Gradient-enhanced Bayesian optimization for aerodynamic design

Emre Özkaya, Long Chen and Nicolas R. Gauger

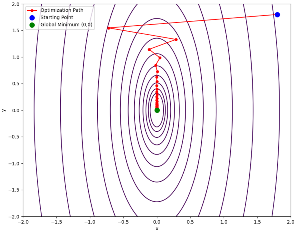
Chair for Scientific Computing
University of Kaiserslautern-Landau (RPTU)

16.09.2025

Motivation: A Unified Framework for Design Optimization

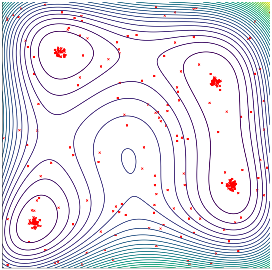
Gradient based optimization

- ✓ Suitable for high-dimensional problems
- ✓ Proven fast convergence (local optima)
- ✗ Constraint treatment may be challenging
- ✗ Sensitive to noise; initial guess matters



Surrogate based optimization

- ✗ Best suited for low- to medium-dimensional problems
- ✓ Facilitates easy constraint handling
- • Slower convergence but global search
- ✓ Robust and easy to parallelize
- ✗ May suffer from over-exploration



Kriging Model

Training data:

$$X = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}]$$

$$\mathbf{y} = [f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \ldots, f(\mathbf{x}^{(N)})]^T.$$

Kriging estimator:

$$\tilde{f}(x) = \mu + r^T R^{-1}(\mathbf{y} - \mu \mathbf{1}_N).$$

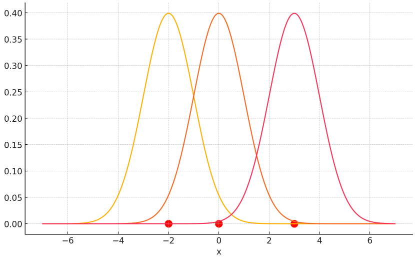
$$R = \begin{bmatrix} \phi(x^{(1)}, x^{(1)}) & \phi(x^{(1)}, x^{(2)}) & \ldots & \phi(x^{(N)}, x^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(x^{(N)}, x^{(1)}) & \phi(x^{(N)}, x^{(2)}) & \ldots & \phi(x^{(N)}, x^{(N)}) \end{bmatrix}$$

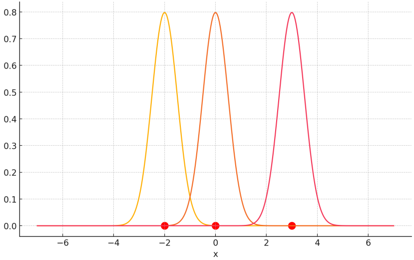
and

$$r = [\phi(x^{(1)}, x), \ldots, \phi(x^{(N)}, x)]^T$$

Kriging Model

Correlation function: $\phi(\mathbf{x}_i, \mathbf{x}) = \exp\left(\sum_{j=1}^{d} -\theta_j (x_j^{(i)} - x_j)^2\right)$





- Large θ: faster decay, less overlap, better conditioned system but less interpolative
- Small θ: slower decay, larger overlap, ill-conditioned system but more interpolative

Gradient Enhancement

The idea is not new.

- Han et al.: GEK, later proposed WGEK to improve robustness.
- Bouhlel et al.: Used PLS + GEK.
- Zimmermann et al.: the SGEK approach.

However, the core problem still unresolved:
ill-conditioned R.

We reinterpret the Kriging model as: Constant term + RBF expansion

$$\tilde{f}(\mathbf{x}) = \mu + \mathbf{r}^T R^{-1}(\mathbf{y} - \mu \mathbf{1}_N) \quad \Leftrightarrow \quad \mu + \sum_{i=1}^{N} w_i \, \phi_i(\mathbf{x})$$

