Making EGO and CMA-ES Complementary for Global Optimization

Hossein Mohammadi, Rodolphe Le Riche, Eric Touboul

Making EGO and CMA-ES complementary for global optimization

Hossein Mohammadi, Rodolphe Le Riche, Eric Touboul

Abstract

The global optimization of expensive-to-calculate continuous functions is of great practical importance in engineering. Among the proposed algorithms for solving such problems, *Efficient Global Optimization (EGO)* and *Covariance Matrix Adaptation Evolution Strategy (CMA-ES)* are regarded as two state-of-the-art unconstrained continuous optimization algorithms. Their underlying principles and their performances are different, yet complementary: EGO fills the design space in an order controlled by a conditional Gaussian process while CMA-ES learns and samples multi-normal laws in the space of design variables. In this paper, a new algorithm, called EGO-CMA, is proposed which combines the EGO and the CMA-ES algorithms. In EGO-CMA, the EGO search is interrupted early and followed by a CMA-ES search whose starting point, initial step size and covariance matrix are calculated from the already sampled points and the associated conditional Gaussian process. EGO-CMA improves the performance of both EGO and CMA in our 2 to 10 dimensional experiments.

Keywords: Black-box global optimization; CMA-ES; EGO; Optimization of expensive functions

1 Introduction

Continuous numerical optimization problems are at the core of many applications in science and engineering. It often happens that the underlying function is not only expensive to evaluate but also mathematically multimodal.

One approach to deal with expensive and multimodal optimization problems is to use surrogate models or metamodels. The idea of employing surrogate models for optimization of costly functions, with a focus on Gaussian Processes, has been reviewed in [13]. The deterministic Efficient Global Optimization (EGO) algorithm [14], which relies on a kriging model (i.e., conditional Gaussian Process), has become a standard for continuous global optimization in less than twenty dimensions when the number of function evaluations is inferior to 1000 [20].

Another popular algorithm in continuous global optimization is the stochastic Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [10]. CMA-ES

is interpreted as a robust local search method in [7]. Its robustness is attributed to invariance properties with respect to objective function scaling and coordinate system rotations. This algorithm was consistently found to be highly performing on the Black-Box Optimization Benchmarking (BBOB) framework for low, moderate, and highly multimodal functions for problems dimensions between 5 and 40 [9] if it is coupled with a restart mechanism. In [1, 6], restart strategies are proposed to prevent premature convergence of CMA-ES to a local minimum.

Much research on global optimization of costly functions has already involved augmenting Evolution Strategies (ES) with metamodels [12]. The general idea is to replace some evaluations of the true objective function with metamodel estimates and trigger true evaluations through an error rate measure. Let us focus here on the Covariance Matrix Adaptation (CMA) variant of evolution strategies. In [15], CMA-ES has been coupled with a local regression metamodel, making the lmm-CMA algorithm, where the metamodel allows savings in the ranking of the candidate solutions. References [18, 17] present the ^{s*}ACM-ES (surrogate Assisted Covariance Matrix adaptation Evolution Strategy), an algorithm with a ranking support vector machine as metamodel and where the number of iterations (generations) done with the metamodel depend on its error rate.

Kriging has sometimes been the metamodel added to the evolution strategies. The motivation for using kriging is the availability of a prediction uncertainty. In the work of [22], a pre-selection of the most promising points is done based on a kriging model, which enables sampling more solutions and makes the search more efficient. Two criteria are investigated as performance measures, the (mean) objective function prediction and the probability of improvement over the best observed point. Note that this probability of improvement can be defined thanks to the kriging uncertainty. In [3], kriging serves as a local metamodel and various performances are measured by different compromises between search intensification around the current best solution and exploration. In [16], a local kriging enables dealing with noisy objective functions by easing the estimation of the objective function expectation.

The optimization algorithm introduced in this paper differs from previous contributions in the fact that the EGO and CMA-ES search principles are invoked one after each other without iterations. The motivation is that EGO is efficient in the early design of experiments stage of the optimization, while CMA-ES is a converging search process.

2 A brief introduction to EGO and CMA-ES

2.1 Efficient Global Optimization (EGO)

EGO algorithm which first proposed in [14] is a method for using conditional Gaussian Processes (GP) to find the unconstrained minimum of a continuous multi-dimensional function. EGO iteratively creates a design of experiments aimed at finding the lower point of a function. At each iteration, one point is added to the existing design points such that a global optimization oriented infill

sampling criterion is maximized. There are different types of infill sampling criteria, see [2], but the expected improvement (EI) criterion is particularly popular. Because it does not have any arbitrary parameters to tune in order to set a compromise between search intensification and exploration. In the following the mathematical background of this algorithm is given.

