

# Multivariate Cauchy EDA Optimisation

Momodou L. Sanyang<sup>1,2</sup> and Ata Kaban<sup>1</sup>

<sup>1</sup> School of Computer Science, University of Birmingham, Edgbaston, UK, B15 2TT  
{M.L.Sanyang, A.Kaban}@cs.bham.ac.uk

<sup>2</sup> School of Information Technology and Communication, University of the Gambia  
Brikama Campus, P.O. Box 3530, Serekunda, The Gambia  
MLSanyang@utg.edu.gm

**Abstract.** We consider Black-Box continuous optimization by Estimation of Distribution Algorithms (EDA). In continuous EDA, the multivariate Gaussian distribution is widely used as a search operator, and it has the well-known advantage of modelling the correlation structure of the search variables, which univariate EDA lacks. However, the Gaussian distribution as a search operator is prone to premature convergence when the population is far from the optimum. Recent work suggests that replacing the univariate Gaussian with a univariate Cauchy distribution in EDA holds promise in alleviating this problem because it is able to make larger jumps in the search space due to the Cauchy distribution's heavy tails. In this paper, we propose the use of a multivariate Cauchy distribution to blend together the advantages of multivariate modelling with the ability of escaping early convergence to efficiently explore the search space. Experiments on 16 benchmark functions demonstrate the superiority of multivariate Cauchy EDA against univariate Cauchy EDA, and its advantages against multivariate Gaussian EDA when the population lies far from the optimum.

**Keywords:** Multivariate Gaussian distribution, Multivariate Cauchy Distribution, Estimation of Distribution Algorithm, Black-box Optimization.

## 1 Introduction

Black-box global optimization is an important problem which has many applications in lots of disciplines. Optimization is at the core of many scientific and engineering problems. Mathematical optimization only deals with very specific problem types, while on the other hand the search heuristics like evolutionary computation work in a black box manner. They are not specialized on specific kinds of functions although they don't have the guarantees that the mathematical optimizations do. This paper presents a method which is classified as a search heuristic, and it is an extension of a recent version called Estimation of Distribution Algorithm (EDA).

EDA is a population based stochastic black-box optimization method that guides the search to the optimum by building and sampling explicit probability models of promising candidate solutions [2]. In EDA, the new population of individuals is generated without using neither crossover nor mutation operations, which is in contrast to

other evolutionary algorithms. In classical EDA, Gaussian distribution is used as the search operator to build a probabilistic model to fit the fittest individuals and create new individuals by sampling from the created model. It has been established that Gaussian EDA is prone to premature convergence [1], [2], [6] when its parameters are estimated using the maximum likelihood estimation (MLE) method. It converges too fast and does not get to the global optimum.

The premature convergence of classical Gaussian EDA attracted many efforts geared towards solving this problem [6], [10]. This paper presents the usage of Multivariate Cauchy distribution, an extension of [8] with a full matrix valued parameter that encodes dependencies between the search variable as an alternative search operator in EDA. We utilize its capability of making long jumps so as to escape premature convergence, which is typical of Gaussian in order to enable EDA algorithms get to the global optimum. Although Cauchy has already been used as an alternative search distribution for EDA [4], [5], [6], [10], it was the univariate version of Cauchy that was utilized, discarding statistical dependences among the search variables. We compare the performance of Multivariate Gaussian EDA with Multivariate Cauchy EDA to establish whether and when the long jumps that Cauchy is able make will be advantageous. We also compare the performance to Univariate Cauchy EDA to establish the advantages of multivariate modelling.

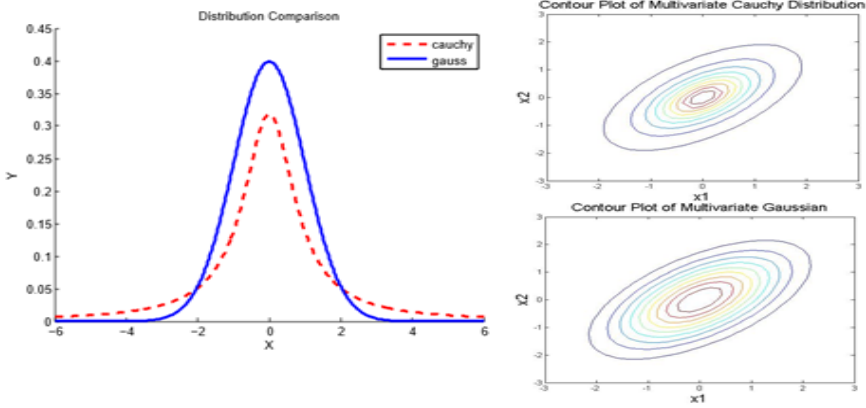
## 2 Differences between Multivariate Gaussian and Multivariate Cauchy Distributions

The main advantage of EDAs is the explicit learning of the dependences among variables of the problem to be tackled and utilizing this information efficiently to generate new individuals to drive the search to the global optimum [10]. Using univariate Cauchy will make it hard to achieve this goal since it does not take on dependences. Therefore, this paper to the best of our knowledge is the first to include the modelling of dependencies in a Cauchy search distribution based EDA algorithm for black-box global Optimization.

