Active learning for the optimization of functions defined over clouds of points

Babacar Sow, Rodolphe Le Riche, Julien Pelamatti, Merlin Keller, Sanaa Zannane

Optimization problems defined over sets of points. Active learning for the computationally intensive case.

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**.

Two alternative approaches

- Computationally cheap case: evolutionary algorithm
- Computationally intensive case: active learning

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.

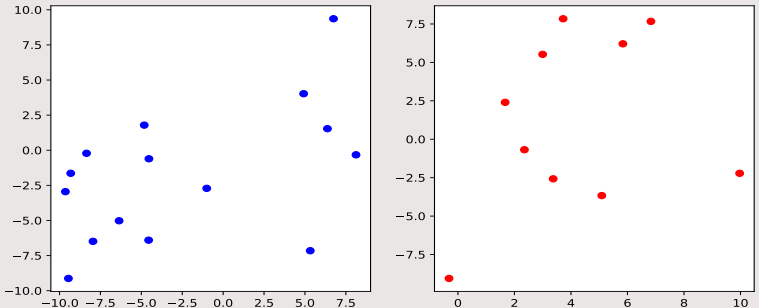

Figure: Two clouds of points in $d = 2$ dimensions with $n = 15$ points for the blue cloud and $n = 10$ points for the red one.

Evolutionary algorithm over clouds of points

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. [6], and Pillai et al. [5] 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 $X^{(j)} = \{x_1^{(j)}, ..., x_n^{(j)}\}$, $j = 1, 2$ we associate $P_{X^{(j)}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i^{(j)}}$
- 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

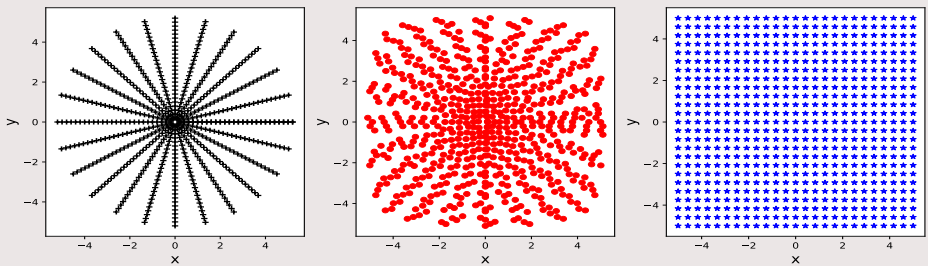

Figure: Two initial clouds at left and right, and their Wasserstein barycenter in the middle

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

- **Escape from contraction:** To define operators taking into account the contracting property, we introduce the following operators over clouds of points, given $\epsilon \sim \mathcal{U}[0,1]$:
- **Random weights crossover:** For two measures (P_{X_1} and P_{X_2}), X_c defined as

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

- **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

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

.

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 $r \geq prob$ **then**
    Do Full Domain Mutation with weight ϵ
else
    Do Boundary Mutation with weight ϵ
end if

The absence of crossover improves the results. 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.

Default crossovers and mutations: comparison algorithm denoted Gauss

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:
- $F_\theta(\{x_1, ..., x_n\}) = \sum_{i=1}^{n} \left(\prod_{j, j \neq i} f_{x_j, \theta}(x_i) \right) f_0(x_i)$
- $F_{\theta\_pen}(\{x_1, ..., x_n\}) = F_\theta(\{x_1, ..., x_n\}) - n\sqrt{n} + 1.5n$

Mindist and Inertia

- $F_{minDist}(\{x_1, ..., x_n\}) = \min_{i \neq j} ||x_i - x_j||$, $F_{inert}(\{x_1, ..., x_n\}) = \sum_{i=1}^{n} ||x_i - \bar{X}||^2$ with $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$

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.

