BCMA-ES: a conjugate prior Bayesian optimization view

Eric Benhamou, David Saltiel, Rida Laraki, Jamal Atif

BCMA-ES: a conjugate prior Bayesian optimization view

Eric Benhamou[1,2], David Saltiel[2,3], Rida Laraki[1], and Jamal Atif[1]

[1] LAMSADE, Dauphine University, France
[2] Ai Square Connect, France
[3] ULCO - LISIC, France

Abstract. CMA-ES is one of the state of the art evolutionary optimization methods because of its capacity to adapt covariance to information geometry. It uses prior information to form a best guess about the distribution of the minimum. We show this can be reformulated as a Bayesian optimization problem for the sampling of the optimum. Thanks to Normal Inverse Wishart (NIW) distribution, that is a conjugate prior for the multi variate normal distribution, we can derive a numerically efficient algorithm Bayesian CMA-ES that obtains similar performance as the traditional CMA-ES on multiple benchmarks and provides a new justification for the CMA-ES updates equations. This novel paradigm for Bayesian CMA-ES provides a powerful bridge between evolutionary and Bayesian optimization, showing the profound similarities and connections between these traditionally opposed methods and opening horizon for variations and mix strategies on these methods.

Keywords: CMA-ES · Bayesian optimization · Normal inverse Wishart · conjugate prior

1 Introduction

The covariance matrix adaptation evolution strategy (CMA-ES) [9] and all its variants are arguably one of the most powerful evolutionary optimization algorithms, finding many applications in machine learning. It is a state-of-the-art optimizer for continuous black-box functions as shown by the various benchmarks of the COmparing Continuous Optimisers (http://coco.gforge.inria.fr) INRIA platform for ill-posed functions. It has led to a large number of papers and articles and we refer the interested reader to [9,11,8,2] and the recent publications in GECCO 2019 [15,7,16].

Briefly, the (μ / λ) CMA-ES is an iterative black box optimization algorithm, that, at each of its iterations, samples λ candidate solutions according to a multivariate normal distribution, evaluates these solutions, retains μ candidates and adjusts the multi variate normal mean and covariance for the next iteration. Each iteration makes an initial guess or *prior* for the distribution parameters (mean and covariance of the multi-variate normal), evaluates the fit function and updates parameters thanks to the revised guess or *posterior*. CMA-ES

has been historically justified using Information Geometry Optimization [14] and [1]. However, because we make some initial guess about parameters of a distribution and then revise it, we could revisit CMA-ES with the light of Bayesian Optimization (BO). The linkage is however not immediate as traditional BO does not place prior (and posterior) on the optimum but rather on the objective function. More specifically, if we aim at solving the following optimization problem:

$$x^\star = \underset{x \in \mathcal{X}}{argmin} f(x)$$

where f is called a black-box function (expensive evaluation of f, and potentially no ability to observe or compute the gradients of f). In this setting, BO treats f as a random function and places a prior over it. It then improves it at candidate samples to compute a posterior function than in turns becomes the next prior and so on until convergence. BO also uses the terminology of surrogate and acquisition function. It repeatedly builds a probabilistic model for the objective function, called the surrogate function $\hat{f}$, then search efficiently with an acquisition function for candidate samples to improve our probabilistic model and start again building a probabilistic model but now at the new points and so on until convergence. The probabilistic model for the objective function aims at inferring our objective function with the minimum number of evaluation of the objective function as it is expensive to evaluate it. The surrogate function is computed thanks to a probabilistic regression model regression. This surrogate function embeds uncertainty for the objective function. The probabilistic regression model is very often using Gaussian process but can also be using random forest or tree Parzen estimator. This comes at a cost, as the regression step requires solving an embedded costly optimization to find the regression parameters. In this work, we propose to change dramatically the point of view while keeping the spirit of a prior and posterior than in turns becomes the next prior until convergence. Instead of inferring the objective function thanks to the surrogate function, we rather aim at inferring the distribution of the optimum, hence the name of Bayesian Optimization with Optimum Simulation (BOOS).

1.1 Contributions

- In this work, we propose a new paradigm to circumvent this embedded optimization by directly simulating the distribution of the optimum. This may seem paradoxical at first sight as the aim of the global optimization is precisely to find this optimum. How could one simulate the final result of the optimization? We show it is possible to compute prior and posterior distributions of the optimum through a multi variate Gaussian distribution and an appropriate conjugate prior for our retained stochastic optimum, scarce evaluation of the objective function with a strategy combining exploration and exploitation thanks to sub-sampling, weighting and re-sampling.
- We provide theoretical justification for doing it thanks to conjugate prior analysis. We also relate this approach to natural gradient, emphasizing the tight link between this modified BO method and gradient descent methods.

- We stress that this novel paradigm for Bayesian optimization provides a powerful bridge between Bayesian and evolutionary optimization revealing the profound resemblances and parallels between these traditionally opposed methods and opening horizon for variations and mix strategies on these methods.
- We evaluate our new method on multiple benchmarks and obtain state-of-the-art performance on stylized experiments

An implementation of BOOS is available at `https://github.com/aisquareconnect/bcmaes`

1.2 Related works

Alternative to BO and new way to find surrogate function has led to multiple works in recent years [13], [12] and [5]. They keep the original BO framework of creating a surrogate function for the objective function but in a lower dimensional space. Initially, the dimension problem was tackle by Gaussian matrix projections and structure learning.

Recently, research has headed towards better subspace thanks to active subspace theory. Hence, [13] reduces the burden computation of the embedded optimization for the surrogate thanks to a projection to a lower dimensional active subspace. They show that the embedded subspace do not deteriorate too much the Gaussian process model as the model error is tightly bounded thanks to the property of the active subspace that provides better performance than traditional Gaussian matrix projections and structure learning.

Similarly, [12] proposes an algorithm (LINEBO) that restricts the problem to a sequence of iteratively chosen one dimensional sub-problems that can be solved efficiently and can be seen as active subspace. Likewise, [5] has changed the method to find a local probabilistic model where the corresponding active subspace is found via an implicit bandit approach.

However, to our knowledge, no one has ever looked at the idea of sampling the optimum and not the function itself.

2 Framework

In this section we recall how Bayesian optimization works. Let us define and objective function f that is defined in a search space $\mathcal{X}$: $f : \mathcal{X} \mapsto \mathbb{R}$. We want to find the global minimum of this black box function $x^* = arg\min_{x \in \mathcal{X}} f(x)$. Based on the Bayes theorem, Bayesian optimization aims at progressively learning the objective function f to locate its global minimum by iteratively improving a probabilistic model that encodes the current knowledge of the objective function and provides probabilistic indication where to sample next. In traditional Bayes statistics, given specific samples $(x_1, ..., x_n)$ and their corresponding objective function value $f(x_1), ..., f(x_n)$, the posterior distribution is the conditional distribution of $f : \mathbb{P}(f|\mathcal{D})$ given all samples $\mathcal{D} = (x_i, f(x_i))_{i=1...n}$ and is proportional

to the prior distribution times the likelihood as follows: $\mathbb{P}(f|\mathcal{D}) \propto \mathbb{P}(\mathcal{D}|f) \times \mathbb{P}(f)$ or said differently $Posterior \propto Likelihood \times Prior$. The posterior encodes all our knowledge of the objective function. Called the surrogate (of the objective) function, this prior acts as the best approximation of the objective function given our current sample knowledge $f|\mathcal{D}$ and is used to direct future sampling. The Bayesian approach incorporates in the surrogate, the uncertainty of the current knowledge of the objective function which is used for acquiring more samples thanks to the *acquisition* function. There are multiple design choices for the acquisition function (Probability of Improvement (PI), Expected Improvement (EI), Lower Confidence Bound (LCB), etc). The Acquisition function (often set to a Gaussian Process regression [1] effectively leverages our belief about the objective function to sample the area of the search space. We do a Copernican revolution and aim now at modeling rather the distribution of

$$x^\star = \underset{x \in \mathcal{X}}{argmin} f(x)$$

This may seem a daunting task as the goal of the Bayesian optimization is precisely to infer $x^\star$. How could one start by the end? It looks like a complete non sense at first sight and unfeasible. This first intuition turns out to be wrong and misleading.