Kriging model which is used in the EGO algorithm has been founded on the theory of Gaussian processes. A GP defines a distribution over functions. Formally, a GP indexed by D is a collection of random variables $\left(Y(\mathbf{x}); \mathbf{x} \in D \subseteq \mathbb{R}^d\right)$ such that for any $n \in \mathbb{N}$ and any $\mathbf{x}^1, ..., \mathbf{x}^n \in D$, $\left(Y(\mathbf{x}^1), ..., Y(\mathbf{x}^n)\right)$ follows a multivariate Gaussian distribution. A GP is parameterized by a mean function, $\mu(.)$, and a covariance function, or kernel, $K(.,.)$ [19].

The choice of kernel plays a key role in the obtained kriging model. The practice is that a parametric family of kernels is selected (e.g., Matérn, polynomial, exponential) and then the unknown parameters are estimated based on the observed values. For example, a squared exponential kernel is expressed as

$$cov\left(Y(\mathbf{x}^i), Y(\mathbf{x}^j)\right) = K\left(Y(\mathbf{x}^i), Y(\mathbf{x}^j)\right) = \sigma^2 \prod_{k=1}^{d} \exp\left(-\frac{\mid x_k^i - x_k^j \mid^2}{2\theta_k^2}\right), \quad (1)$$

in which σ^2 is a scale parameter known as process variance. The parameter θ_k is called characteristic length-scales and determine the correlation length. These parameters are usually estimated by ML or cross-validation. Interested readers are referred to [19] for more about kernels.

Now suppose that kernel $K(.,.)$ and its associated centered ($\mu(.) = 0$) GP are given. The matrix of design points $\mathbf{X} = \left(\mathbf{x}^1, ..., \mathbf{x}^n\right)$ denotes the locations where the samples are taken with the response values $\mathbf{y} = \left(y^1, ..., y^n\right)^\top$. To take into account this information, one can extract the posterior distribution of the underlying GP $(Y(\mathbf{x}))_{\mathbf{x}\in D}$. The posterior mean and variance of the conditional GP are given by [19]:

$$m(\mathbf{x}) = \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{y} \ , \quad (2)$$

$$v(\mathbf{x}) = \sigma^2 - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}) \ , \quad (3)$$

where $\mathbf{c}(\mathbf{x}) = \left(K(\mathbf{x}, \mathbf{x}^1), ..., K(\mathbf{x}, \mathbf{x}^n)\right)^\top$ is the vector of covariances between a new point $\mathbf{x}$ and the n already observed sample points. The $n \times n$ matrix $\mathbf{C}$ is a covariance matrix between the data points and its elements are defined as $\mathbf{C}_{i,j} = K\left(\mathbf{x}^i, \mathbf{x}^j\right)$.

A point is expected to improve the objective function if its predicted value is better than the current best point or the uncertainty in its prediction is such that the possibility of producing a better solution is high. Let y_{min} be the minimum sampled value of the true function we have observed yet. The improvement over the current y_{min} can be defied as

$$I(\mathbf{x}) = max\{0, y_{min} - Y(\mathbf{x})\}. \quad (4)$$

Then, EI is computed as follows:

$$EI(\mathbf{x}) = \begin{cases} (y_{min} - m(\mathbf{x}))\Phi\left(\frac{y_{min} - m(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{min} - m(\mathbf{x})}{s(\mathbf{x})}\right) & if\ s(\mathbf{x}) > 0 \\ 0 & if\ s(\mathbf{x}) = 0, \end{cases} \quad (5)$$

where $s(\mathbf{x}) = \sqrt{v(\mathbf{x})}$, $\phi(.)$ and $\Phi(.)$ are the probability density function (pdf) and the cumulative distribution function (cdf) of the standard normal distribution respectively.

At each iteration of EGO, the EI criterion is maximized and the optimal solution is added to the current design points. Then, the function value is evaluated at the optimal solution. The new function value is added to $\mathbf{y}$ and the parameters of kriging model are re-estimated.

The EI function is highly multimodal. Hence, the optimization is usually performed using stochastic optimization algorithms. For example, in the Scilab KRISP toolbox [need reference], the maximization of the EI is done by CMA-ES algorithm. Using such stochastic algorithms makes EGO stochastic as well. But if the EI maximization is done deterministically, EGO is a deterministic global optimization algorithm, as opposed to CMA-ES.

2.2 Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

First introduced by Hansen, Ostermeier, and Gawelczyk [11], CMA-ES is considered as the state-of-the-art algorithm for numerical black-box optimization if sufficient budget is afforded. CMA-ES is an iterative stochastic optimization algorithm where at each iteration, a population of search points are generated according to a multivariate normal law.