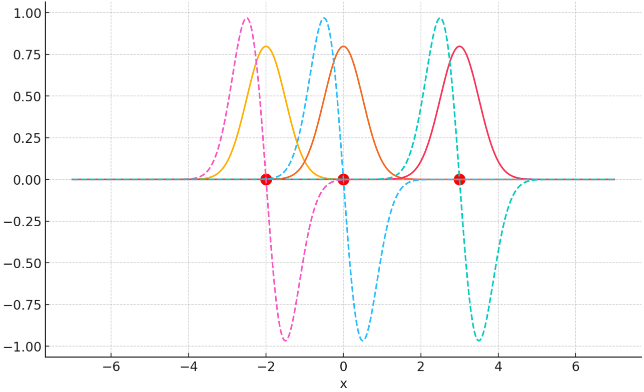
with the weight vector:

$$\mathbf{w} = R^{-1}(\mathbf{y} - \mu \mathbf{1}_N)$$

Derivative Enhanced Model

Idea: add new basis functions using the directional derivatives of ϕ

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x}) + \sum_{i=1}^{N} w_{N+i}\, \mathbf{d}_i \cdot \nabla \phi_i(\mathbf{x}) + \mu, \quad \mathbf{d}_i = \frac{\nabla f(\mathbf{x}^{(i)})}{||\nabla f(\mathbf{x}^{(i)})||}.$$



RoDeO[1]: **Ro**bust **De**sign **O**ptimization Package

- GPLv3 license
- Usage through C++ API or *.xml config files
- **Surrogate models:**
  - Kriging (default)
  - Generalized directional derivative enhanced model (adjoint/tangent-linear)
- **Efficient Global Optimization (EGO)**
- **Gradient Enhanced Efficient Global Optimization (GE-EGO)**
- **Multi-objective optimization: Epsilon-constraint method:**
  - Easy to implement
  - Easy to parallelize
  - Works fine with non-convex Pareto fronts

[1] https://github.com/eoezkaya/RoDeO.git

Minimization Benchmark Results[2]

Problem	Samples	Dim.	ISRES	DIRECT	ESCH	CRS	EGO	GE-EGO
Himmelblau	100	2	2.0725	0.0013	13.636	1.9713	0.0369	0.0049
Alpine02	200	5	-49.000	-80.183	-24.975	-47.418	-70.404	-87.807
Eggholder	100	2	-738.05	-931.59	-684.16	-791.35	-846.82	-853.93
Rosenbrock	100	2	0.6705	0.1111	0.5694	0.5054	0.0238	0.0126
Borehole	150	8	13.835	9.3536	12.888	10.866	7.8201	8.4296
Xor	200	9	1.8446	5.3502	3.8157	1.7023	1.4560	1.2854
Mishra03	100	2	0.0817	-0.0477	0.0702	-0.0632	-0.1037	-0.0904
Mishra05	100	2	-0.0317	-0.0543	0.2209	-0.0919	-0.0943	-0.1021
Michalewicz	300	10	-3.6881	-7.4239	-4.5648	-3.7406	-7.0055	-6.9416
Wingweight	300	10	159.61	129.78	139.81	134.43	124.11	124.65
Paviani	200	10	-18.742	-38.701	-7.5512	-16.810	-27.734	-43.146
Rastrigin	400	15	174.54	191.32	140.88	169.65	24.315	59.846
Dixon-Price	200	16	70302	2033	49920	64987	4706	67
Cola	400	17	57.767	82.356	52.570	57.512	17.977	17.340
Ackley	200	20	19.281	13.006	19.273	19.328	14.184	18.0106

[2] Lower values are always better

Design Optimization of a Welded Beam[3]

Design Variables: $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$

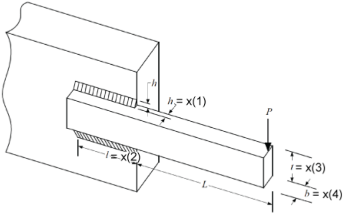
Objectives:

$$\min \quad f_1 = 1.10471\, x_1^2 x_2 + 0.04811\, x_3 x_4 (14 + x_2)$$

$$\min \quad f_2 = \frac{6000 \cdot 4 \cdot 14^3}{30 \times 10^6 \cdot x_3^3 x_4}$$

Constraints:

- $\tau(x) \leq 13{,}600$ (Shear stress)
- $\sigma(x) \leq 30{,}000$ (Normal stress)
- $x_4 - x_1 \geq 0$ (Geometry)
- Buckling capacity ≥ 6000