- First of all, although we do not know $x^\star$, we can still make assumptions on the distribution of $x^\star$.
- Secondly, adding and modeling uncertainty on f, which is what Bayesian optimization is doing is, is not directly addressing the goal of finding its minimum. It is rather using a circonvoluted approach that takes detours to find the right solution.
- Thirdly, adding uncertainty on the function f itself is adding much more uncertainty as the functional space of functions $f : \mathcal{X} \mapsto \mathbb{R}$ is way bigger than the space where lies the minimum $\mathcal{X}$. [4] is indeed famous (with its diagonal argument) to have shown under the Continuum hypothesis that the cardinality of the functional space $f : \mathbb{R} \mapsto \mathbb{R}$ is $\aleph_2$ which is infinitely larger than the one of $\mathbb{R}$ that is only $\aleph_1$. Indeed, $2^{\aleph_1} = \aleph_2$. Saying simply, if we assume that $\mathbb{R}$ can be discretized into 10,00 potential values (this is a finite approximation of an infinite set), the cardinality of the functions of $f : \mathbb{R} \mapsto \mathbb{R}$ is $10,000^{10,1000} = 10^{4.10^4}$ which shows how much bigger on finite sets (1 followed by 40 thousands zeros compared to 1 followed by 4 zeros for the domain definition!), the functional space is compared to the domain definition! Hence modeling uncertainty on the functional space is very ineffective compared to modeling uncertainty on the minimum.

We will show now how we can apprehend having uncertainty on $x^\star$.

[1] Gaussian Process regression assumes that the distribution of $f|\mathcal{D}$ is a Gaussian process with mean $\lambda_{f|\mathcal{D}}$ and variance $\Sigma_{f|\mathcal{D}}$

2.1 Notations

We will iterate over a number of iterations. At iteration t, we will draw n samples candidate points for the optimum written as $\mathcal{X}_t = (x_1^t, ..., x_n^t)$ and evaluate the sample $f(x_i^t)_{i=1,...,n}$. The straight sample mean and variance are given by :

$$\overline{x}_t = \frac{1}{n}\sum_{i=1}^{n} x_i^t \tag{1}$$

$$\overline{\sigma}_t^2 = \frac{1}{n-1}\sum_{i=1}^{n} \left(x_i^t - \overline{x}_t\right)\left(x_i^t - \overline{x}_t\right)^T \tag{2}$$

Later in subsection 2.4, we will see how we compute the sample mean and variance using information about the objective function f.

2.2 Modeling uncertainty on the minimum

Let us assume that $x^\star$ follows a distribution whose parameters are unknown. Like in traditional Bayesian optimization, it is worth using a tractable distribution. We will hence assume that $x^\star$ follows a multi variate Gaussian distribution. To represent uncertainty on the parameters of the multi variate Gaussian distribution, it is also worth using a conjugate prior as this will ease computation. We denote by $\pi(\theta)$ the distribution on $x^\star$ parametrized by parameters θ.

Definition 1. *A prior distribution $\pi(\theta)$ is said to be a conjugate prior for the if the posterior distribution $\pi(\theta|\mathcal{D})$ remains in the same distribution family as the prior.*

It is also a well known result (see for instance [3] that a good candidate for a conjugate prior for the multi variate normal, $\mathcal{N}(.;\mu,\Sigma)$, is the normal inverse Wishart distribution, parametrized by $\lambda_0, \kappa_0, \nu_0, \psi$ whose formula is $f(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\lambda_0,\kappa_0,\nu_0,\psi) = \mathcal{N}\left(\boldsymbol{\mu}\middle|\lambda_0, \frac{1}{\kappa_0}\boldsymbol{\Sigma}\right)\mathcal{W}^{-1}(\boldsymbol{\Sigma}|\psi,\nu_0)$, where $\mathcal{W}^{-1}$ is the inverse Wishart and $\mathcal{N}$ a multi variate normal. Because the update is the optimal one given the prior information, the Bayesian updates should work well in initial iterations. Numerical experiments confirm this intuition.

As stated in the definition 1, conjugate prior means that the posterior distribution is also a normal-inverse-Wishart with updated parameters $\text{NIW}(\lambda_0^\star, \kappa_0^\star, v_0^\star, \psi^\star)$ given by:

$$\begin{aligned} \lambda_0^\star &= \frac{\kappa_0\lambda_0 + n\overline{x}}{\kappa_0 + n}, \qquad \kappa_0^\star = \kappa_0 + n, \qquad v_0^\star = v_0 + n \\ \psi^\star &= \psi + (n-1)\overline{\sigma}^2 + \frac{\kappa_0 n}{\kappa_0 + n}\left(\overline{x} - \lambda_0\right)\left(\overline{x} - \lambda_0\right)^T \end{aligned} \tag{3}$$

with $\overline{x}$ the sample mean: $\overline{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\overline{\sigma}^2$ the sample variance given by $\overline{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^n \left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T$. Hence, if we are able to estimate both the sample mean and variance, we can update the parameters of the prior to get the posterior. This is the key property to update mean and covariance in BOOS.

2.3 Simulating the minimum

To simulate the minimum $x^\star$ and not f is the key of this new Bayesian optimization. Let us for the moment assume that we are able to get the right sample mean and variance (by some oracle). We will come back to this point in subsection 2.4.

First of all, theoretically, to simulate the new sample, we should first simulate the uncertainty on the parameters $\lambda_t, \kappa_t, \nu_t, \psi_t$ and given these parameters do another simulation on $\mathcal{N}(.|\mu_t, \Sigma_t)$ given $\lambda_t, \kappa_t, \nu_t, \psi_t$. This will require doing simulations within simulations making this algorithm quite slow. A first trick is instead to use the mean value (using a mean field approximation) and compute the conditional expectation given $\lambda_t, \kappa_t, \nu_t, \psi_t$ for the multi variate normal parameters:

Proposition 1. *We can compute the mean variance approximation of the mean and covariance of the multi variate normal as follows:*

$$\mu_t = \mathbb{E}[\mu|\lambda_t, \kappa_t, \nu_t, \psi_t] = \lambda_t \tag{4}$$

$$\textit{and } \Sigma_t = \mathbb{E}[\Sigma|\lambda_t, \kappa_t, \nu_t, \psi_t] = \frac{\psi_t}{v_t - n - 1} \tag{5}$$

Proof. See Supplementary 6.1.

Hence instead of doing simulations within simulations, we can just simulate a multi variate normal whose parameters are μ_t and Σ_t. Combining equations (4), and (3), we obtain the recursive update equations summarized by the proposition below.

Proposition 2. *Using recursion, at iteration t, we can compute the mean variance approximation of the mean and covariance of the multi variate normal as follows:*

$$\mu_t = \frac{\kappa_0 \lambda_0}{\kappa_0 + tn} + \frac{n \sum_{j=0}^{t-1} \overline{x}_j}{\kappa_0 + tn}, \tag{6}$$

$$\begin{aligned} \textit{and } \Sigma_t = & \frac{\psi_0}{v_0 + (t-1)n - 1} \\ & + \frac{\sum_{j=0}^{t-1} \frac{\kappa_j n}{\kappa_j + n} (\overline{x}_j - \mu_j)(\overline{x}_j - \mu_j)^T}{v_0 + (t-1)n - 1} \\ & + \frac{(n-1)\sum_{j=0}^{t-1} \overline{\sigma}_j^2}{v_0 + (t-1)n - 1} \end{aligned} \tag{7}$$

Proof. See supplementary 6.2.

We can now prove the convergence of this approach as follows.

Proposition 3. *Under the assumption that the samples $x_1, ..., x_{...}$ are independent and identically distributed with finite first fourth centered moments and with the first two centered moments given by μ_∞ and Σ_∞, we obtain that the BOOS scheme converges to the right normal distribution almost surely, that is*

$$\mu_t \xrightarrow[t \to \infty]{a.s.} \mu_\infty \tag{8}$$

$$\textit{and } \Sigma_t \xrightarrow[t \to \infty]{a.s.} \Sigma_\infty \tag{9}$$

where $\Sigma_\infty = \lim_{t\to\infty} \frac{(n-1)\sum_{j=0}^{t-1} \overline{\sigma}_j^2}{v_0+(t-1)n-1}$.

Proof. See supplementary 6.3

2.4 Using the objective function

All the above schemes assume that we are able to simulate the likelihood and more precisely compute the sample mean and variance. It is remarkable that we only need the sample mean and variance according to the likelihood and no other moments. However, as this is precisely the objective of our optimization method, we are facing some kind of circularity. We will show how we can break this circularity and have good guess about the sample mean and variance at iteration t. At iteration t, we draw sample candidate points, $\mathcal{X}_t = (x_1^t, ..., x_n^t)$ and evaluate the sample $f(x_i^t)_{i=1,...,n}$. If we were completely accurate, the minimum of f should be located in μ_t with some variance Σ_t. However, the evaluation of the function gives us some indication that this not completely accurate and that we need to revise our initial guess.