Let $\mathbf{m}^{(g)}$ be the mean vector of the multivariate normal distribution in generation g. The ith individual denoted by $\mathbf{x}_i^{(g+1)}$ is generated through:

$$\mathbf{x}_i^{(g+1)} = \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2 \mathbf{C}^{(g)}\right) = \mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right),\ i = 1, ..., \lambda, \quad (6)$$

where $\sigma^{(g)} \in \mathbb{R}^+$ is called mutation step size and $\mathbf{C}^{(g)} \in \mathbb{R}^{d\times d}$ is a covariance matrix. The former controls the step length and the later governs the shape of the distribution ellipsoid. It should be noted that the initialized covariance matrix is the identity matrix; $\mathbf{C}^{(0)} = \mathbf{I}$.

After generating λ individuals, they are evaluated and ranked according to the objective (fitness) function. Then μ best of them are selected. We denote the ith best search point by $\mathbf{x}_{i:\lambda}$. This selection that is only based on the fitness ranking makes CMA-ES invariant with respect to any monotonous transformation of the objective function. In general, λ and μ are determined as follows:

$$\lambda = 4 + \lfloor 3\ln(d) \rfloor, \quad (7)$$

$$\mu = \lfloor \frac{\lambda}{2} \rfloor. \quad (8)$$

How to update the mean, the covariance matrix and the step size for the next generation, i.e., $\mathbf{m}^{(g+1)}$, $\mathbf{C}^{(g+1)}$, and $\sigma^{(g+1)}$ has critical influence on the algorithm performance. The mean of the next generation is obtained from

$\mathbf{x}_{1:\lambda}^{(g+1)}, ..., \mathbf{x}_{\mu:\lambda}^{(g+1)}$ as follows:

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(g+1)} = \mathbf{m}^{(g)} + \sigma^{(g)} \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}, \tag{9}$$

where $\mathbf{y}_{i:\lambda} = \frac{\left(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}\right)}{\sigma^{(g)}}$ and ω_i denotes the weight. This update moves the mean vector towards the best solutions. Note that the weights in Equation (9) are strictly positive and normalized:

$$\sum_{i=1}^{\mu} \omega_i = 1, \ \omega_1 \geqslant \omega_2 ... \geqslant \omega_\mu > 0, \tag{10}$$

and their default values can be found in [8].

The update of the step size and the covariance matrix uses the notion of "evolution path". The evolution path contains the correlation between consecutive steps and stores information of the previous updates. We refer to [10] for more information in this regard. We end up this section by giving a summary of CMA-ES algorithm.

1. Initialize the distribution parameters: $\mathbf{m}^{(0)}, \mathbf{C}^{(0)}, \sigma^{(0)}$.
2. Set parameters λ and μ to their default values.
3. While stopping criterion not met:
   (a) Generate new population sampled from multivariate normal distribution: $\mathbf{x}_i^{(g+1)} = \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2 \mathbf{C}^{(g)}\right) = \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right), \ i = 1, ..., \lambda$.
   (b) Update the mean value $\mathbf{m}^{(g+1)}$, the step size $\sigma^{(g+1)}$ and the covariance matrix $\mathbf{C}^{(g+1)}$.

3 Comparing EGO and CMA-ES

3.1 Test functions and experimental setup

We have employed four analytical test functions: Sphere, Ackley, Rastrigin, and Michalewicz. These functions are defined in Table 1. The Sphere function is unimodel, separable and differentiable. This function is used to observe the pure convergence speed of the algorithms. The Ackley function has many local minima with a large hole at the center which is the location of the global minimum. The Rastrigin function is highly multimodal, but locations of the minima are regularly distributed. The Michalewicz function is a multimodal function with $d!$ local minima. The parameter a exists in the Michalewicz function defines the steepness of the valleys and ridges; a larger a leads to a more difficult search.

The search space of the functions have been rescaled to $[-5, 5]^d$. $d = 2, 5, 10$ is the search space dimensionalities. The global optimum of the functions,

Table 1: Test functions

Name	Function	Defined region
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{d} (x_i)^2$	[-5.12, 5.12]
Ackley	$f(\mathbf{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right)$ $+ a - \exp(1), a = 20, b = 0.2, c = 2\pi$	[-32.768, 32.768]
Rastrigin	$f(\mathbf{x}) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	[-5.12, 5.12]
Michalewicz	$f(\mathbf{x}) = -\sum_{i=1}^{d}\sin(x_i)\sin^{2a}\left(\frac{ix_i^2}{\pi}\right), a = 10$	[0, π]

except Michalewicz, are located at $(2.5, ..., 2.5)_{1\times d}$. The total number of calls to the objective function or budget is $70 \times d$.

