Evolutionary algorithm over clouds of points: focus on Wasserstein barycenter interpolation

Phd Student: Babacar SOW

3d year PhD student

Project: ANR SAMOURAI

University: Ecole Des Mines de Saint-Etienne

Supervisors: Rodolphe LE RICHE (CNRS/LIMOS), Julien PELAMATTI, Merlin KELLER, Sanaa ZANNANE (EDF)

Table of contents

Context and problem formulation

Optimization of functions defined over clouds of points

- Deal with functions assumed to be **black box**.
- In this presentation, we consider functions having inputs in the form of **bag of vectors** (or point clouds).
- These types of functions are encountered in many domains, such as: **image processing**, **design of experiments** and **optimization**, ...

The design variable

- $X = \{x_1, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ and $n_{\min} \leq n \leq n_{\max}$. It will be referred to as a **cloud of points**.
- It is a specific type of mixed variables.

Example of an industrial problem : optimizing the layout of a wind-farm

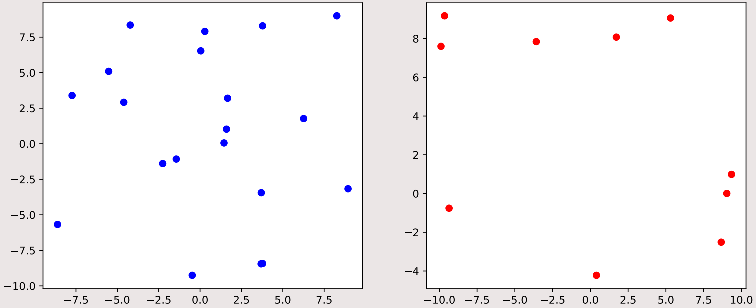
A set of points model

- **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 in a compact domain. The idea is to find a design maximizing annual production for instance.

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 each of their inputs.



Two clouds of points in $d = 2$ dimensions with $n = 20$ points for the blue cloud and $n = 10$ points for the red one. How to compare them ?

Optimization with Evolutionary Algorithm

Difficulties

- F is a black-box function, no information about its smoothness, a fortiori its convexity.
- The presence of a mixed variables make it difficult to define gradients.

Evolutionary algorithms

- Handle the candidate solutions using geometry and or interaction.

Related works

- We can find in [2], [5], and [4] 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.

The structure of the algorithm

Evolutionary algorithm structure

- λ the size of population, N_{iter} the number of iterations, F the function to be optimized and P the population.
- Choose λ clouds randomly to initialize $Pop = \{X_i, i = 1, ..., \lambda\}$.
- For $k = 1, \ldots, N_{\text{iter}}$
  - Compute $F(X_i)$ for $i = 1, ..., \lambda$.
  - Create $2 * \lambda$ new clouds by crossover.
  - Mutate each new cloud.
  - Selection: keep the λ best clouds among the parents and the children of the population.

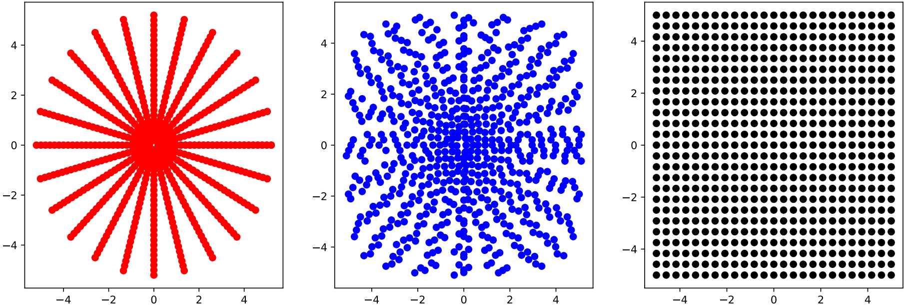
Generation of new clouds of points (children)

- How to cross and mutate clouds of points ?

Interpolation using Wasserstein-barycenter

With the discrete uniform measures

- To each cloud of points $X = \{x_1, ..., x_n\}$, we associate $P_X = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$
- We can compute a new cloud by interpolation with Wasserstein barycenter.