For each candidate point we compute its assumed density $d_i^t = \mathcal{N}(x_i^t, \mu_t, \Sigma_t)$ where $\mathcal{N}(., \mu_t, \Sigma_t)$ denotes the p.d.f. of the multi variate normal distribution. We divide these density by their sum to get weights $(w_i^t)_{i=1..n}$ that are positive and sum to one as follows: $w_i^t = \frac{d_i^t}{\sum_{j=1}^n d_j^t}$. Let us denote by $\{x_{(i),f\uparrow}^t\}_{i=1..n}$ the samples $(x_i^t)_{i=1..n}$ sorted according to their objective function value in increasing order: $f(x_{(1),f\uparrow}^t) \leq f(x_{(2),f\uparrow}^t) \leq ... \leq f(x_{(n),f\uparrow}^t)$. Let us denote by $\{w_{(i),w\downarrow}^t\}_{i=1..n}$ the weights sorted in decreasing order : $w_{(1),w\downarrow}^t \geq w_{(2),w\downarrow}^t \geq ... \geq w_{(n),w\downarrow}^t$

Hence for n simulated points, we can have a two columns matrix that is transformed into another two columns matrix: simulataneous sorting in f in increasing order and w in decreasing order independently.

$$\begin{pmatrix} x_1^t, & w_1^t \\ ... & ... \\ x_n^t, & w_n^t \end{pmatrix} \rightarrow \begin{pmatrix} x_{(1),f\uparrow}^t, & w_{(1),w\downarrow}^t \\ ... & ... \\ x_{(n),f\uparrow}^t, & w_{(n),w\downarrow}^t \end{pmatrix} \tag{10}$$

We can now compute the empirical mean $\overline{x}_t$ as follows:

$$\overline{x}_t = \sum_{i=1}^n w_{(i),w\downarrow}^t \cdot x_{(i),f\uparrow}^t \tag{11}$$

For the sample variance at iteration t, we do a similar strategy

$$\overline{\sigma}_t = \sum_{i=1}^n w_{(i),w\downarrow}^t \cdot \left(x_{(i),f\uparrow}^t - \overline{x}_t\right)\left(x_{(i),f\uparrow}^t - \overline{x}_t\right)^T \tag{12}$$

Compared to standard CMA -ES that weights sample candidate with a logarithmic decrease, we here use the distribution at each step to overweight best candidates.

3 Link with CMA ES

It is striking that it provides very similar formulae to the standard CMA-ES update. Recall that these updates given for the mean m_t and covariance C_t can

be written as follows:

$$\begin{aligned}
\mu_{t+1} &= \mu_t + c_m \left(\overline{x}_t^{\lambda,\mu} - \mu_t \right) \\
\Sigma_{t+1} &= \underbrace{(1 - c_1 - c_\mu + c_s)}_{\text{discount factor}} \Sigma_t + c_1 \underbrace{p_c p_c^T}_{\text{rank one matrix}} \\
&\quad + c_\mu \underbrace{\sum_{i=1}^{\mu} w_i \frac{x_{i:\lambda} - m_k}{\sigma_k} \left(\frac{x_{i:\lambda} - m_t}{\sigma_t} \right)^T}_{\text{rank } \min(\mu, n-1) \text{ matrix}}
\end{aligned} \tag{13}$$

where $\overline{x}_t^{\lambda,\mu} = \sum_{i=1}^{\mu} w_i \, x_{i:\lambda}$ is the CMA-ES sample mean and the notations $\mu_t = m_t, w_i, c_m, x_{i:\lambda}, \Sigma_t = C_t, c_1, c_\mu, c_s$, etc... are given for instance in [6]. Our scheme using normal-inverse Wishart update 3 leads to very similar equations:

$$\begin{aligned}
\mu_{t+1} &= \mu_t + w_t^\mu \left(\overline{x}_t - \mu_t \right), \\
\Sigma_{t+1} &= \underbrace{w_t^{\Sigma,1}}_{\text{discount factor}} \Sigma_t + w_t^{\Sigma,2} \underbrace{\left(\overline{x}_t - \mu_t \right) \left(\overline{x}_t - \mu_t \right)^T}_{\text{rank one matrix}} \\
&\quad + w_t^{\Sigma,3} \underbrace{\overline{\sigma}_t}_{\text{rank (n-1) matrix}}
\end{aligned}$$

$$\begin{aligned}
\text{where} \qquad w_t^\mu &= \frac{n}{\kappa_t + n}, \\
w_t^{\Sigma,1} &= \frac{v_t - n - 1}{v_t - 1}, \\
w_t^{\Sigma,2} &= \frac{\kappa_t n}{(\kappa_t + n)(v_t - 1)}, \\
w_t^{\Sigma,3} &= \frac{n - 1}{v_t - 1}
\end{aligned} \tag{14}$$

where the sample mean and variance are given in equations (11) and (12)

Proof. See supplementary 6.4

4 Experiments

We compare Bayesian CMA-ES (BCMA-ES) to standard CMA-ES on four 2-dimensional traditional functions: Sphere (also known as the Cone), Rastrigin, Schwefel 1 and 2 (see for instance [10] for their definition). Because of the importance of the seed in these two algorithms, we take the average convergence and a confidence interval over 30 function evaluations and multiple starting points.

This is shown in the various figures 1, 2, 3 and 4. The standard CMA-ES used is the one provided in the open source python package cma and displayed in *blue*, while BCMA-ES is in *red*. Full details of the implementation of the experiment is provided in supplementary materials with python source code.

What we can notice is that Bayesian CMA-ES achieves initially faster convergence. As we increase the number of iterations, standard CMA-ES performs

better and better and start performing faster convergence for around 30 iterations. We will need to investigate why this is the case and if we could adapt the computation of our sample mean and variance as given by equations (11) and (12) to benefit from the performance of standard CMA-ES. This would lead to a mixed strategy that uses initially Bayesian CMA-ES and then standard CMA-ES sample mean and variance equations.
We can summarize the different results for our four functions to see the impact of the starting points and achieve an initial faster convergence. The results are shown in table 1.

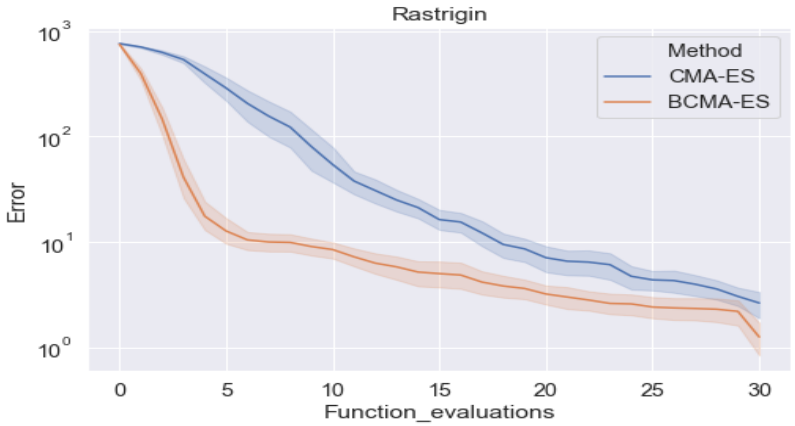

Fig. 1. Convergence for the Rastrigin function with a starting point at [-20,-20]

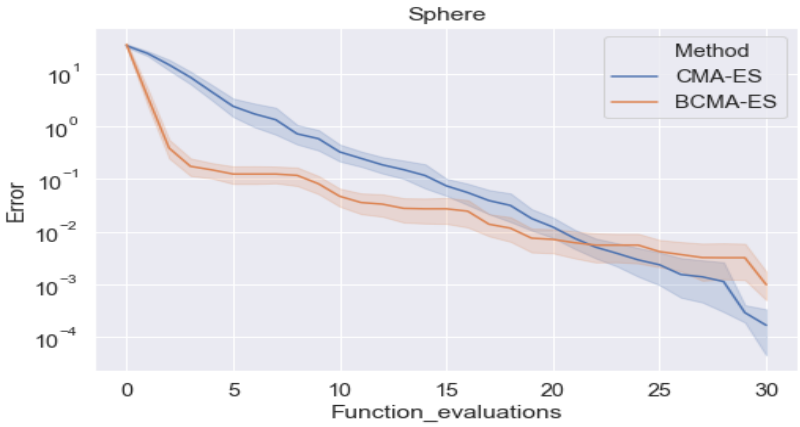

Fig. 2. Convergence for the Sphere function with a starting point at [5,5]