The initial design points of EGO are determined by Latin Hypercube Sample (LHS). The number of these points is three times the problem dimension. We repeat EGO three times on each function defined in $\mathbb{R}^d$. However, the number of repetition for CMA-ES is 10 to reduce randomness of the algorithm.

For running EGO and CMA-ES, the R packages *DiceOptim* and *cmaes* have been used, respectively. Figure 1 illustrates one run of EGO and CMA-ES on Sphere function in dimension 5. The solid line represents the function value of the points obtained by the optimization algorithm and the dashed-dotted line shows the best observed function value thus far. In the figure, EGO has an early improvement (Figure 1a) while CMA-ES converges to the minimum as the number of calls to the objective function increases (Figure 1b).

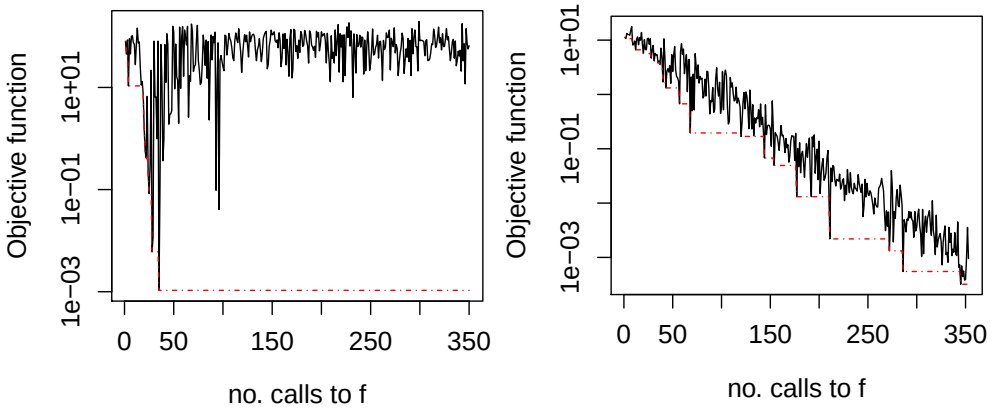
3.2 The analysis of EGO and CMA-ES

To compare EGO and CMA-ES the median of the best function values obtained by each algorithm is calculated. In addition, we consider three different starting points for CMA-ES. The results of this comparison in dimension 5 are illustrated in Figure 2. For the sake of brevity, the performance of EGO and CMA-ES in dimensions 2 and 10 is not shown here. But the results are close to Figure 2.

The analysis of the figures reveals that EGO algorithm is quick at the beginning and then slow down after some iterations. Moreover, it dos not converge to the global optimum. On the other side, CMA-ES shows a monotone improvement and tends to converge the global minimum if it does not stall in local minima.

To shed more light on the search principles of EGO and CMA-ES a visual example is used here. Figure 3 depicts the search points obtained during the optimization of Ackley function by each algorithm. What is clear from the figure is that EGO is an space-filling algorithm. It tries to find the global minimum by filling the holes in the search space, Figure 3a. However, the search points in CMA-ES algorithm tend to converge the optimum, Figure 3b.

To investigate the characteristics of the two algorithms in higher dimensions,



(a) EGO

(b) CMA-ES

Figure 1: The performance of EGO and CMA-ES on the Sphere function in dimension 5. The x-axis is the number of calls to the objective function and the y-axis is the logarithmic value of the function. The solid line shows the history of the function values observed during optimization and the dash-dotted line represents the best ever observed function value.

we use a criterion called *discrepancy*. This criterion measures how far a given distribution of points deviates from a perfectly uniform one [5]. Let $|S|$ denotes the number of points in a set S. The discrepancy of the design matrix $\mathbf{X}$ is defined by [4]:

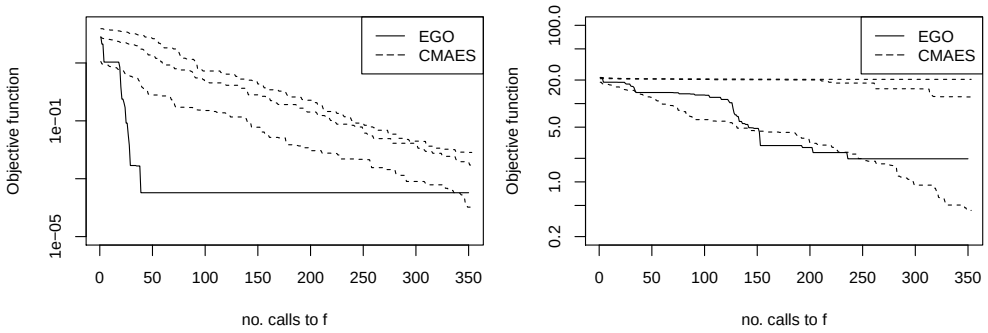
$$D(\mathbf{X}) = \left\| \frac{|\mathbf{X} \cap c^d|}{n} - Vol\left(c^d\right) \right\|, \tag{11}$$