WBGEA vs Gauss

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

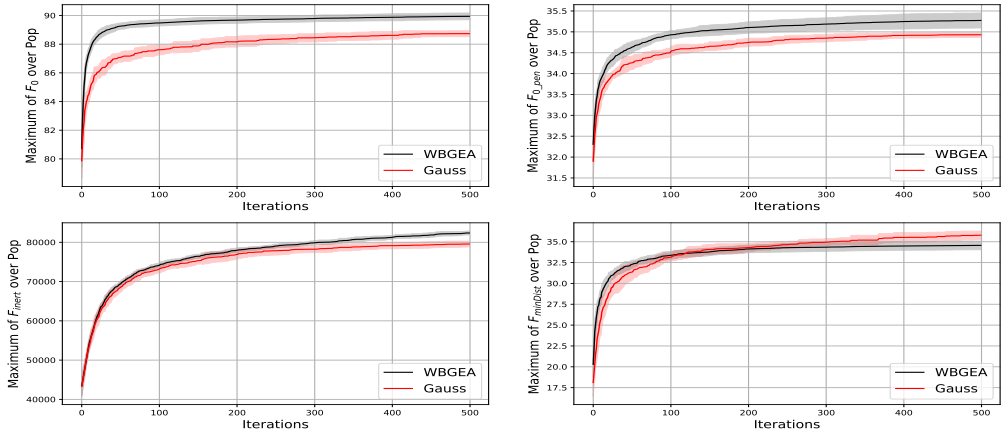

Figure: Average over 20 (+/- std. deviation) of the evolutions of the maximum of the functions in each population over the evolutionary algorithms iterations.

Best designs

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

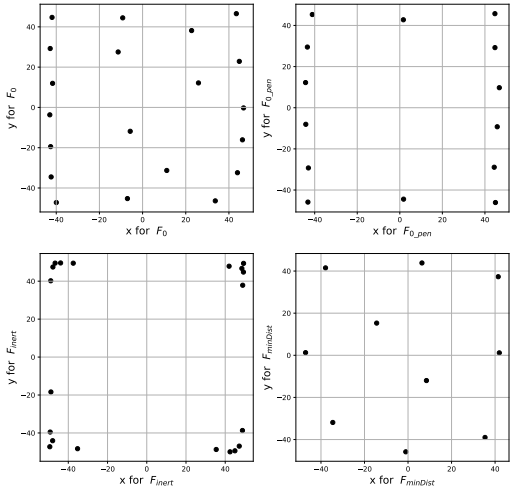

Figure: Best observed designs corresponding, respectively, to the test cases F_0, F_{0_pen}, F_{inert} and $F_{minDist}$ (left to right, top to bottom).

Gaussian processes and active learning over clouds of points

Bayesian Approach

A Gaussian process prior

- Gaussian processes are defined by a mean function m and a covariance kernel k over the input spaces $\mathcal{X}$; it can be used as a prior law to approximate a costly function $F \in \mathcal{F}$.
- Observing $D = \{(X_1, y_1)...(X_N, y_N)\}$ where $X_i \in \mathcal{X}$ and $y \in \mathbb{R}$ as training data with $y_i = f(X_i)$, the predictive mean m and variance Σ at a new point X are given by:

$$\mu(X; D) = m(X) + K(\mathbb{X}, X)^T K(\mathbb{X}, \mathbb{X})^{-1}(y - m(\mathbb{X}))$$

$$\Sigma(X, X; D) = K(X, X) - K(\mathbb{X}, X)^T K(\mathbb{X}, \mathbb{X})^{-1} K(\mathbb{X}, X)$$

with $\mathbb{X} = [X_1, ..., X_N]$ and $y = [y_1, ..., y_N]$, K is a matrix.

Necessary Conditions on k

- k must be symmetric and positive definite, i.e, for any M distinct clouds of points, for any vector $c \in \mathbb{R}^M$, the following inequality must hold: $\sum_{i=1}^{M} \sum_{j=1}^{M} c_i c_j k(X_i, X_j) \geq 0$

Kernel between clouds of points

Kernel through measures and features

- In Sow et al. [7], we discussed kernels based on features, sliced-Wasserstein distance and embedding models for regression problems.

Embedding models

- We assume that there exists a Reproducing Kernel Hilbert Space, $\mathcal{H}$, with a characteristic kernel $k_{\mathcal{H}}$ over the space of the vectors x
- The characteristic nature guarantees the injectivity of the embedding map (Muandet et al. [4]): $P_X \longmapsto \mu_X(.) = \int P_X(\mathrm{x}) k_{\mathcal{H}}(\mathrm{x}, .)\mathrm{dx}$.
- $MMD^2(P_X, P_{X'}) = ||\mu_X - \mu'_X||^2_{\mathcal{H}}$
- With kernel $k_{\mathcal{H}}$, for and any uniform discrete laws: $MMD^2(P_X, P_{X'}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\mathcal{H}}(\mathrm{x}_i, \mathrm{x}_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} k_{\mathcal{H}}(\mathrm{x}'_i, \mathrm{x}'_j) - 2\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k_{\mathcal{H}}(\mathrm{x}_i, \mathrm{x}'_j)$

MMD kernel

- The following kernel over clouds of points is symmetric and semi-definite positive (see Muandet et al. [4]):

$$k(X, X') = \sigma^2 exp\left(-\frac{||\mu_X - \mu_{X'}||^2_{\mathcal{H}}}{2\theta^2}\right)$$