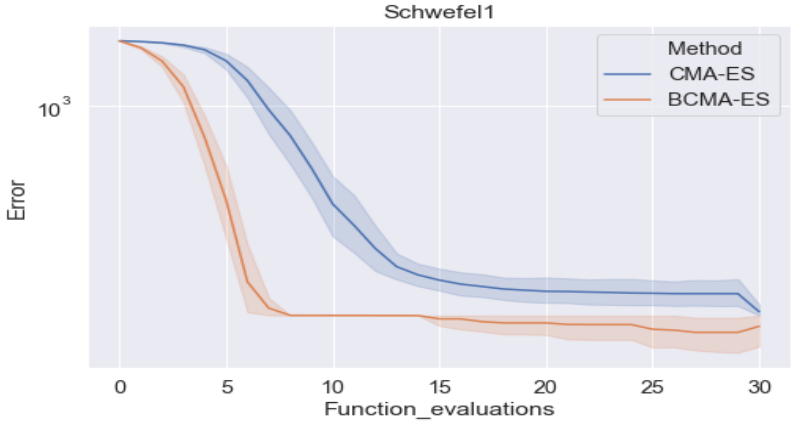

Fig. 3. Convergence for the Schwefel 1 function with a starting point at [-400,-400]

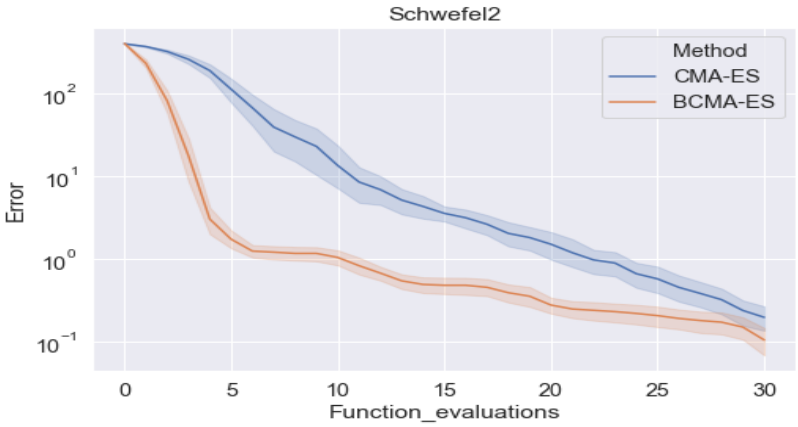

Fig. 4. Convergence for the Schwefel 2 function with a starting point at [20,20]

For each functions, we started at a point that is not very close to the optimum. We are particularly interested in measuring the robustness of the optimization algorithm should we start far away from the optimum. To have meaningful graphics, the convergence is plotted in standard logarithmic scale. The thin confindence interval indicates that the two algorithms are experimentally not very sensitive to the random seed. The x axis represents the iterations in our two algorithms while the y axis the difference between the value of the objective function in the optimum and the value of the objective function estimated at our best candidates in log term. We have provided in supplementary materials other comparison plots for different starting points: see section 7. This shows that the results remain the same regardless of the starting points and that Bayesian CMA-ES is in general quite robust to a starting point far away from the optimum. All the results are however summarized in table 1.

5 Conclusion

Here, we have revisited CMA-ES and provide a Bayesian version of it. Taking a conjugate prior, we find the optimal update for the mean and variance. We proivde ways to incorporate objective function feedback to compute sample mean and variance. Numerical experiments show that this new version is competitive to standard CMA-ES on traditional functions such as Sphere, Schwefel 1, Rastrigin and Schwefel 2. The initial faster convergence is due to the Bayesian optimal posterior update. Further work should examine why CMA-ES continues increasing the rate of variance contraction while BCMA-ES does not achieve it.

Table 1. Comparison results for multiple starting points

function	starting point	cma error	bcma error	cma error / bcma error
Rastrigin	[-20 -20]	135.72	48.39	35.7%
	[-10 -10]	31.31	12.35	39.4%
	[-5 -5]	11.49	4.81	41.9%
	[5 5]	10.60	5.23	49.3%
	[10 10]	30.31	13.19	43.5%
	[20 20]	131.38	52.11	39.7%
Sphere	[-20 -20]	116.60	40.54	34.8%
	[-10 -10]	20.78	6.92	33.3%
	[-5 -5]	3.36	1.24	36.9%
	[5 5]	3.09	1.35	43.6%
	[10 10]	20.02	7.62	38.1%
	[20 20]	115.64	43.97	38.0%
Schwefel1	[-400 -400]	618.67	409.05	66.1%
	[-200 -200]	704.56	388.25	55.1%
	[-100 -100]	614.77	312.35	50.8%
	[100 100]	734.49	256.62	34.9%
	[200 200]	434.48	365.37	84.1%
	[400 400]	16.11	5.68	35.3%
Schwefel2	[-20 -20]	63.11	23.18	36.7%
	[-10 -10]	12.85	4.58	35.6%
	[-5 -5]	2.68	1.14	42.5%
	[5 5]	2.33	1.15	49.2%
	[10 10]	12.15	4.76	39.2%
	[20 20]	60.96	24.52	40.2%

References

1. Akimoto, Y., Auger, A., Hansen, N.: Comparison-based natural gradient optimization in high dimension. GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference (07 2014)
2. Akimoto, Y., Auger, A., Hansen, N.: CMA-ES and advanced adaptation mechanisms. GECCO, Denver **2016**, 533–562 (2016)
3. Benhamou, E., Saltiel, D., Vérel, S., Teytaud, F.: BCMA-ES: A bayesian approach to CMA-ES. CoRR (2019)
4. Cantor, G.: Uber eine elementare frage der mannigfaltigskeitslehre. Jahresber. Deutsch. Math. Vereinig. **1**, 75–78 (1891)
5. Eriksson, D., Pearce, M., Gardner, J., Turner, R.D., Poloczek, M.: Scalable global optimization via local bayesian optimization. Proceedings of NIPS 2019 (2019)
6. Hansen, N.: The CMA evolution strategy: A tutorial. CoRR (2016), `http://arxiv.org/abs/1604.00772`
7. Hansen, N.: A Global Surrogate Assisted CMA-ES. In: GECCO 2019. pp. 664–672. ACM, Prague, Czech Republic (Jul 2019)
8. Hansen, N., Auger, A.: CMA-ES: evolution strategies and covariance matrix adaptation. GECCO 2011 pp. 991–1010 (2011)
9. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)
10. Hussain, K., Salleh, M., Cheng, S., Naseem, R.: Common benchmark functions for metaheuristic evaluation: A review. International Journal on Informatics Visualization **1**, 218–223 (11 2017). https://doi.org/10.30630/joiv.1.4-2.65
11. Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. Evol. Comput. **15**(1), 1–28 (Mar 2007)
12. Kirschner, J., Mutny, M., Hiller, N., Ischebeck, R., Krause, A.: Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. Proceedings of IMCL 2019 (2019)
13. Nayebi, A., Munteanu, A., Poloczek, M.: A framework for bayesian optimization n embedded subspaces. Proceedings of IMCL 2019 (2019)
14. Ollivier, Y., Arnold, L., Auger, A., Hansen, N.: Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. Journal of Machine Learning Research **18**(18), 1–65 (2017)
15. Touré, C., Hansen, N., Auger, A., Brockhoff, D.: Uncrowded Hypervolume Improvement: COMO-CMA-ES and the Sofomore framework. In: GECCO 2019. Prague, Czech Republic (Jul 2019)
16. Vermetten, D., van Rijn, S., Back, T., Doerr, C.: Online selection of CMA-ES variants. In: GECCO 2019. pp. 951–959. Prague (Jul 2019)

6 Supplementary materials for BCMA-ES: a conjugate prior Bayesian optimization view

6.1 proof of proposition 1

Recall that under the normal inverse Wishart distribution, the mean μ follows a normal distribution whose parameters are λ_t and $\frac{1}{\kappa_t}\Sigma$. Hence we, have

$$\begin{aligned}\mu_t &= \mathbb{E}[\mu|\lambda_t, \kappa_t, \nu_t, \psi_t] \\ &= \mathbb{E}[\mathcal{N}(.|\lambda_t, \frac{1}{\kappa_t}\Sigma)] = \lambda_t \end{aligned} \tag{15}$$