where $\|.\|$ represents an appropriate norm over all d dimensional rectangular subsets c^d of the unit hypercube $[0,1]^d$. A small value of $D(\mathbf{X})$ means that the design point $\mathbf{X}$ is close to a uniform design.

If EGO and CMA-ES are compared based on the discrepancy criterion, the discrepancy of search points in EGO is less than CMA-ES. The reason is that while EGO tends to fill the search space, CMA-ES tries to converge to the minimum. As an example, the discrepancy of the two algorithm has been calculated on Ackley function in dimensions 5 which is 0.002 for EGO and 0.12 for CMA-ES.

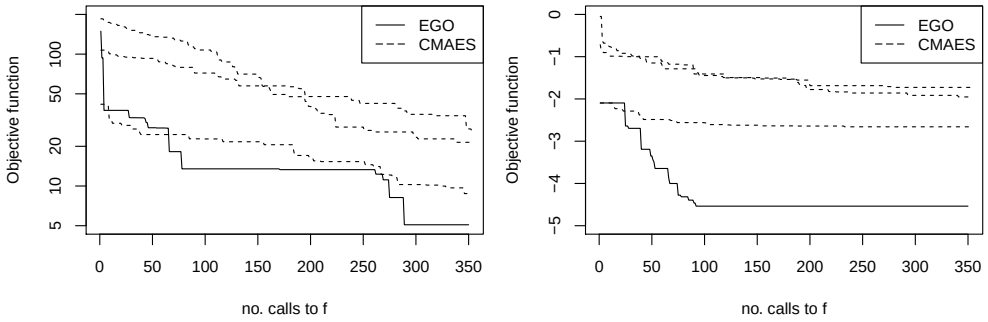
4 EGO-CMA algorithm

EGO-CMA algorithm makes benefit of both EGO and CMA-ES. The logic which is used in this algorithm is that the search space is first explored by EGO and then CMA-ES is used to converge the optimum. So in the EGO-CMA a switch take place from EGO to CMA-ES.



(a) Sphere function

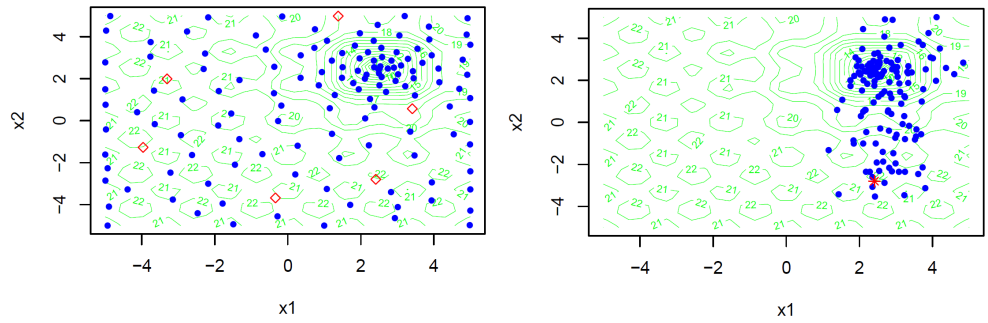
(b) Ackley function



(c) Rastrigin function

(d) Michalewicz function

Figure 2: The performance of EGO and CMA-ES on four test functions in dimension 5.



(a)

(b)

Figure 3: A 2D illustration of the difference between EGO (3a) and CMA-ES (3b). The test function is Ackley. The bullets are the points generated by the optimization algorithms. The diamonds in the leftmost picture are the initial DoE for EGO. The asterisk in the rightmost picture shows the starting point of CMA-ES.

We recall that EGO shows a quick improvement at the beginning but slows down after some budget. One reasonable stopping criterion could be the occurrence of a long plateau, with respect to the total budget, in the diagram of EGO performance. Then the best point obtained by EGO, $\mathbf{x}^{best}$, is selected and CMA-ES starts from this point. In our implementation, EGO stopped if there is no improvement after $0.15 \times budget$ calls to the objective function.