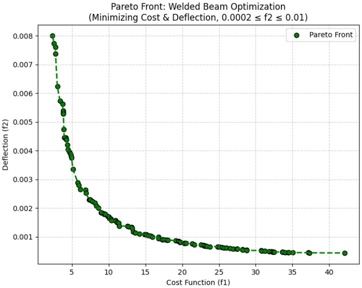
Bounds:

$$0.125 \leq x_1 \leq 5$$
$$0.1 \leq x_2, x_3, x_4 \leq 10$$

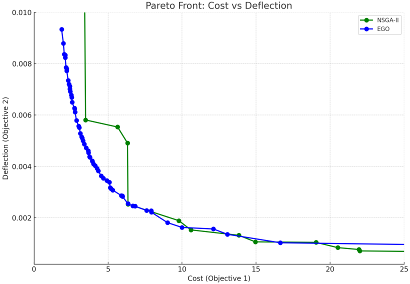
[3] https://de.mathworks.com/help/gads/multiobjective-optimization-welded-beam.html

Comparison: Brute Force - EGO - NSGA-II

Brute-force Pareto front



EGO vs NSGA-II results (400 samples)

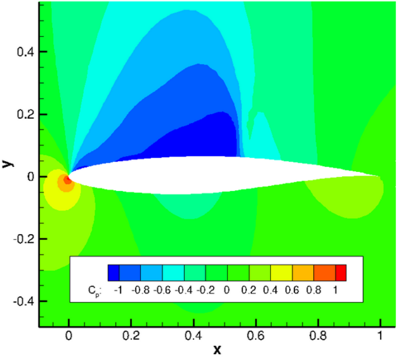


- **NSGA-II**: Broad exploration, less precision. Zooming requires additional constraints.
- **EGO**: High-resolution front. Natural zooming and better exploitation.

Multi-Objective Shape Optimization of RAE2822[4]

Optimization problem:

$$\min C_D, -C_L \quad \text{s.t. area} > \text{area}_0$$

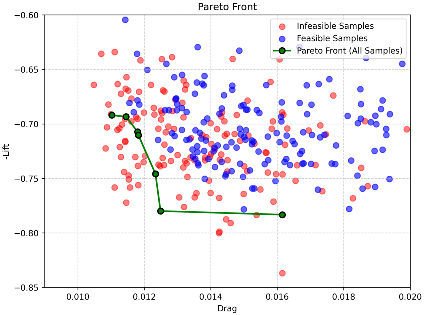


- $Ma = 0.8,\ Re = 6.5 \times 10^6$, $\alpha = 1.25°$
- Hicks-Henne parameterization with 38 design variables
- Discrete adjoint method for gradient computation
- 300 LHS samples for DoE + 300 evaluations during optimization

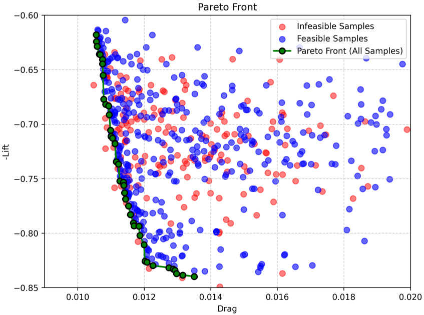
[4] https://su2code.github.io/tutorials/Turbulent_2D_Constrained_RAE2822/

Results

After DoE (300 Samples)

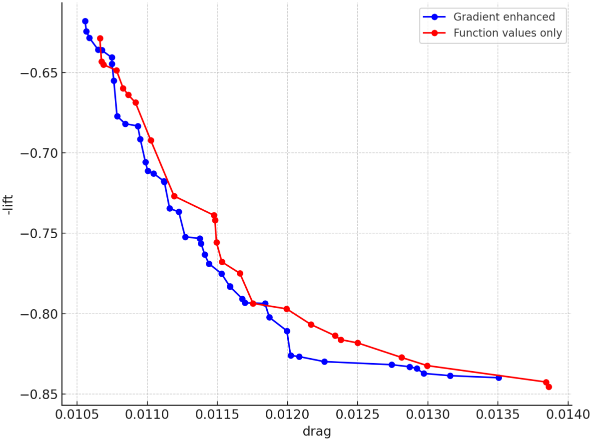


After Optimization (600 Samples)



- In total adjoint solver was called 13 times
- 41 Pareto optimal designs

GE-EGO vs EGO



- **Improved** Pareto front
- **Higher resolution** of the front with GE-EGO
- Only **2% overhead** due to adjoint computations

Thank you for your attention!