Likewise, under the normal inverse Wishart distribution, the variance Σ follows an inverse Wishart distribution whose parameters are ψ_t, ν_t with an expectation given by $\frac{\psi_t}{v_t - n - 1}$. Hence, we have

$$\begin{aligned}\Sigma_t &= \mathbb{E}[\Sigma|\lambda_t, \kappa_t, \nu_t, \psi_t] \\ &= \mathbb{E}[\mathcal{W}^{-1}(.|\psi_t, \nu_t)] = \frac{\psi_t}{v_t - n - 1} \end{aligned} \tag{16}$$

which concludes the proof. □

6.2 proof of proposition 2

Using recursion, it is trivial to prove that

$$\kappa_t = \kappa_0 + tn, \tag{17}$$

$$\kappa_t \lambda_t = \kappa_0 \lambda_0 + n \sum_{j=0}^{t-1} \overline{x}_j \tag{18}$$

which provides the first expression for μ_t.
Likewise, to get the *mean variance* approximation of the covariance Σ_t, we can notice that

$$\nu_t = \nu_0 + tn, \tag{19}$$

$$\begin{aligned}\psi_t = \psi_0 &+ \sum_{j=0}^{t-1} \frac{\kappa_j n}{\kappa_j + n} (\overline{x}_j - \lambda_j)(\overline{x}_j - \lambda_j)^T \\ &+ (n-1) \sum_{j=0}^{t-1} \overline{\sigma}_j^2 \end{aligned} \tag{20}$$

which concludes the proof by noticing that $\lambda_j = \mu_j$. □

6.3 proof of proposition 3

The samples $x_1, ..., x_n$ are independent and identically distributed random vector with finite first two centered moments μ_∞ and Σ_∞. Using the strong law of large

numbers, we have

$$\frac{n\sum_{j=0}^{t-1}\overline{x}_j}{\kappa_0+tn}=\frac{tn}{\kappa_0+tn}\frac{\sum_{j=1}^{tn}x_i}{tn}\xrightarrow[t\to\infty]{a.s.}\mu_\infty \tag{21}$$

We also have

$$\frac{\kappa_0\lambda_0}{\kappa_0+tn}\xrightarrow[t\to\infty]{a.s.}0 \tag{22}$$

which shows the first result.

Concerning the second result, we need to look at the variable $\overline{\sigma}_t^2$, that has its first two centered moments finite by assumption. Again using the strong law of large numbers, we have $\frac{(1-1/n)\sum_{j=0}^{t-1}\overline{\sigma}_j^2}{t}$ converges almost surely to a finite quantity that we denote Σ_∞. Hence

$$\frac{(n-1)\sum_{j=0}^{t-1}\overline{\sigma}_j^2}{v_0+(t-1)n-1}\xrightarrow[t\to\infty]{a.s.}\Sigma_\infty \tag{23}$$

We also have

$$\frac{\psi_0}{v_0+(t-1)n-1}\xrightarrow[t\to\infty]{a.s.}0 \tag{24}$$

To conclude we need to study the following term:

$$\begin{aligned}\frac{\kappa_t n}{\kappa_t+n}\left(\overline{x}_t-\mu_t\right)\left(\overline{x}_t-\mu_t\right)^T &= n\frac{\kappa_0+tn}{\kappa_0+tn+n}\left(\overline{x}_t-\mu_t\right)\left(\overline{x}_t-\mu_t\right)^T\\ &\xrightarrow[t\to\infty]{a.s.}0\end{aligned} \tag{25}$$

as $\overline{x}_t-\mu_t\xrightarrow[t\to\infty]{a.s.}0$, which concludes the proof. □

6.4 proof of proposition 14

Under proposition 1, we have

$$\begin{aligned}\mu_{t+1}=\lambda_{t+1}&=\lambda_t+\frac{n}{\kappa_t+n}((\overline{x}_t-\lambda_t)\\ &=\mu_t+\frac{n}{\kappa_t+n}\left(\overline{x}_t-\mu_t\right)\end{aligned} \tag{26}$$

Likewise, we have

$$\begin{aligned}\Sigma_{t+1}&=\frac{\psi_{t+1}}{v_{t+1}-n-1}\\ &=\frac{1}{v_t-1}(\psi_t+\frac{\kappa_t n}{\kappa_t+n}\left(\overline{x}_t-\lambda_t\right)\left(\overline{x}_t-\lambda_t\right)^T+(n-1)\overline{\sigma}_t^2)\\ &=\frac{1}{v_t-1}\Bigg((v_t-n-1)\Sigma_t+\frac{\kappa_t n}{\kappa_t+n}\left(\overline{x}_t-\mu_t\right)\left(\overline{x}_t-\mu_t\right)^T\\ &\qquad+(n-1)\overline{\sigma}_t^2\big)\end{aligned} \tag{27}$$

□

7 Other convergence comparisons for different starting points

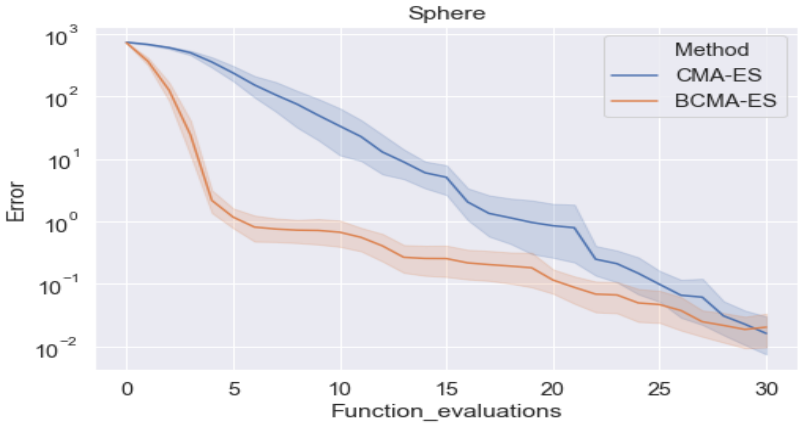

Fig. 5. Convergence for the Sphere function with a starting point at [-20,-20]

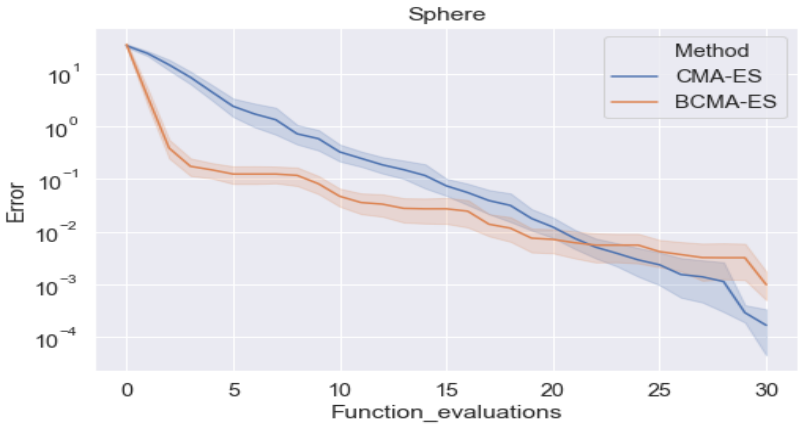

Fig. 6. Convergence for the Sphere function with a starting point at [-10,-10]

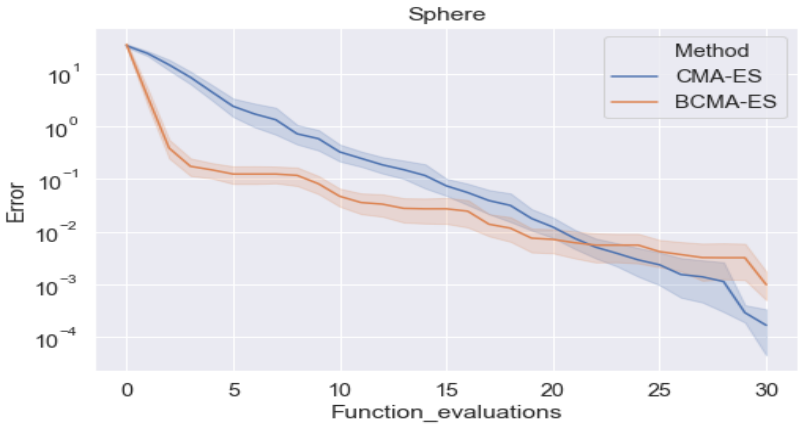

Fig. 7. Convergence for the Sphere function with a starting point at [-5,-5]

