Optimization of functions defined over sets of points in polygons with evolutionary algorithms based on Wasserstein barycenters

Babacar SOW (EMSE), Rodolphe LE RICHE (CNRS/LIMOS), Julien PELAMATTI (EDF R&D), Merlin KELLER (EDF R&D), Sanaa ZANNANE (EDF R&D)

Project: ANR SAMOURAI

University: Ecole Des Mines de Saint-Etienne

Table of contents

Example of an industrial problem: optimization of a wind-farm layout

A set of points model

- **Each point** (vector) represents the positions of a turbine.
- **The set of points** corresponds to the positions of all the turbines.
- Find an **optimal layout** of turbines minimizing the wake effects.

Context and problem formulation

Optimization of functions defined over *clouds* (sets) of points

- Dealing with functions assumed to be black-box
- We consider functions having inputs in the form of **bags of vectors** (or point clouds).
- These types of functions are encountered in many domains, such as: **image processing**, **design of experiments optimization**, ...

Variable of interest

- $\mathcal{X}$: space of all sets of n unordered points $\{x_1, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ and $n_{\min} \leq n \leq n_{\max}$.
- $X \in \mathcal{X}$ is a set of points and will be referred to as a **cloud of points**.

Approach

- Computationally cheap case: evolutionary algorithm

Mixed aspect: no order and varying size

Comparing two clouds of points with different sizes

The functions of interest are **permutation-invariant** with respect to their inputs.

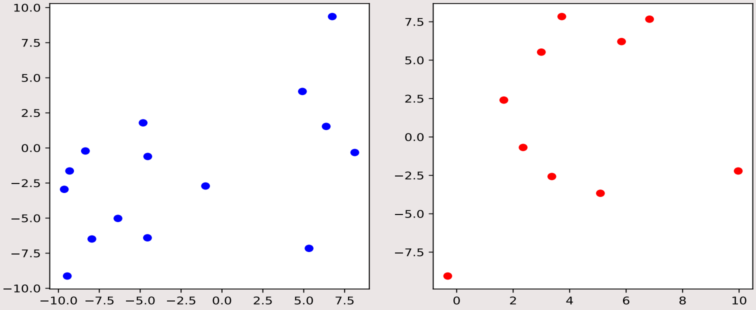

Figure: Two clouds of points in $d = 2$ dimensions with $n = 15$ points for the blue cloud and $n = 10$ points for the red one.

Evolutionary algorithm over clouds of points in convex domains

Optimization with Evolutionary Algorithm

Difficulties

- F is a black-box function, no information about its smoothness, *a fortiori* its convexity.
- All these aspects combined make it difficult to define gradients.

Related works for windfarm design

- We can find in Bilbao and Alba [2], Pillai et al. [4], and Pillai et al. [3] algorithms, optimizing positions, based respectively on simulated annealing, genetic algorithm and particle swarm optimization.
- Authors suppose predefined positions and use binary encoding. Our work differs by letting points vary continuously.

Evolutionary algorithms

- We adopt an evolutionary algorithm that evolves an initial population, creates new ones by **crossover** and **mutation** and stops after a fixed number of iterations.

How to define crossover and mutation over clouds of points ?

With the discrete uniform measures modeling

- For two cloud of points $X^{(j)} = \{\mathrm{x}_1^{(\mathrm{j})}, ..., \mathrm{x}_\mathrm{n}^{(\mathrm{j})}\}$, $j = 1, 2$ we associate $P_{X^{(j)}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathrm{x}_\mathrm{i}^{(\mathrm{j})}}$
- We can compute a new cloud of points by finding an intermediary uniform measure.

Wasserstein distance

- For two measures μ and ν defined over $\mathbb{R}^d$, the Wasserstein distance of order p is defined as follows : $W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \rho(x, x')^p \mathrm{d}\pi(x, x')$
  - $\rho(x, x')$ corresponds to the Euclidean distance between x and x'
  - $\Pi(\mu, \nu)$ is the set of all probability measures defined over $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν.

Wasserstein barycenter

- *A barycenter* (ν^*) *of* N *measures* $\nu_1, ..., \nu_N$ *is defined as to minimize* $f(\nu) = \sum_{i=1}^{N} \epsilon_i W_p^p(\nu, \nu_i)$, with $\epsilon_i \geq 0, \sum_{i=1}^{N} \epsilon_i = 1$ see Agueh and Carlier [1].