X_1 and X_2 (respectively in black and red are two initial clouds and X (in blue) represents their Wasserstein barycenter

Wasserstein distance to define barycenter

Wasserstein distance

- For two measures μ and ν defined over $\mathbb{R}^d$, the Wasserstein distance of order p is defined as follows : $W_p^p = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \rho(x, x')^p \mathrm{d}\pi(x, x')$
  - $\rho(x, x')$ correspond 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* $\nu_1, \dots, \nu_N$ *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 Barycenter as a crossover

Wasserstein Barycenter

- **Equal weights crossing :** For two measures (P_{X_1} and P_{X_2}) we can take $P_{X_c} = \arg\min_{P_X}(W_2^2(P_X, P_{X_1}) + W_2^2(P_X, P_{X_2}))$, to be a new design.
- **Random weights crossing :** For two measures (P_{X_1} and P_{X_2}) and a random $\epsilon \in [0, 1]$ we can take $P_{X_c} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_1}) + (1-\epsilon) W_2^2(P_X, P_{X_2})$, to be a new design.

Comments on the sizes

- If P_{X_1} and P_{X_2} have two different supports' sizes, k_1 and k_2, two crossings are done yielding two new designs (with supports k_1 and k_2).
- Algorithms to compute this minimum are discussed in Cuturi and Doucet [3].

Contracting effect

Theorem

Consider $\mathcal{P}'$ be the set of discrete measures over R^d with finite support and ϵ a positive real number. Let P_{X_1}, P_{X_2} and P_{X^*} be defined respectively as $\sum_{i=1}^{n} \alpha_i \delta_{x_i^1}, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0$, $\sum_{j=1}^{m} \beta_j \delta_{x_j^2}, \sum_{j=1}^{m} \beta_j = 1, \beta_j \geq 0$, $\sum_{l=1}^{k} \lambda_l \delta_{x_l^*}, \sum_{l=1}^{k} \lambda_l = 1, \lambda_l \geq 0$ with

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

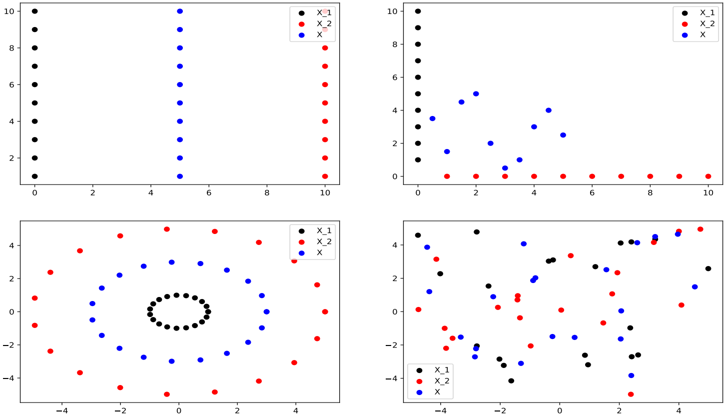
If the above is verified we have $\forall l \in \{1, ..., k\}, x_l^* \in \overline{Conv(x_1^1, ..., x_n^1, x_1^2, ..., x_m^2)}$ where $\overline{Conv(x_1^1, ..., x_n^1, x_1^2, ..., x_m^2)}$ is the closed convex hull of the set $\{x_1^1, ..., x_n^1, x_1^2, ..., x_m^2\}$

Escape from contraction !

One needs to define mutations helping to escape from contraction. How ?

Crossings' examples

The given clouds are contained in the convex hull of the union of the initial clouds.



X_1 and X_2 respectively in black and red are two initial clouds and X (in blue) represents their Wasserstein barycenter

Mutations

We introduce the following mutations over clouds of points. X_m is a mutation of X_c.

Definitions

- $\epsilon \sim \mathcal{U}[0, 1]$
- **Boundary-Mutation (BM)** : $P_{X_m} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_c}) + (1-\epsilon) W_2^2(P_X, P_{X_c \cup Dom}))$.

  Dom is a cloud of points at the domain boundary, it can be **randomly sampled**. To fix ideas, for a polygon, *Dom* is be the union of points randomly sampled on the sides On each side, one samples a point.

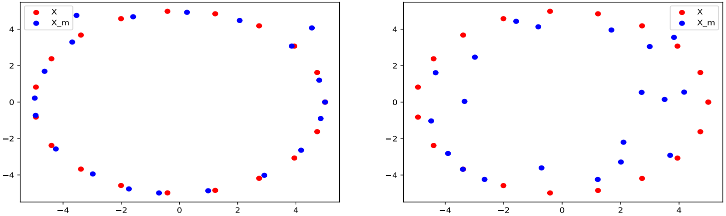
- **Weighted-Sample-Wasserstein (WSM)** :
  $P_{X_m} = \arg\min_{P_X} \epsilon W_2^2(P_X, P_{X_c}) + (1-\epsilon) W_2^2(P_X, P_{X_{rand}})$ with the following :
  - X_{rand} is a randomly sampled.
- What mutation do we adopt ?