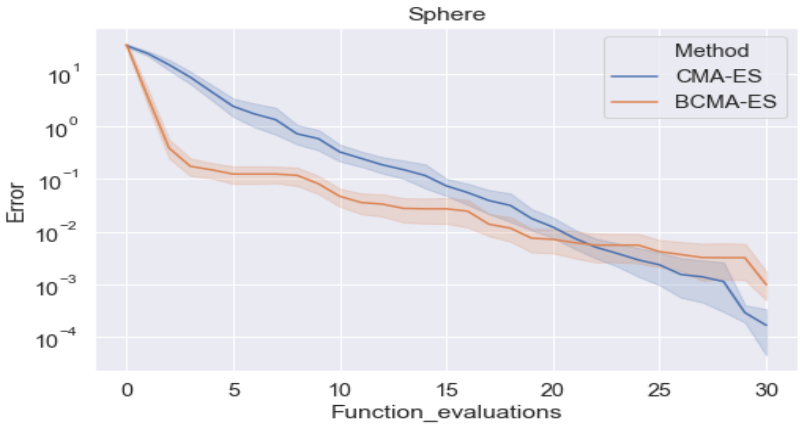

Fig. 8. Convergence for the Sphere function with a starting point at [5,5]

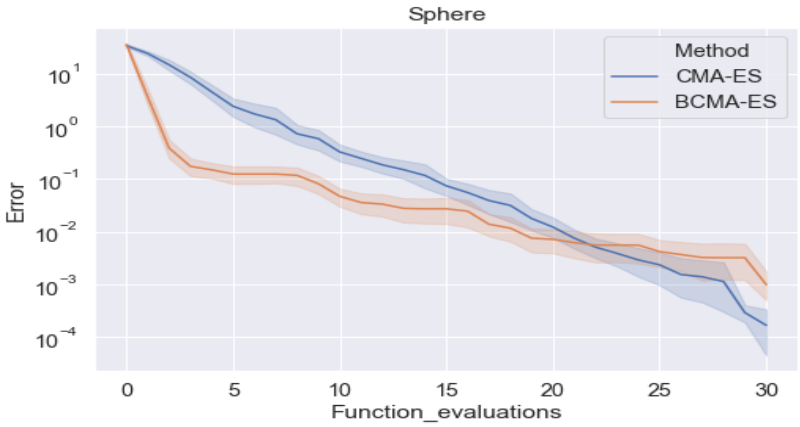

Fig. 9. Convergence for the Sphere function with a starting point at [10,10]

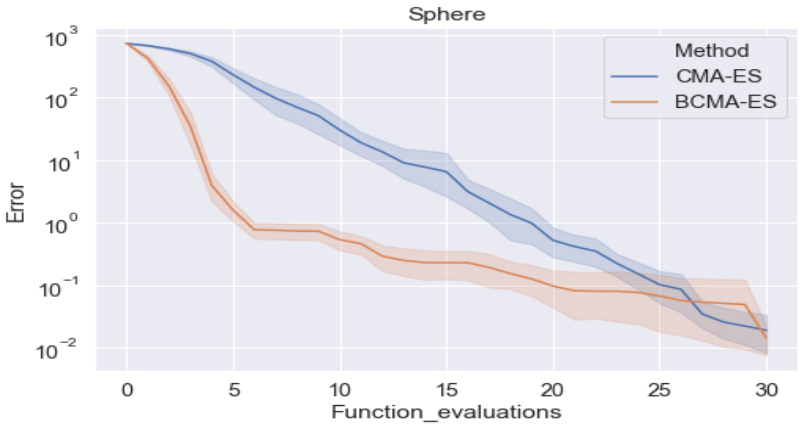

Fig. 10. Convergence for the Sphere function with a starting point at [20,20]

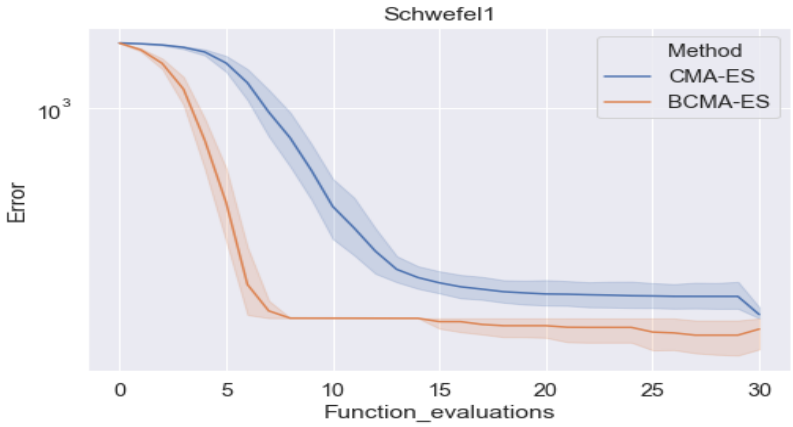

Fig. 11. Convergence for the schwefel1 function with a starting point at [-400,-400]

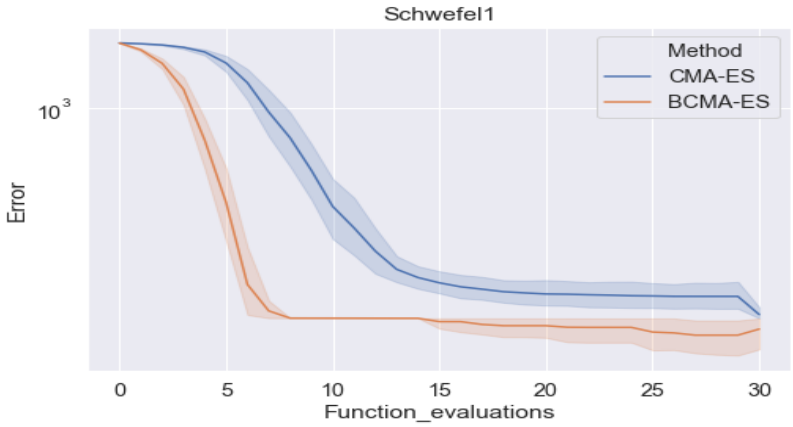

Fig. 12. Convergence for the schwefel1 function with a starting point at [-200,-200]

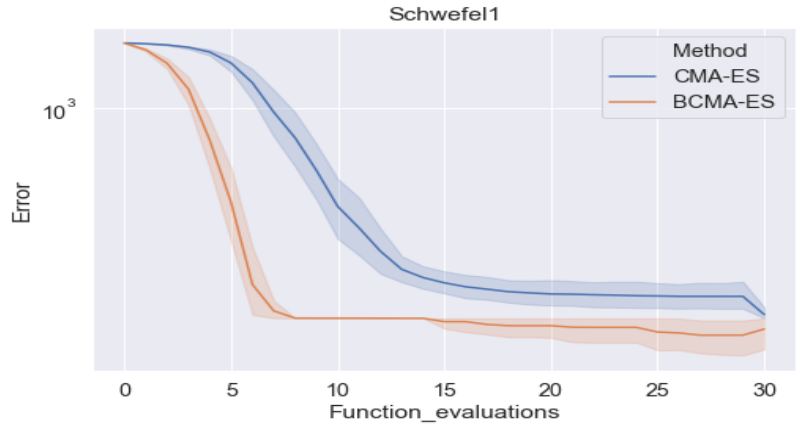

Fig. 13. Convergence for the schwefel1 function with a starting point at [-100,-100]