- The associated Gaussian process is used in Sow et al. [7] to approximate the previously presented test functions.
- The hyper-parameters (σ^2, θ) and those of $k_{\mathcal{H}}$(Matérn 5/2 for the remainder) are found by fitting the model to the observed data through a maximization of the likelihood.
- We get the best results with MMD on a large family of test functions (see Sow et al. [7])

Active learning for Bayesian optimization (BO)

Gaussian process and acquisition function

- $Y \sim \mathcal{N}(\mu(X, D), \Sigma(X, X, D))$
- Expected improvement (EI): $EI(X) = E_{Y \sim \mathcal{N}(\mu(X,D),\Sigma(X,X,D))}[max(y^{min} - Y, 0)]$
- $EI(X) = (y^{min} - \mu(X, D_t))\Phi(\frac{(y^{min} - \mu(X,D_t)}{\sqrt{\Sigma(X,X,D_t)}}) + \sqrt{\Sigma(X, X, D_t)}\phi(\frac{(y^{min} - \mu(X,D_t)}{\sqrt{\Sigma(X,X,D_t)}})$, Jones, Schonlau, and Welch [3] with Φ and ϕ representing respectively the cumulative distribution and density function of the normal law.

BO

- Choose kernel k, computational budget T, and initial design $D_0 = \{(X_1, y_1)...(X_N, y_N)\}$ at $t = 0$
- For $t = 1, \ldots, T$:
  - Build Gaussian process with kernel k and D_t
  - Find X_{new}, the maximum of the acquisition criterion EI with WBGEA
  - Set $D_{t+1} \longleftarrow \mathrm{D}_t \bigcup(\mathrm{X}_{new}, y_{new})$, $y^{min} \longleftarrow \min(\mathrm{y}^{min}, y_{new})$

Results

We fix the budget to $T = 100$. A random initial set of 50 clouds is chosen. The hyper-parameters of the kernel are updated every 5 iterations by maximization of the likelihood.

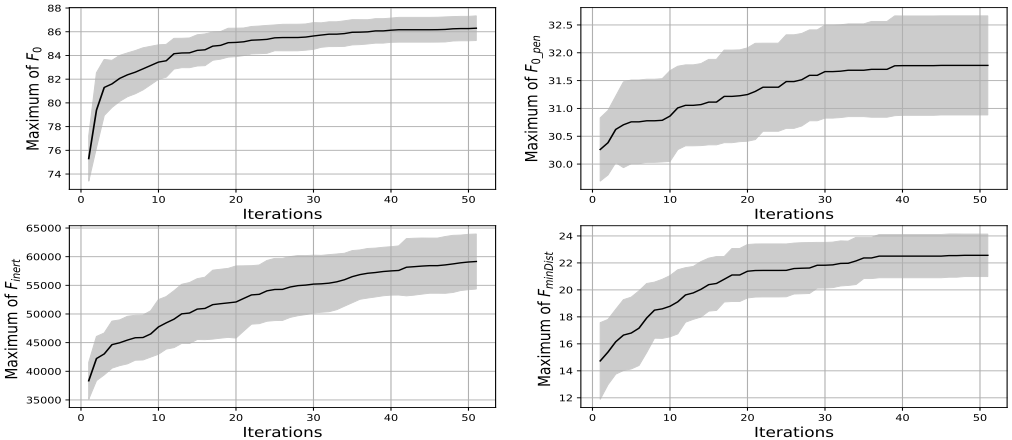

Figure: Average over 20 (+/- std. deviation) of the evolutions of the maximum of the functions over the BO iterations.

Conclusions

Bayesian optimization results

- The combination of MMD kernels and WBGEA yield good results for approximation and optimization on wind-farms functions, a bit less on other functions (F_{inert} and $F_{minDist}$).
- We present below the percentage of the maximum value attained by Bayesian optimization (BO) with respect to that attained by WBGEA (denoted by Percentage_BO_WBGEA):

Functions	F_0	F_{0_pen}	F_{inert}	$F_{minDist}$
Percentage_BO_WBGEA	95.95%	90.07%	71.81%	65.28%

Perspectives

- Apply to real-life industrial test cases
- Define criteria of design of experiments over clouds of points.
- Constraint points to non convex domains.
- Improve the choice of kernel and acquisition criterion.

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] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), pp. 455–492.

[4] Krikamol Muandet et al. "Kernel mean embedding of distributions: A review and beyond". In: *Foundations and Trends® in Machine Learning* 10.1-2 (2017), pp. 1–141.

Bibliography II

[5] 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.

[6] 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.

[7] Babacar Sow et al. "Learning functions defined over sets of vectors with kernel methods". In: *5 th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2023)*. 2023.