One (of the two) mutation with a random weight (ϵ)

We adopt the following mutation after experimental results.

A one mutation

- $r \sim \mathcal{U}[0,1]$
- If $r \geq 0.5$:
  - Do BM
- Else :
  - Do WSW



X and X_m (respectively in red and blue) are initial cloud and the mutated one with Wasserstein barycenter. BM on left and WSW on right.

Default crossovers and mutations : comparison algorithm denoted Cg

Crossing by random choice of points among parents

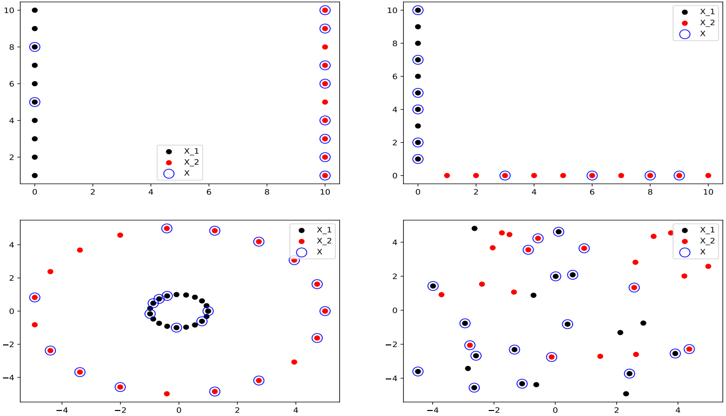
- Let $X^1 = \{{x^1}_1, ... x^1_{n_1}, \emptyset_{n_1+1}, ..., \emptyset_{n_{max}}\}$ and $X^2 = \{{x^2}_1, ... x^2_{n_2}, \emptyset_{n_2+1}, ..., \emptyset_{n_{max}}\}$
- $X^c = \{x_1, ... x_n, \emptyset_{n+1}, ..., \emptyset_{n_{max}}\}$ is their crossover. And $\forall i \in \ \{1, ..., n_{max}\}$, 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 points according to m
- Perturb each point with a truncated Gaussian with a diagonal covariance matrix where the variance is given by the following :
    - proportional to $\sigma^2 = E[\|X - X'\|^2]$.

Crossings' examples

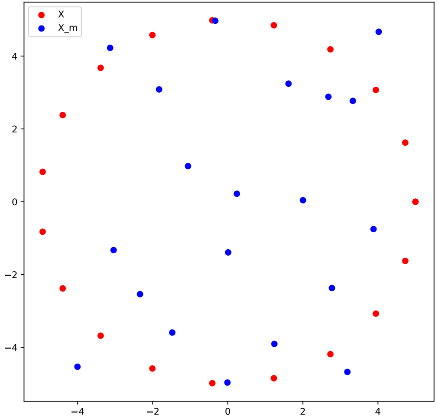
The new clouds contain points sampled randomly from the two initial clouds : points don't move.



X_1 and X_2 respectively in black and red are two initial clouds and X (in blue) is an example of their classical crossover

Gaussian mutation

Each point of the cloud is mutated with a truncated Gaussian.



X and X_m (respectively in red and blue) are initial cloud and the mutated one with truncated Gaussian.

Numerical experiments

Experiment set-up

- In the following we fix $\lambda = 300$, $N_{iter} = 500$.
- **WBGEA** (Wasserstein-Barycenter Evolutionary Algorithm with equal weights in crossing),
- **(WBGEA_rc)** (random weights in crossing)
- **(WBGEA_nc** (without crossing)
- **Cg** : the comparison algorithm algorithm
- **Cg_nc** (the comparison algorithm without crossing)
- For each algorithm, we compute the diversity of the population during the iteration with the followoing :
  - Consider $Pop = \{X_i, i = 1, ..., \lambda\}$
  - P_{X^*} being the Wasserstein barycenter of the population and its size defined as the mode of all sizes in Pop.
  - Diversity of Pop is $(1/\lambda) \sum_{i=1}^{\lambda} W_p^p(P_{X^*}, P_{X_i})$
- We compare all the algorithms on the following test functions.