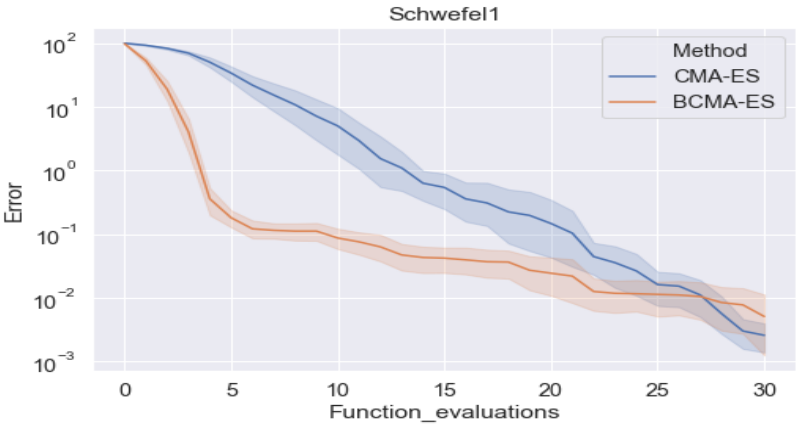

Fig. 14. Convergence for the schwefel1 function with a starting point at [100,100]

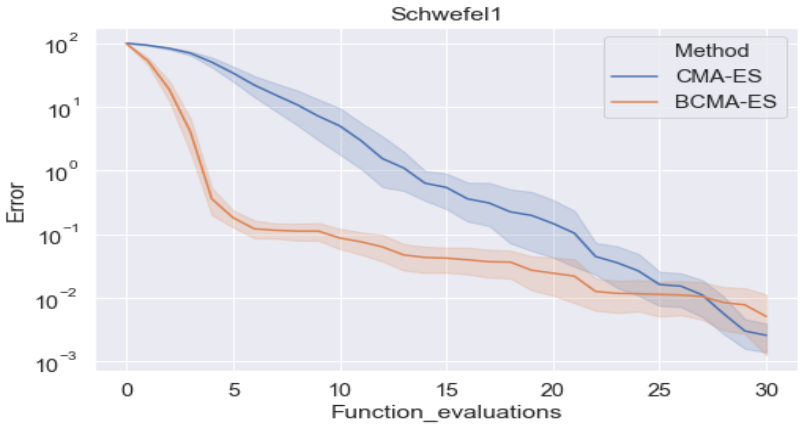

Fig. 15. Convergence for the schwefel1 function with a starting point at [200,200]

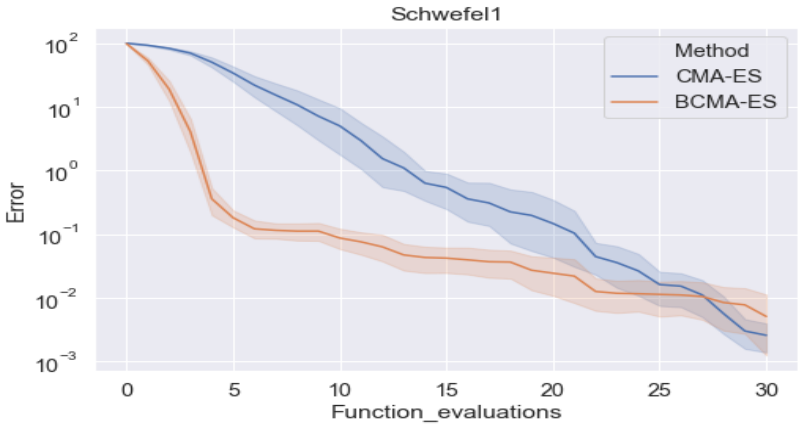

Fig. 16. Convergence for the schwefel1 function with a starting point at [400,400]

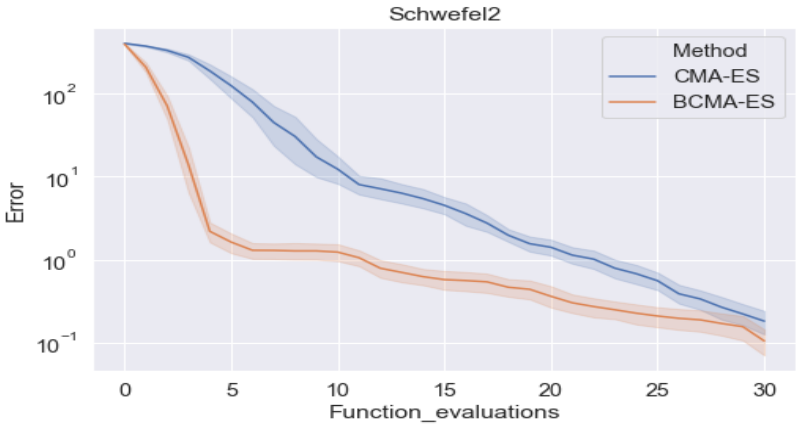

Fig. 17. Convergence for the schwefel2 function with a starting point at [-20,-20]

Fig. 18. Convergence for the schwefel2 function with a starting point at [-10,-10]

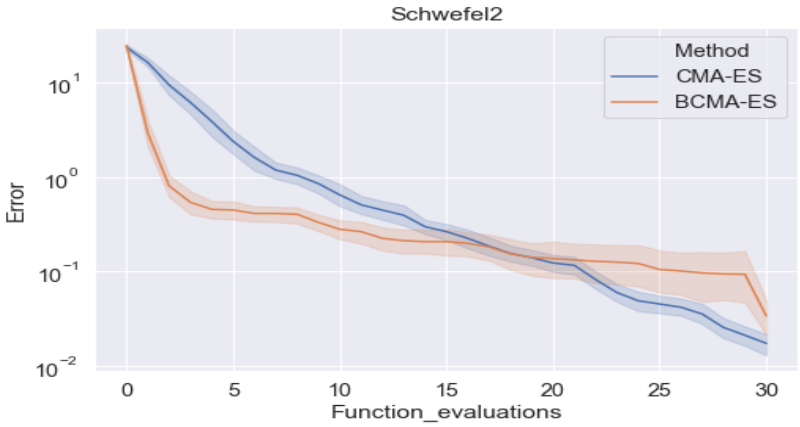

Fig. 19. Convergence for the schwefel2 function with a starting point at [-5,-5]

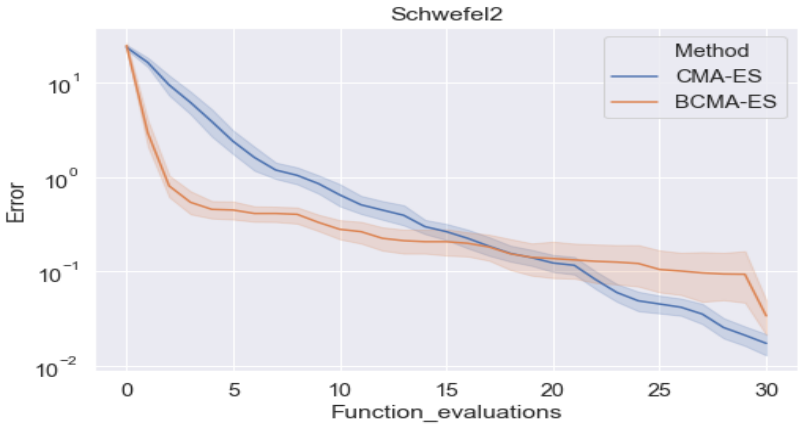

Fig. 20. Convergence for the schwefel2 function with a starting point at [5,5]

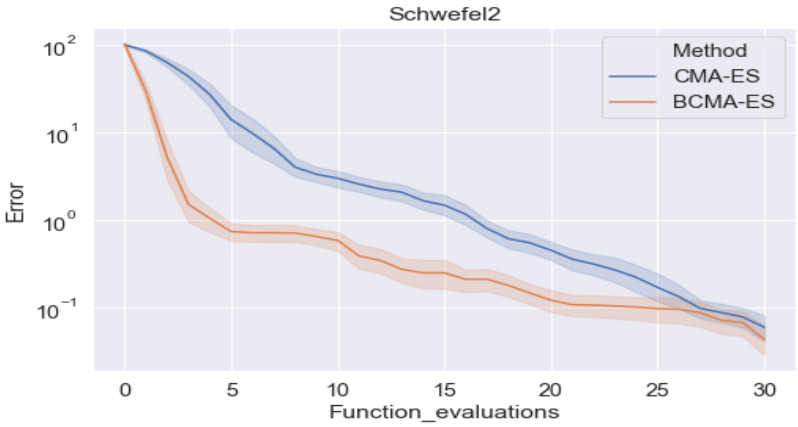

Fig. 21. Convergence for the schwefel2 function with a starting point at [10,10]