At each iteration of EGO algorithm a kriging model is fitted to the design points. In EGO-CMA we try to make use of the fitted kriging model as an approximation of the true function. In fact, EGO-CMA is started from $\mathbf{x}^{best}$ with a "good" covariance matrix and a "good" step size calculated from the points already sampled and the fitted kriging model. What is a "good" covariance matrix and step size is discussed below.

Assume that we are given a convex-quadratic objective function $f_{\mathbf{H}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x}$, where the Hessian matrix is positive definite. The optimal covariance matrix in Evolution Strategies (ESs) is a covariance matrix whose lines of the equiprobable mutation steps are aligned with the level sets of the objective function [21]. And this is the case when the covariance matrix of a search distribution is proportional to the inverse of $\mathbf{H}$.

The Hessian matrix $\mathbf{H}$ can be decomposed into:

$$\mathbf{H} = \mathbf{B}\mathbf{D}^2\mathbf{B}^\top, \tag{12}$$

where $\mathbf{B}$ is an orthogonal matrix, $\mathbf{B}^\top\mathbf{B} = \mathbf{B}\mathbf{B}^\top = \mathbf{I}$. Columns of $\mathbf{B}$ are orthonormal basis of eigenvectors. $\mathbf{D}$ is a diagonal matrix with square roots of eigenvalues of $\mathbf{H}$ as diagonal elements. Substituting the Hessian matrix by $\mathbf{B}\mathbf{D}^2\mathbf{B}^\top$ and defining variable $\mathbf{t}$ as $\mathbf{t} = \mathbf{D}\mathbf{B}^\top\mathbf{x}$, the objective function becomes $f_{\mathbf{H}}(\mathbf{t}) = \frac{1}{2}\mathbf{t}^\top\mathbf{t}$.

According to Equation (6), any search point has normal distribution. In the space of variable $\mathbf{t}$, search points have the following normal distribution

$$\begin{aligned} \mathbf{t}_i^{(g+1)} = \mathbf{D}\mathbf{B}^\top \mathcal{N}\left(\mathbf{m}^{(g)}, \sigma^2\mathbf{C}^{(g)}\right) = \\ \mathbf{D}\mathbf{B}^\top\mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}\left(\mathbf{0}, \mathbf{D}\mathbf{B}^\top\mathbf{C}^{(g)}\mathbf{B}\mathbf{D}\right),\ i = 1, ..., \lambda\ . \end{aligned} \tag{13}$$

The covariance matrix of the above distribution is the identity matrix times the step size $\sigma^{(g)}$ if and only if $\mathbf{C}^{(g)} = \mathbf{B}\mathbf{D}^{-2}\mathbf{B}^\top$ which is the inverse of $\mathbf{H}$. In other words, when the covariance matrix is proportional to the inverse of Hessian matrix, CMA-ES can be seen as locally optimizing a Spherical function.

In CMA-ES, the parameter step size σ aims at achieving fast convergence to the global optimum. If $R = \left\|\mathbf{x}^* - \mathbf{m}^{(g)}\right\|$ denotes the distance between the global minimum $\mathbf{x}^*$ and the mean vector $\mathbf{m}^{(g)}$, then the most expected value of σ achieving this goal can be calculated from

$$\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)} = \sigma^{(g)}\mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right) \ \Rightarrow\ \sigma^{*(g)} = \frac{R}{\left\|\mathcal{N}\left(\mathbf{0}, \mathbf{C}^{(g)}\right)\right\|}\ . \tag{14}$$

The initial mean vector and covariance matrix of CMA-ES in the EGO-CMA algorithm are $\mathbf{m}^{(0)} = \mathbf{x}^{best}$ and $\mathbf{C}^0 = \mathbf{H}^{-1}$. To ease the calculation of

$\sigma^{*(g)}$ in Equation (14), we prefer to work in the space of the variable $\mathbf{t}$. In this space, the initial mean vector is $\mathbf{D}\mathbf{B}^\top\mathbf{x}^{best}$ and the distance R is equivalent to $R' = \left\|\mathbf{D}\mathbf{B}^\top(\mathbf{x}^* - \mathbf{x}^{best})\right\|$. Moreover, the random variable $\frac{\mathbf{t}_i^{(g+1)} - \mathbf{D}\mathbf{B}^\top\mathbf{x}^{best}}{\sigma^{(g)}}$ has standard normal distribution. Since

$$\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \sim \mathcal{N}\left(\sqrt{d - 0.5},\ 0.5\right) , \tag{15}$$

a "good" initial step size is the one that fulfills

$$\sigma^{*(0)} = \frac{R'}{\sqrt{d - 0.5}}. \tag{16}$$

To obtain $\sigma^{*(0)}$ we need to calculate the distance R. We propose a creative approach to approximate R using Taylor expansion. The second order Taylor expansion of the objective function at point $\mathbf{x}^{best}$ is:

$$f(\mathbf{x}) \simeq f(\mathbf{x}^{best}) + \bigtriangledown f(\mathbf{x}^{best})^\top(\mathbf{x} - \mathbf{x}^{best}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{best})\mathbf{H}(\mathbf{x}^{best})(\mathbf{x} - \mathbf{x}^{best}). \tag{17}$$