Test functions

Inspired from wind-farms

- We consider the following family of test functions mimicking wind-farms productions

$$F_\theta(\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}) = \sum_{i=1}^{n} \left(\prod_{j, j \neq i} f_{\mathrm{x}_j, \theta}(\mathrm{x}_i) \right) f_0(\mathrm{x}_i) \quad (1)$$

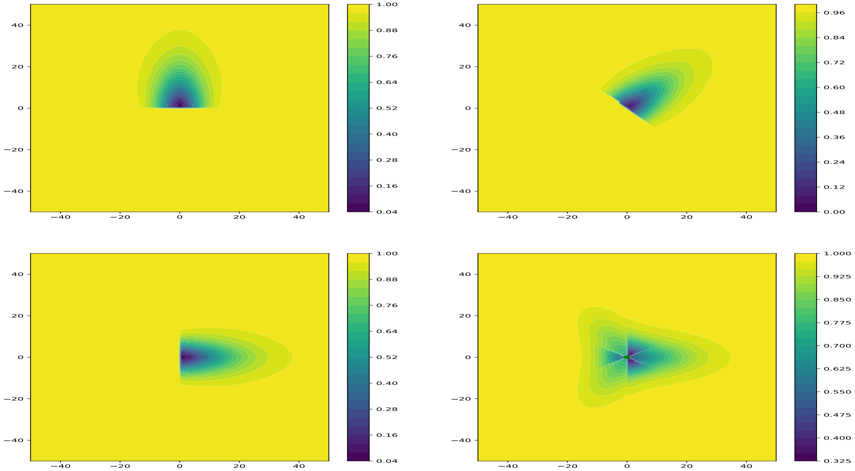
Mindist and Inertia

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

The input of the functions

- The number of points of the inputs vary between 10 and 20. We maximize the functions.

Wind farm test functions



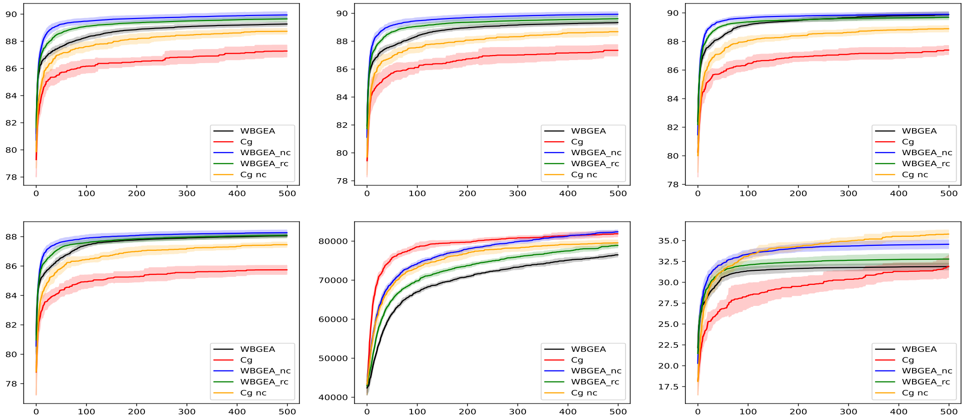
Representation of $f_{x,\theta}$ (x=(0,0)) with $\theta = 90^\circ$ at top left, $\theta = 45^\circ$ top right, $\theta = 0^\circ$ bottom left, and averaged directions at bottom right. We denote the corresponding functions respectively F_{90}, F_{45}, F_0, F_{4d}.

Presentation of results

- We apply 5 algorithms on each test function.
- The results are presented respectively in the following order F_0, F_{90}, F_{45}, F_{4d}, F_{inert}, $F_{minDist}$
- The x-axis correspond to the number of iterations.
- Concerning the algorithm's performances, the y-axis correspond to the mean over 20 of the maximum of each population.
- We do the same for the diversity (y-axis correspond to the mean over 20 of the diversity of the population).