An important difference between Gaussian and Cauchy is that Cauchy is heavier-tailed. This means that it is more prone to producing values that fall far from its mean, thus, giving Cauchy more chance of sampling further down its tail than the Gaussian. This gives Cauchy a higher chance of escaping premature convergence than the Gaussian [6], [10].

For the same reason, the Gaussian search distribution is good when the individuals are close to the optimum while Cauchy is better when the individuals are far from the optimum. Both of these findings were previously made in the context of traditional evolutionary computation [8] where univariate version of these distributions were used to implement the mutation operator. Our hypothesis, which we test in this paper, is that these advantages are carried forward to EDA based optimization where in addition, multivariate modelling enables a more directed search.

Figure 1 shows the probability density functions for both Gaussian and Cauchy distributions in one and two dimensions.



**Fig. 1.** Cauchy density (red dashed), along with standard normal density (blue) (Left), Multivariate Cauchy (Top Right) and Multivariate Gaussian (Bottom Right)

The leftmost plot in figure 1 shows the probability density function of a Cauchy versus a Gaussian in 1D. We see the heavy tail of Cauchy falling down slower than Gaussian. On the right, the plots depict contour plots of the 2D versions of these densities, with Cauchy on the top right and Gaussian at the bottom right. In the 2D versions, Cauchy has a flatter tail on the base as can be seen by the wider space between the second outermost and the outermost contour lines than those of the Gaussian. The parameter matrix  $\Sigma = [1 \ .6; \ .6 \ 1]$  was used on both 2D version for plotting the contours.

### 3 Algorithm Presentation

The algorithms used in this paper for comparison are Multivariate Gaussian EDA (MGEDA), Multivariate Cauchy EDA (MCEDA) and Univariate Cauchy EDA (UCEDA). MGEDA takes on board the sample correlations between the variables of the selected individuals through a full covariance matrix, and MCEDA encodes pairwise dependencies among the search variables through its matrix valued parameter. UCEDA neglects dependencies among the search variables. Algorithm 1 describes generic EDA algorithm.

1. Set  $t := 0$ . Generate  $M$  points randomly to give an initial population  $P$ .
- Do**
2.     Evaluate fitness for all  $M$  points in  $P$ .
3.     Select some individuals  $P^{sel}$  from  $P$ .
4.     Estimate the statistics of  $P^{sel}$
5.     Use statistics in step (4) to sample new population  $P^{new}$ .
6.     Set  $P$  to  $P^{new}$
- Until Termination criteria are met.**

**Algorithm 1.** The Pseudocode of a simple EDA with Population size  $M$ .

This algorithm is a typical EDA, which proceeds by initially generating a population of individuals and then evaluates their fitness to select the fittest ones based on their fitness using the truncation selection. For the MGEDA, we compute the maximum likelihood estimates (MLE) of the mean ( $\mu$ ) and the covariance ( $\Sigma$ ) of the fittest individuals and use these parameters to generate new ones by sampling from a multivariate Gaussian distribution with parameters  $\mu$  and  $\Sigma$ . For MCEDA, we use the same estimates to sample from a Multivariate Cauchy distribution in step 5. In UCEDA, we use  $\mu$  and the diagonal elements of  $\Sigma$  to sample each from Univariate Cauchy. The new population is formed by replacing the old individuals by the new ones.

### 3.1 A Note on Parameter Estimation

The philosophy in EDA is to estimate the density of the selected individuals so that when new individuals are sampled from the model, they will follow the same distribution. Fortunately, for Gaussian this works. Parameter estimation in Cauchy distributions was studied in statistics [3] where an Expectation and Maximization (EM) algorithm was developed to find the maximum likelihood estimate of a multivariate Cauchy distribution from a set of points, which we implemented for our study.



**Fig. 2.** A plot showing the behavior of EDA when the search distribution is a Cauchy distribution

However, we found that when we estimate the Cauchy's parameter (Using EM), then the obtained model of the selected individuals (a Cauchy density) will disregard any outliers. This is of course what a robust density estimator is meant to do- however for optimization those outliers may be some rare and very good solutions that got close to an optimum. Fig 2 illustrates such an example.

As you can see in fig. 2, which was a snap shot taken from an iteration of the experiments we conducted, two selected individuals are close to the optimum and as such are good individuals but they are outliers with respect to the density of the rest of the selected individuals. This is the reason why in algorithm 1, the Multivariate Cauchy distribution was used only in the sampling step.