Minimization of Equation (17) gives an approximation of $\mathbf{x}^*$ by which we can calculate R as follows:

$$R = \left\|\mathbf{x}^* - \mathbf{x}^{best}\right\| = \left\|-\mathbf{H}^{-1}(\mathbf{x}^{best}) \bigtriangledown f(\mathbf{x}^{best})\right\|. \tag{18}$$

If the last kriging model used for approximating true function is not convex, the Hessian matrix is not positive semidefinite. When it happens, we force the Hessian matrix to be positive semidefinite. Considering the eigenvalue decomposition in (12), we substitute the negative eigenvalues by 10^{-6}. However, this might increase the condition number of the Hessian matrix which is the ratio of the largest to the smallest eigenvalue. To improve the condition number, we add a positive value, δ, to the elements on the main diagonal of the Hessian matrix. δ which increases increases all the eigenvalues by δ can be calculated from

$$\delta = \frac{CL\lambda_{min} - \lambda_{max}}{1 - CL} , \tag{19}$$

where CL stands for condition number limit and λ_{min} and λ_{max} indicate the smallest and largest eigenvalue of the Hessian matrix. Based on our experiments we suggest to set the condition number limit CL equal to 10^3.

To sum up, a summary of EGO-CMA algorithm is provided below.

1. Start an EGO with the initial design of experiments determined by LHS.
2. When there is no further improvement after $0.15 \times$ *budget* objective function evaluations, stop EGO.
3. Select the point with the smallest function value and set it as $\mathbf{x}^{best}$.
4. Compute the Hessian matrix $\mathbf{H}$ at $\mathbf{x}^{best}$ using the last kriging model of the EGO algorithm.
5. Set $\mathbf{m}^{(0)} = \mathbf{x}^{best}$, $\mathbf{C}^{(0)} = \mathbf{H}^{-1}$ and compute $\sigma^{*(0)}$ from (16).
6. Start CMA-ES with the default values in the previous step. Stop when the budget is exhausted.

5 Simulation Results

The performance of EGO-CMA is tested here with the default parameter setting explained in Section 4. Each run of EGO-CMA is repeated 5 times on each function and then the results are compared with EGO and CMA-ES. We again recall that each curve is the median of the repetitions. Also, we only consider the CMA-ES with the starting point that leads to superior performance. The results of this comparison in dimension 5 and 10 are illustrated in Figures 4 and 5, respectively.

The results show that EGO-CMA outperforms CMA-ES in all functions and also EGO except the Rastrigin function in dimension 5. In this case, EGO-CMA switches to CMA-ES before EGO detects the global minimum location. Note that when budget is greater than 250, EGO improves the objective function value.

In the case of the Sphere function, EGO can roughly detect the location of global minimum quickly which allows EGO-CMA to further increase the accuracy. The accuracy of EGO-CMA is about 10^{-8} and 10^{-5} for the Sphere function in dimension 5 and 10, respectively.

6 Conclusions

This paper presents a new algorithm, called EGO-CMA, for unconstrained continuous black-box optimization. EGO-CMA combines the strengths of EGO and CMA-ES: while EGO is a space-filling strategy, CMA-ES is a robust local search.

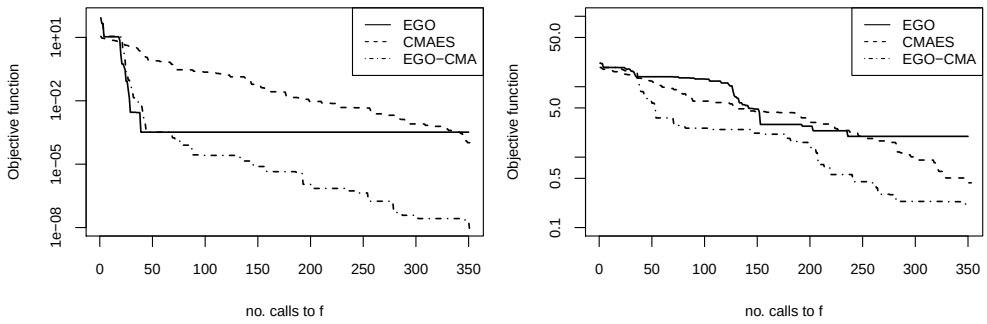
In the proposed algorithm, search space is first explored by EGO and then a switch to CMA-ES take place. CMA-ES is started from the point with the smallest function value obtained by EGO. Moreover, the initial covariance matrix and the step size of CMA-ES are calculated from the design points already sampled and the kriging model of the EGO algorithm. Therefore, The cooperation between the two algorithms goes beyond a plain succession as the Gaussian process learned by EGO allows improving the initialization of key parameters of CMA-ES. The results of our 2 to 10 dimensional experiments show that EGO-CMA outperforms EGO and CMA-ES for the same budget.

