Free Nelder-Mead, Particle Swarm, CMA-ES

Subgradient-based Subgradient Method, Convex Bundle Method, Proximal Bundle Method

Gradient-based Gradient Descent, Conjugate Gradient, Stochastic, Momentum, Nesterov, Averaged, …
Quasi-Newton with (L-)BFGS, DFP, Broyden, SR1,…
Levenberg-Marquard

Hessian-based Trust Regions, Adaptive Regularized Cubics (ARC)

nonsmooth Chambolle-Pock, Douglas-Rachford, Cyclic Proximal Point

constrained Augmented Lagrangian, Exact Penalty, Frank-Wolfe

nonconvex Difference of Convex Algorithm, DCPPA

manoptjl.org/stable/solvers/

Illustrating a few Keyword Arguments

Given cost `f(M,p)` and gradient `grad_f(M,p)`, a manifold `M` and a start point `p0`.

- `q = gradient_descent(M, f, grad_f, p0)` to perform gradient descent
- With Euclidean cost `f(E,p)` and gradient `∇f(E, p)`, use for conversion

  ```
  q = gradient_descent(M, f, ∇f, p0; objective_type=:Euclidean)
  ```

- print iteration number, cost and change every 10th iterate

  ```
  q = gradient_descent(M, f, grad_f, p0;
          debug=[:Iteration, :Cost, :Change, 10, "\n"]
      )
  ```

- record `reocord=[:Iterate, :Cost, :Change], return_state=true`
  Access: `get_solver_result(q)` and `get_record(s)`
- modify stop: `stopping_criterion = StopAfterIteration(100)`
- cache calls `cache=(:LRU, [:Cost, :Gradient], 25)` (uses `LRUCache.jl`)
- count calls `count=[:Cost, :Gradient]` (prints with `return_state=true`)

The Difference of Convex Algorithm in Manopt.jl

The algorithm is implemented and released in Julia using `Manopt.jl`[1].
It can be used with any manifold from `Manifolds.jl`

A solver call looks like

```
q = difference_of_convex_algorithm(M, f, g, ∂h, p0)
```

where one has to implement `f(M, p)`, `g(M, p)`, and `∂h(M, p)`.

- a sub problem is generated if keyword `grad_g=` is set
- an efficient version of its cost and gradient is provided
- you can specify the sub-solver using `sub_state=`
  to also set up the specific parameters of your favourite algorithm

[1] see https://manoptjl.org/stable/solvers/difference_of_convex/

A Numerical Example

Rosenbrock and First Order Methods

Problem. We consider the classical Rosenbrock example[2]

$$\operatorname*{arg\,min}_{x \in \mathbb{R}^2} a\bigl(x_1^2 - x_2\bigr)^2 + \bigl(x_1 - b\bigr)^2,$$

where $a, b > 0$, usually $b = 1$ and $a \gg b$, here: $a = 2 \cdot 10^5$.

Known Minimizer $x^* = \begin{pmatrix} b \\ b^2 \end{pmatrix}$ with cost $f(x^*) = 0$.

Goal. Compare first-order methods, e. g. using the (Euclidean) gradient

$$\nabla f(x) = \begin{pmatrix} 4a(x_1^2 - x_2) \\ -2a(x_1^2 - x_2) \end{pmatrix} + \begin{pmatrix} 2(x_1 - b) \\ 0 \end{pmatrix}$$

[2] available online in ManoptExamples.jl

A “Rosenbrock-Metric” on $\mathbb{R}^2$

In our Riemannian framework, we can introduce a new metric on $\mathbb{R}^2$ as

$$G_p := \begin{pmatrix} 1+4p_1^2 & -2p_1 \\ -2p_1 & 1 \end{pmatrix}, \text{ with inverse } G_p^{-1} = \begin{pmatrix} 1 & 2p_1 \\ 2p_1 & 1+4p_1^2 \end{pmatrix}.$$

We obtain $(X, Y)_p = X^{\mathrm{T}} G_p Y$

The exponential and logarithmic map are given as

$$\exp_p(X) = \begin{pmatrix} p_1 + X_1 \\ p_2 + X_2 + X_1^2 \end{pmatrix}, \qquad \log_p(q) = \begin{pmatrix} q_1 - p_1 \\ q_2 - p_2 - (q_1 - p_1)^2 \end{pmatrix}.$$

`Manifolds.jl`:
Implement these functions on `MetricManifold(ℝ^2, RosenbrockMetric())`.

The Riemannian Gradient w.r.t. the new Metric

Let $f\colon \mathcal{M} \to \mathbb{R}$. Given the Euclidean gradient $\nabla f(p)$, its Riemannian gradient $\operatorname{grad} f\colon \mathcal{M} \to T\mathcal{M}$ is given by

$$\operatorname{grad} f(p) = G_p^{-1} \nabla f(p).$$

While we could implement this denoting $\nabla f(p) = \begin{pmatrix} f_1'(p) & f_2'(p) \end{pmatrix}^{\mathrm{T}}$ using

$$\Bigl\langle \operatorname{grad} f(q), \log_q p \Bigr\rangle_q = (p_1 - q_1) f_1'(q) + (p_2 - q_2 - (p_1 - q_1)^2) f_2'(q),$$

but it is automatically done in `Manopt.jl`.