Wasserstein distance to define barycenter

Wasserstein barycenter: illustration

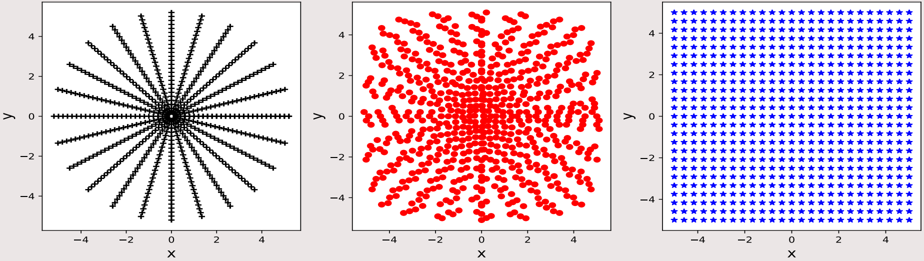

Figure: Two initial clouds at left and right, and their Wasserstein barycenter in the middle

Contracting effect

Theorem

Consider $\mathcal{P}'$ to be the set of discrete measures over $\mathbb{R}^d$ with finite support and $\epsilon \in [0,1]$. Let P_{X_1}, P_{X_2} and P_{X^*} be defined respectively as

- $P_{X_1} = \sum_{i=1}^{n} \alpha_i \delta_{x_i^1}, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i > 0,$
- $P_{X_2} = \sum_{j=1}^{m} \beta_j \delta_{x_j^2}, \sum_{j=1}^{m} \beta_j = 1, \beta_j > 0$,
- $P_{X^*} = \sum_{l=1}^{k} \lambda_l \delta_{x_l^*}, \sum_{l=1}^{k} \lambda_l = 1, \lambda_l > 0$,

with P_{X^*} the unique minimizer of $\underset{P_X \in \mathcal{P}'}{\arg\min}\, \epsilon W_2^2(P_X, P_{X_1}) + (1-\epsilon) W_2^2(P_X, P_{X_2})$.

If the above is verified, we have:

$$\forall l \in \{1, ..., k\}, x_l^* \in \overline{Conv(x_1^1, ..., x_n^1, x_1^2, ..., x_m^2)}$$

where $\overline{Conv(x_1^1, ..., x_n^1, x_1^2, ..., x_m^2)}$ is the closed convex hull of the set $\{x_1^1, ..., x_n^1, x_1^2, ..., x_m^2\}$

Evolutionary operators: crossovers

- Given $\epsilon \sim \mathcal{U}[0,1]$
- **Equal weights crossover:** For two measures (P_{X_1} and P_{X_2}), X_c defined as

$$P_{X_c} = \arg\min_{P_X} W_2^2(P_X, P_{X_1}) + W_2^2(P_X, P_{X_2})$$

- **Random weights crossover:** For two measures (P_{X_1} and P_{X_2}), X_c defined as

$$P_{X_c} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_1}) + (1-\epsilon) W_2^2(P_X, P_{X_2})$$

- **What crossover ?**

Evolutionary operators: mutations

- **Escape from contraction:** To define operators taking into account the contracting property, we introduce the following operators over clouds of points
- **Full Domain mutation:** given X_c and X_{rand} a cloud of points randomly sampled in the domain, X_m defined as

$$P_{X_m} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_c}) + (1-\epsilon) W_2^2(P_X, P_{X_{rand}})$$

.

- **Boundary mutation:** given X_c and X_{bound} a cloud of points randomly sampled at the domain boundary, X_m defined as

$$P_{X_m} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_c}) + (1-\epsilon) W_2^2(P_X, P_{X_c \cup X_{bound}})$$

.

- How to arrange the two mutations ?

Alternating Mutation

A first type of mutation based on Wasserstein operators alternates, with a random weight, between the Boundary and the Full Domain mutations. It is detailed in Algorithm 1.

Algorithm 1 Alternating Wasserstein Mutation

Input: X cloud to mutate, *prob* the probability to perform a Boundary Mutation
Output: The mutated cloud(s)

```
Draw ϵ and r uniformly in [0, 1]
if r ≥ prob then
    Do Full Domain Mutation with weight ϵ
else
    Do Boundary Mutation with weight ϵ
end if
```

The two mutations can also be done successively in a deterministic way.