Acknowledgement

The authors would like to acknowledge support by the French national research agency (ANR) within the Modèles Numérique project "NumBBO- Analysis, Improvement and Evaluation of Numerical Blackbox Optimizers".

References

[1] A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary*



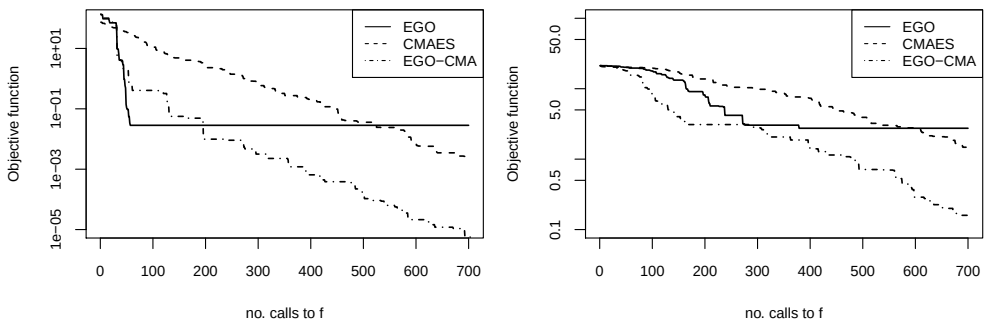
(a) Sphere function

(b) Ackley function

(c) Rastrigin function

(d) Michalewicz function

Figure 4: The comparison of EGO, CMA-ES and EGO-CMA on four test functions in dimension 5.



(a) Sphere function

(b) Ackley function

(c) Rastrigin function

(d) Michalewicz function

Figure 5: The comparison of EGO, CMA-ES and EGO-CMA on four test functions in dimension 10.

Computation, volume 2, pages 1769–1776, Piscataway, NJ, USA, 2005. IEEE Press.

[2] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.

[3] Dirk Bueche, Nicol N. Schraudolph, and Petros Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics*, 35:183–194, 2004.

[4] Keith R. Dalbey and George N. Karystinos. Fast generation of space-filling latin hypercube sample designs. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010.

[5] Delphine Dupuy, Celine Helbert, and Jessica Franco. DiceDesign and DiceEval: two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 2011.

[6] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395. ACM, July 2009.

[7] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312 –317, May 1996.

[8] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*, January 2009.

[9] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, pages 1689–1696, New York, NY, USA, 2010. ACM.

[10] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[11] Nikolaus Hansen, Andreas Ostermeier, and Andreas Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *Sixth International Conference on Genetic Algorithms*, pages 312–317. Morgan Kaufmann, 1995.

[12] Yaochu Jin. Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges. *Swarm and Evolutionary Computation*, pages 61–70, 2011.

[13] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[14] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, December 1998.

[15] Stefan Kern, Nikolaus Hansen, and Petros Koumoutsakos. Local metamodels for optimization using evolution strategies. In *Parallel Problem Solving from Nature - PPSN IX*, pages 939–948. Springer, 2006.

[16] J.W. Kruisselbrink, M.T.M. Emmerich, A.H. Deutz, and T. Baeck. A robust optimization approach using Kriging metamodels for robustness approximation in the CMA-ES. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

[17] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Self-Adaptive Surrogate-Assisted Covariance Matrix Adaptation Evolution Strategy. In T. Soule et al., editor, *Genetic and Evolutionary Computation Conference (GECCO)*, pages 321–328. ACM Press, July 2012.

[18] Ilya Loshchilov, Marc Schoenauer, and Michele Sebag. Intensive Surrogate Model Exploitation in Self-adaptive Surrogate-assisted CMA-ES (saACM-ES). In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 439–446. ACM Press, July 2013.

[19] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[20] Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[21] G. Rudolph. On correlated mutations in evolution strategies. In *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, pages 107–116. North-Holland, Amsterdam, 1992.

[22] H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by gaussian processes with improved pre-selection criterion. In *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, Canberra, Australia*, pages 692–699. IEEE Press, 2003.