The Experiment Setup

Algorithms. We now compare

1. The Euclidean gradient descent algorithm on $\mathbb{R}^2$,
2. The Riemannian gradient descent algorithm on $\mathcal{M}$,
3. The Difference of Convex Algorithm on $\mathbb{R}^2$,
4. The Difference of Convex Algorithm on $\mathcal{M}$.

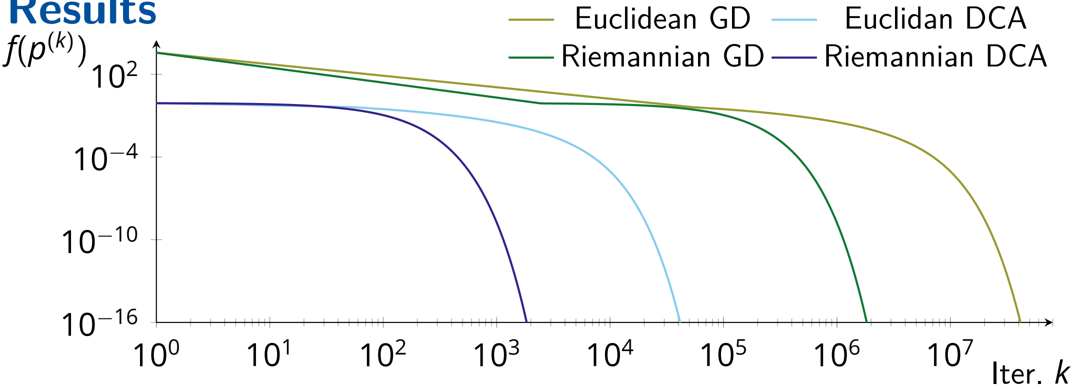
For DCA third we split f into $f(x) = g(x) - h(x)$ with

$$g(x) = a\left(x_1^2 - x_2\right)^2 + 2\left(x_1 - b\right)^2 \quad \text{and} \quad h(x) = \left(x_1 - b\right)^2.$$

Initial point. $p_0 = \frac{1}{10}\begin{pmatrix}1\\2\end{pmatrix}$ with cost $f(p_0) \approx 7220.81$.

Stopping Criterion.
$d_{\mathcal{M}}(p^{(k)}, p^{(k-1)}) < 10^{-16}$ or $\lVert \operatorname{grad} f(p^{(k)}) \rVert_p < 10^{-16}$.

Results



Algorithm	**Runtime** (sec.)	**# Iterations**
Euclidean GD	305.567	53 073 227
Euclidean DCA	58.268	50 588
Riemannian GD	18.894	2 454 017
Riemannian DCA	7.704	2 459

Summary

- Nonsmooth, nonconvex problems on manifold: difference of convex

$$\operatorname*{arg\,min}_{p\in\mathcal{M}} g(p) - h(p)$$

- The Difference of Convex Algorithm
- Relation to Fenchel Duality on Hadamard manifolds
- Convergence on Hadamard manifolds
- `Manifolds.jl` and `Manopt.jl`
- Numerically solve optimization problems on Riemannian manifolds

Outlook.

- couple `Manopt.jl` with (Euclidean) AD tools using `ManifoldDiff.jl`
- What is (Fenchel) duality on manifolds?

Selected References

Almeida, Y. T., J. X. d. C. Neto, P. R. Oliveira, and J. C. d. O. Souza (2020). "A modified proximal point method for DC functions on Hadamard manifolds". In: *Computational Optimization and Applications* 76.3, pp. 649–673. DOI: 10.1007/s10589-020-00173-3.

Axen, S. D., M. Baran, RB, and K. Rzecki (2023). "Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds". In: *ACM Transactions on Mathematical Software*. Accepted for pulication. DOI: 10.1145/3618296. arXiv: 2106.08777.

RB (2022). "Manopt.jl: Optimization on Manifolds in Julia". In: *Journal of Open Source Software* 7.70, p. 3866. DOI: 10.21105/joss.03866.

RB, O. P. Ferreira, E. M. Santos, and J. C. d. O. Souza (2024). "The difference of convex algorithm on Hadamard manifolds". In: *Journal of Optimization Theory and Applications*. DOI: 10.1007/s10957-024-02392-8. arXiv: 2112.05250.

Silva Louzeiro, M., RB, and R. Herzog (2022). "Fenchel Duality and a Separation Theorem on Hadamard Manifolds". In: *SIAM Journal on Optimization* 32.2, pp. 854–873. DOI: 10.1137/21M1400699. arXiv: 2102.11155.

Souza, J. C. d. O. and P. R. Oliveira (2015). "A proximal point algorithm for DC fuctions on Hadamard manifolds". In: *Journal of Global Optimization* 63.4, pp. 797–810. DOI: 10.1007/s10898-015-0282-7.

Interested in Numerical Differential Geometry? Join numdiffgeo.zulipchat.com!

ronnybergmann.net/talks/2024-Aachen-Difference-of-Convex-Manopt.pdf