Default crossovers and mutations: comparison algorithm denoted Ref_gen

Crossing by random choice of points among parents

- Let $X^1 = \{{x^1}_1, ...x^1_{n_1}, \emptyset_{n_1+1}, ..., \emptyset_{n_{max}}\}$ and $X^2 = \{{x^2}_1, ...x^2_{n_2}, \emptyset_{n_2+1}, ..., \emptyset_{n_{max}}\}$
- $X^c = \{x_1, ...x_n, \emptyset_{n+1}, ..., \emptyset_{n_{max}}\}$ is their crossover if $\forall i \in \{1, ..., n_{max}\}$, x_i is randomly sampled in $\{{x^1}_i, {x^2}_i\}$ with a Bernoulli law (1/2). Rearrange to have full points on left.

Gaussian Mutation

- Let $X^c = \{x_1, ...x_n, \emptyset_{n+1}, ..., \emptyset_{n_{max}}\}$
- Sample m randomly in $\{n-1, n, n+1\}$
- Add or remove point according to m
- Perturb each point with a truncated Gaussian with a diagonal covariance matrix where the variance is given by the following :
  - $\sigma^2 = 0.01 * E[\|X - X'\|^2]$.

Test functions

Inspired from wind-farms

- We consider the following family of test functions mimicking wind-farms productions:
- $F_\theta(\{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{n}\}) = \sum_{i=1}^{n} \left(\prod_{j, j \neq i} f_{\mathrm{x}_j, \theta}(\mathrm{x}_\mathrm{i}) \right) f_0(\mathrm{x}_\mathrm{i})$

Mindist and Inertia

- $F_{minDist}(\{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{n}\}) = \min_{i \neq j} ||\mathrm{x}_i - \mathrm{x}_j||$, $F_{inert}(\{\mathrm{x}_1, ..., \mathrm{x}_\mathrm{n}\}) = \sum_{i=1}^{n} ||\mathrm{x}_i - \bar{X}||^2$ with $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{x}_\mathrm{i}$

The input of the functions

- Number of points vary between 10 and 20, in a fixed rectangular domain. Number of iterations and populations sizes are respectively 500 and 300. We maximize the functions.

Diversity of population

Wasserstein-based diversity

The diversity can be calculated at each iteration in the following way:

$$\text{Div}(\text{pop}) = \frac{1}{\lambda} \sum_{X_i \in \text{pop}} W_2^2(P_{X^*}, P_{X_i}) \;, \tag{1}$$

where $\text{pop} = \{X_i, i = 1, ..., \lambda\}$ is a population of sets, P_{X_i} the associated discrete measures, and P_{X^*} is the Wasserstein barycenter of the clouds of pop. The support's size of P_{X^*} is chosen to be the mode of all sizes in pop.

Some notations

- WBGEA_1t denotes the algorithm based on Wasseerstein operators with equal weights crossover, WBGEA_1t_rc (random weight crossover) and WBGEA_1t_nc (no crossover)
- Ref_gen denotes the baseline comparison algorithm and Ref_gen_nc its version without crossover.

WBGEA vs Ref_gen

The results indicate that the algorithm based on Wasserstein operators denoted as WBGEA yields better results except on $F_{minDist}$.

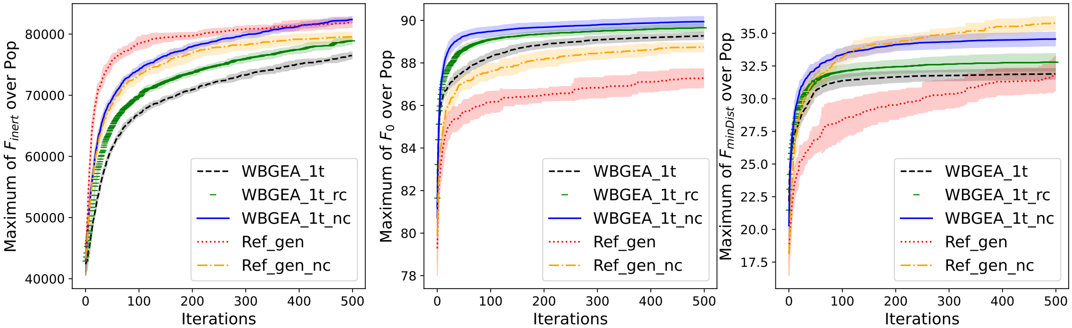

Figure: Average over 20 (+/- std. deviation) of the evolutions of the maximum of the functions in each population over the evolutionary algorithms iterations.

WBGEA vs Ref_gen

The diversities of the populations in the algorithms based on Wasserstein operators vanish to zero more quickly.