Progress of the maximum of each population during iterations

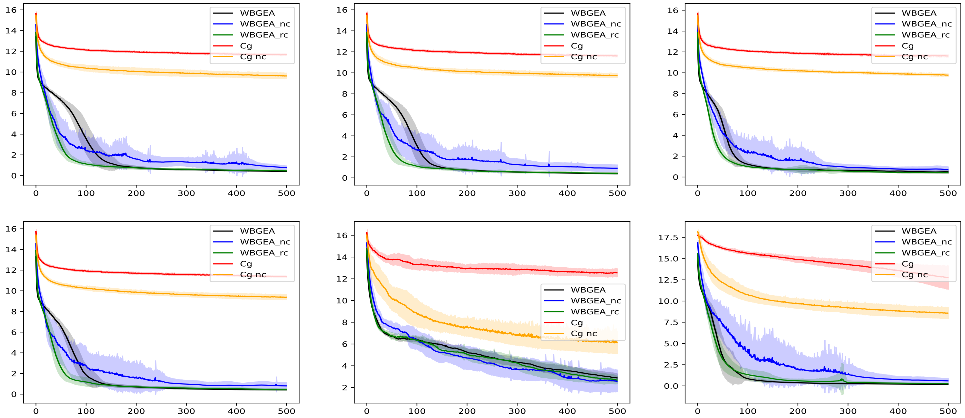
The absence of crossing improves all the algorithms. Random weights in crossing works better than equal ones in WBGEA. WBGEA_nc outperforms all the algorithms except on $F_{minDist}$.



The curves correspond respectively to F_0, F_{90}, F_{45}, F_{40d}, F_{inert} and $F_{minDist}$ (left to right, top to bottom).

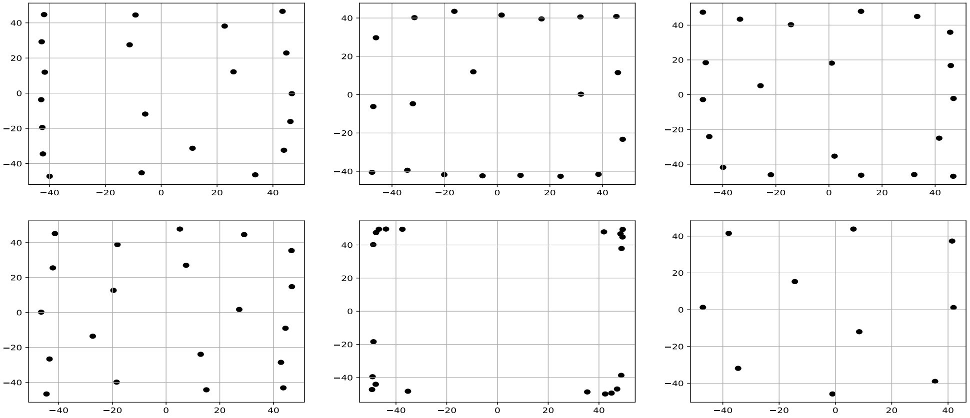
Diversity of the algorithms

When crossing with equal weights in WBGEA, one observes a quite similar diversity of population than with random ones for long term. The absence of crossing helps to keep more the diversity of population in WBGEA. We observe a higher diversity with Cg.



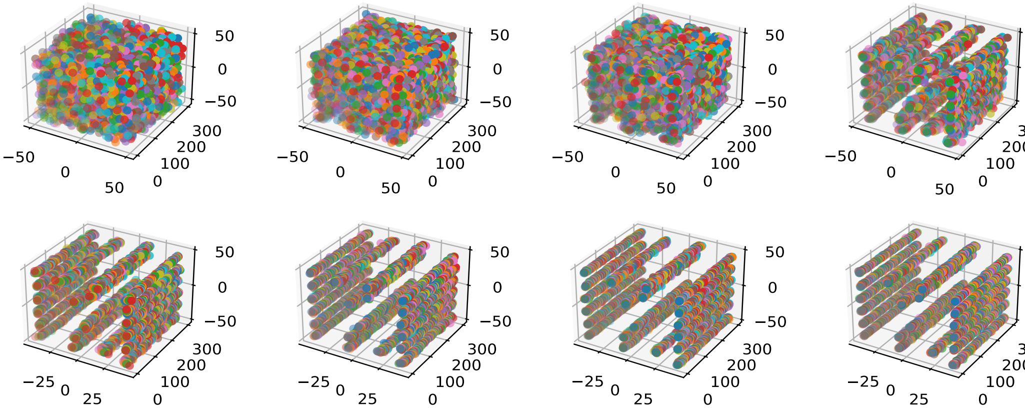
The curves correspond respectively to F_0, F_{90}, F_{45}, F_{40d}, F_{inert} and $F_{minDist}$

We observe an adaptation of the returned designs to the optimized functions. The points are placed optimally according to the wind's direction for wind-farms analytical functions.