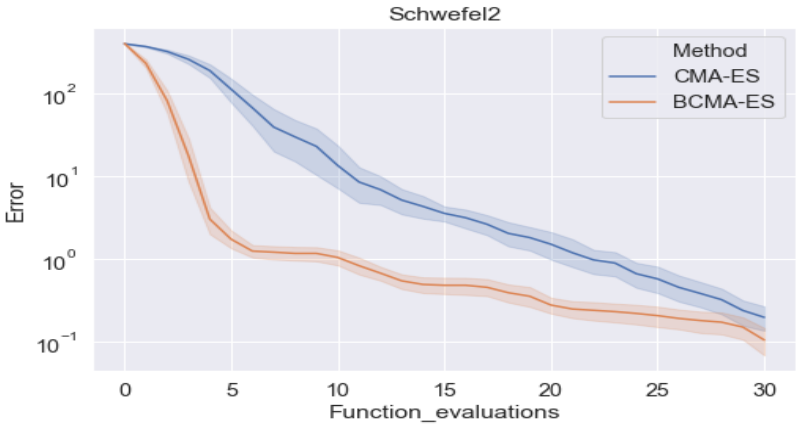

Fig. 22. Convergence for the schwefel2 function with a starting point at [20,20]

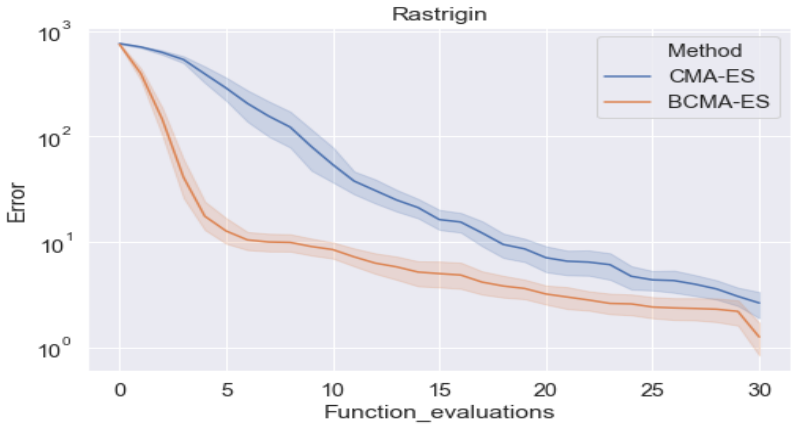

Fig. 23. Convergence for the rastrigin function with a starting point at [-20,-20]

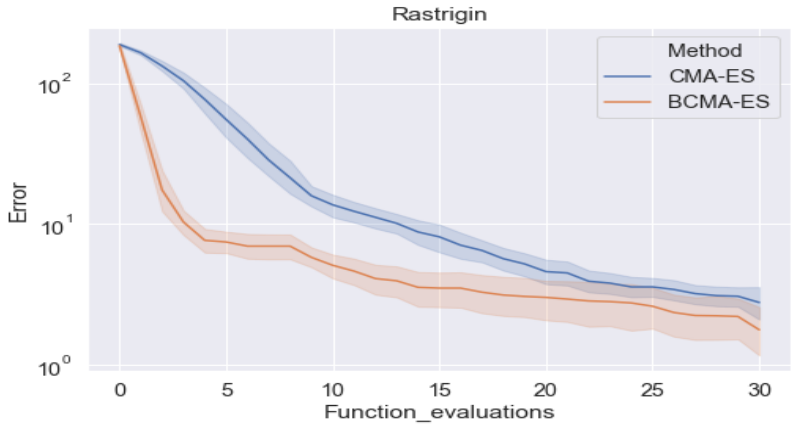

Fig. 24. Convergence for the rastrigin function with a starting point at [-10,-10]

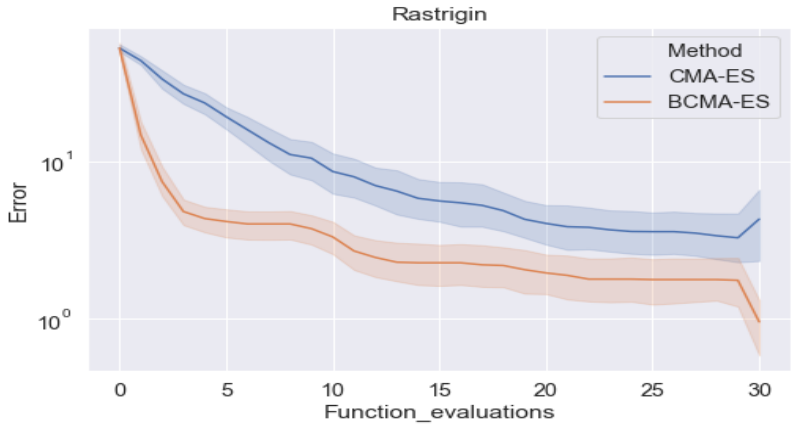

Fig. 25. Convergence for the rastrigin function with a starting point at [-5,-5]

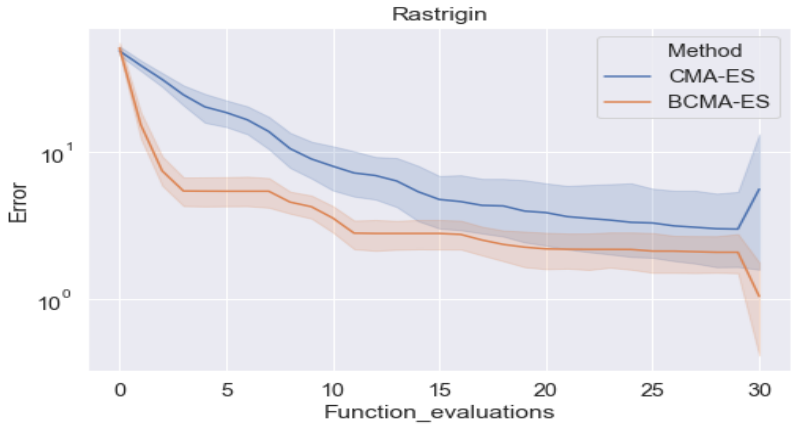

Fig. 26. Convergence for the rastrigin function with a starting point at [5,5]

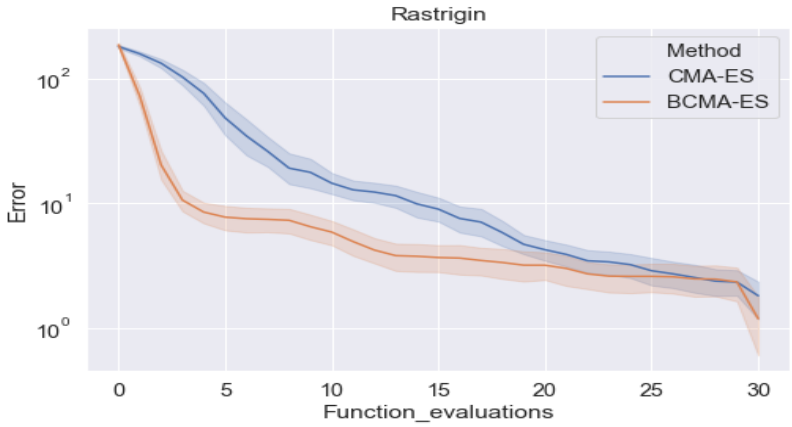

Fig. 27. Convergence for the rastrigin function with a starting point at [10,10]

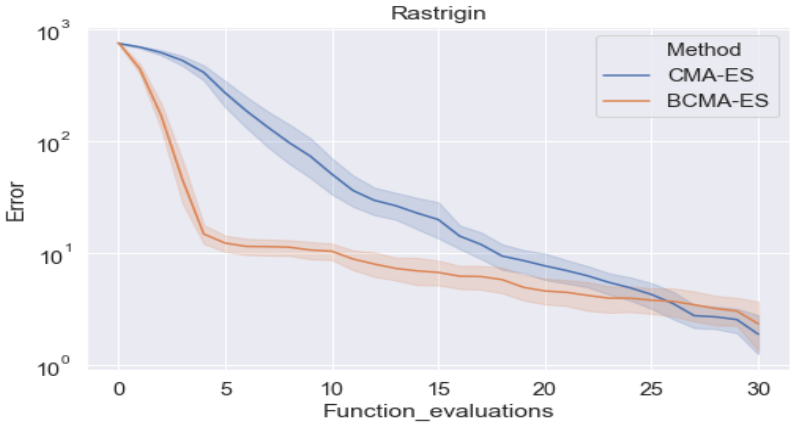

Fig. 28. Convergence for the rastrigin function with a starting point at [20,20]