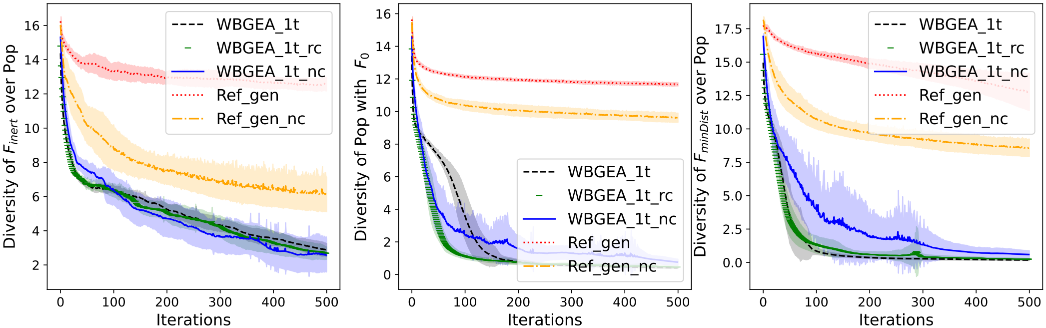

Figure: Average over 20 (+/- std. deviation) of the evolutions of the diversities of the populations over the evolutionary algorithms iterations.

Best designs

It can be seen that the designs are consistent with the simulated physical phenomena.

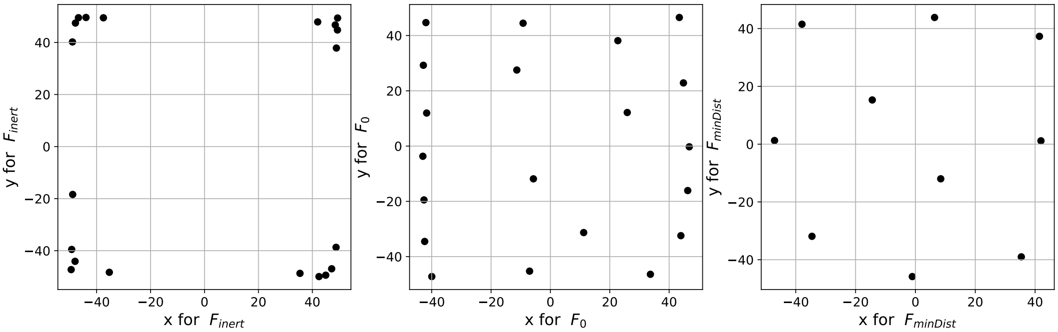

Figure: Best observed designs corresponding, respectively, to the test cases F_{inert}, F_0 and $F_{minDist}$ (left to right).

Conclusions I

- *Numerical tests suggest a mutation independence principle: for composite mutations made of different types of perturbations, like the boundary and the full domain mutations, the perturbations should be applied independently.*
- *For WBGEA, crossover with random weights yield better results but the absence of crossover is again more competitive on the test functions.*
- *The Wasserstein crossover reduces diversity and the classical crossover creates diversity.*
- *The Wasserstein operators seem to be adapted to optimize functions where the optimal design present regularity as alignments.*

Extend the operators in non convex domains

Non convex domains

- In the previous methods, the domain of the points are supposed to be convex.
- Let's consider now non convex domains including exclusion zones.

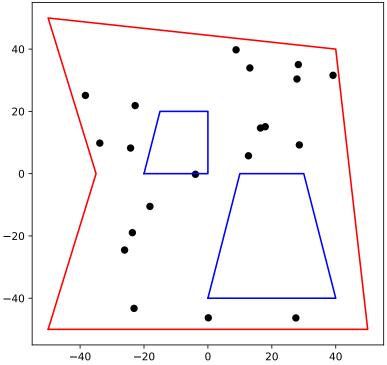

Figure: A non convex domain delimited by a red line and containing holes in blue as exclusion zones

Feasible clouds can give non feasible

- **Barycenter of two feasible clouds of points can produce a non feasible cloud**

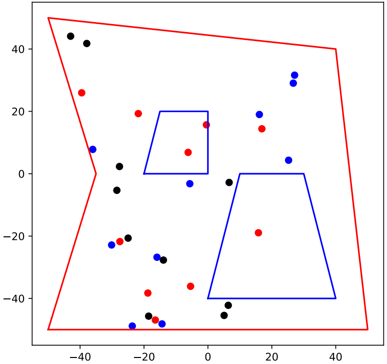

Figure: The two initial clouds of points in black and blue and their barycenter in red.

Projecting points

- A way to repair non feasible clouds of points is to project the non feasible points.