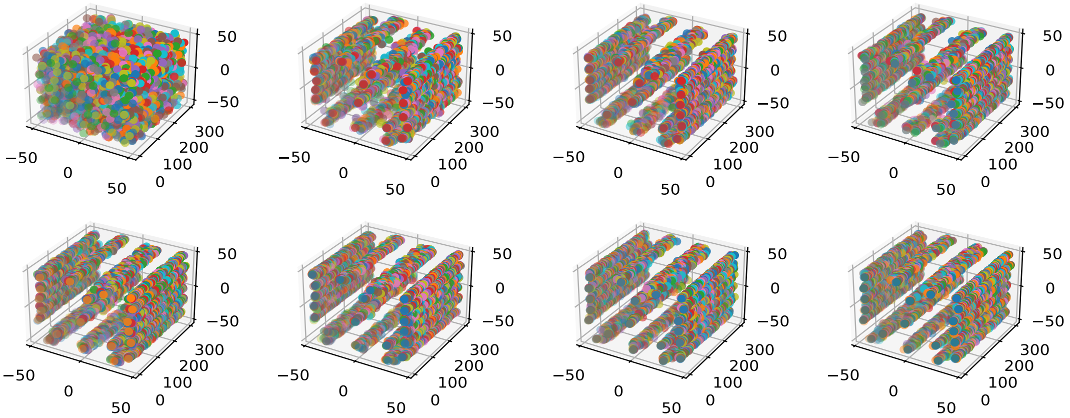
The designs correspond respectively to F_0, F_{90}, F_{45}, F_{4d}, F_{inert} and $F_{minDist}$ (left to right, top to bottom).

We can visualize the similarity of the clouds of points in the population progressing during the iterations.



The designs of populations correspond respectively to the iterations 0, 50, 75, 100, 125, 150, 175, 200 (left to right, top to bottom).

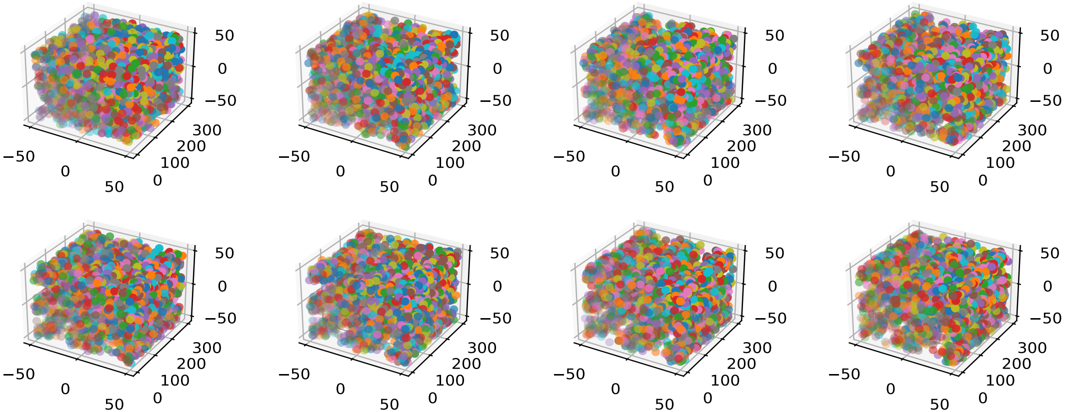
Without crossing, the progression of the similarity is accelerated during the first iterations and slightly reduced at the end.



The designs of populations correspond respectively to the iterations 0, 50, 75, 100, 125, 150, 175, 200 (left to right, top to bottom).

Population's view of Cg_nc the $F_{minDist}$

We can observe the great diversity due to the Gaussian mutation !



The designs of populations correspond respectively to the iterations 0, 50, 75, 100, 125, 150, 175, 200 (left to right, top to bottom).

Perspectives

Evolutionary algorithms over clouds of points

- Optimizing over non convex domains.
- Optimize with colored clouds of points.
- Extend to clouds with points of dimension $d > 2$.
- Combining with Gaussian process : Bayesian Optimization.

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] Marco Cuturi and Arnaud Doucet. "Fast computation of Wasserstein barycenters". In: *International conference on machine learning*. PMLR. 2014, pp. 685–693.

[4] 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.

Bibliography II

[5] 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.