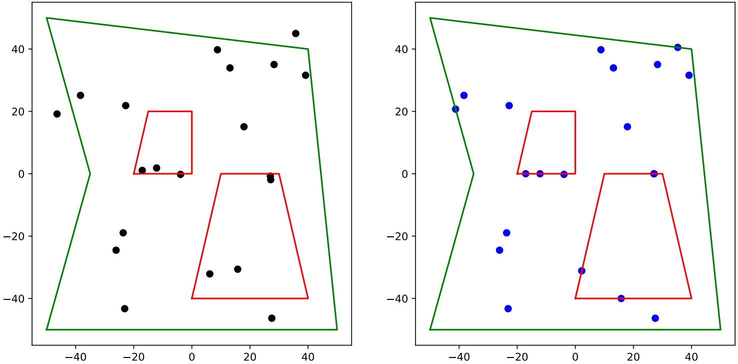

Figure: On left, the initial cloud of point in black contains non feasible points. On right, the the cloud of points in blue contains the projected points int he feasible domain.

Minimal deformation

Theorem: projecting points minimize the deformation

- Let's denote D_f the feasible domain.
- $P_X = \sum_{i=1}^{l} \alpha_i \delta_{x_i} + \sum_{i=l+1}^{n} \alpha_i \delta_{x_i}$, $\alpha_i > 0$
- $\forall k \in \{l+1, ..., n\}, x_k \notin D_f$
- $\forall k \in \{l+1, ..., n\}$, consider x_k^* the nearest feasible point of x_k
- $P_{X^*} = \sum_{i=1}^{l} \alpha_i \delta_{x_i} + \sum_{i=l+1}^{n} \alpha_i \delta_{x_i^*}$
- Then for any $P_Y = \sum_{j=1}^{m} \beta_j \delta_{y_j}$ ($\beta_j > 0$) containing only feasible points:

$$W_2(P_X, P_{X^*}) \leq W_2(P_X, P_Y)$$

Feasible Optimal designs

Even with the projections, the optimal designs reflect the simulated physical phenomena.

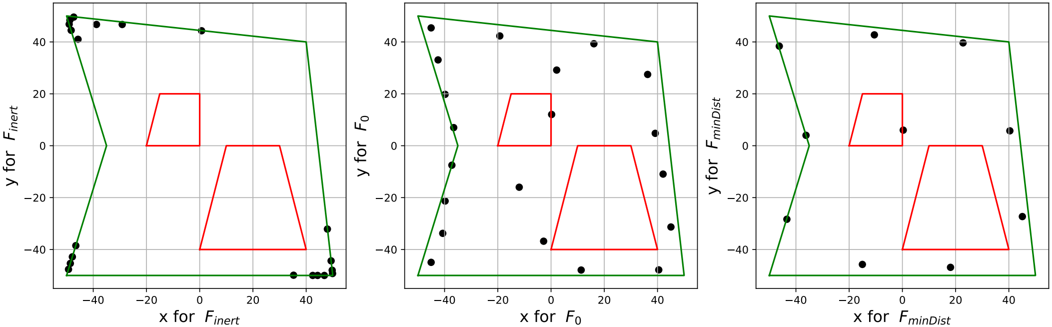

Figure: Best observed designs corresponding, respectively, to the test cases F_{inert}, F_0 and $F_{minDist}$ (left to right).

Conclusions II and perspectives

Conclusions II

- We have proposed evolutionary algorithms based on Wasserstein operators to optimize clouds of points in non convex domains.
- The Wasserstein operators are able to capture geometrical information related to the objectives functions.

Perspectives

- Theoretical convergence properties.
- Include in Bayesian optimization framework for time-consuming functions.

Thanks For Your Attention !

Bibliography I

[1] Martial Agueh and Guillaume Carlier. "Barycenters in the Wasserstein space". In: *SIAM Journal on Mathematical Analysis* 43.2 (2011), pp. 904–924.

[2] Martin Bilbao and Enrique Alba. "Simulated annealing for optimization of wind farm annual profit". In: *2009 2nd International symposium on logistics and industrial informatics*. IEEE. 2009, pp. 1–5.

[3] Ajit C Pillai et al. "Comparison of offshore wind farm layout optimization using a genetic algorithm and a particle swarm optimizer". In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 49972. American Society of Mechanical Engineers. 2016, V006T09A033.

[4] Ajit C Pillai et al. "Optimisation of offshore wind farms using a genetic algorithm". In: *International Journal of Offshore and Polar Engineering* 26.03 (2016), pp. 225–234.