## 4 Experiments

Our hypothesis is that MCEDA has better performance than both UCEDA and MGEDA when the initial population is far from the optimum and also when the population size is small. In turn Multivariate Gaussian should perform better when the population is close to the optimum. To test this hypothesis, we conducted an extensive experiment on 16 benchmark functions taken from [7]. In the following subsections, we will describe the functions, parameter settings, then we present results with analysis and conclude.

### 4.1 Benchmark Test Functions

The comparisons of the three EDA algorithms were carried out on the suite of benchmark functions from the CEC'2005 competition. 16 test functions were used in this set of experiments. Among the functions tested, 5 are unimodal and 11 multimodal. All the global optima are within the given box constraints. However, problem 7 was without a search range and with the global optimum outside of the specified initialization range. All problems are minimization. Please see details of the functions in [7].

### 4.2 Parameter Setting

The dimensionality of all the problems is 2. We carried out three sets of experiments with the initial population size set to 20, 200 and 500 respectively. The percentage of individuals retained is 30%, which is a most widely used selection ratio. We did 25 independent runs for each problem on a fixed budget of 10,000 function evaluations in each case. The initialization was uniformly random within the search space. We also created harder versions of these problems by initializing far from the optimum to establish whether MCEDA can still perform in this situation.

### 4.3 Performance Criteria

The main performance criterion was the difference in fitness values (error) between the best individual found and the global optimum.

**Table 1.** Statistical Comparison of MCEDA and MGEDA on Problems 01-16 with initial Population far from the optimum and has size 200

	MCEDA		MGEDA		Rank sum Test	
	Mean	Std. dev.	Mean	Std. dev.	H	P
P01	<b>0</b>	<b>0</b>	1.6180e+03	508.7709	1	9.72e-011
P02	<b>0</b>	<b>0</b>	1.4116e+03	287.7206	1	9.72e-011
P03	<b>0</b>	<b>0</b>	3.8746e+08	1.2402e+08	1	9.72e-011
P04	<b>0</b>	<b>0</b>	953.7703	183.1595	1	9.72e-011
P05	<b>0</b>	<b>0</b>	7.0883e+03	411.0024	1	9.72e-011
P06	<b>0</b>	<b>0</b>	8.0103e+08	1.5906e+08	1	9.72e-011
P07	<b>0.0621</b>	<b>0.0464</b>	0.7204	0.7672	1	0.0010
P08	18.0903	3.9646	19.2004	4.0001	0	0.0625
P09	0.9025	0.6808	1.3412	0.9730	0	0.1057
P10	<b>0.6757</b>	<b>0.6134</b>	3.8126	2.4724	1	4.04e-007
P11	1.1429	0.4346	<b>0.0359</b>	<b>0.0993</b>	1	2.47e-010
P12	193.2284	184.7226	<b>0.4156</b>	<b>0.6739</b>	1	3.93e-009
P13	<b>0.0067</b>	<b>0.0147</b>	23.858	8.4094	1	6.57e-010
P14	<b>0.0220</b>	<b>0.0036</b>	0.7357	0.2264	1	1.41e-009
P15	<b>0.0649</b>	<b>0.0402</b>	52.3710	7.9035	1	1.41e-009
P16	<b>0</b>	<b>0</b>	1.7225	4.6068	1	0.0412

**Table 2.** Statistical Comparison of MCEDA and MGEDA on Problems 01-16 with uniform initialisation and small Population size 20

	MCEDA		MGEDA		Rank sum Test	
	Mean	Std. dev.	Mean	Std. dev.	H	P
P01	<b>0</b>	<b>0</b>	35.1160	163.9960	1	4.4787e-009
P02	<b>0</b>	<b>0</b>	116.3862	213.6704	1	1.3101e-009
P03	<b>0</b>	<b>0</b>	7.3128e+04	3.5850e+05	1	3.6574e-010
P04	<b>0</b>	<b>0</b>	132.4800	311.6559	1	1.3101e-009
P05	<b>0</b>	<b>0</b>	1.4107e+03	1.2712e+03	1	4.4787e-009
P06	<b>0</b>	<b>0</b>	14.0273	20.0025	1	3.6574e-010
P07	0.0858	0.1346	0.4516	1.1137	0	0.2288
P08	19.7487	4.1438	<b>17.6353</b>	<b>6.5513</b>	1	2.4248e-008
P09	3.5818	4.0211	0.6134	0.5887	0	0.2111
P10	<b>0.7562</b>	<b>0.9646</b>	1.3387	1.9365	1	0.0092
P11	0.5856	1.1481	<b>0.2505</b>	<b>0.3295</b>	1	0.0195
P12	2.0355e+03	1.1826e+03	<b>10.2325</b>	<b>20.2760</b>	1	9.2160e-010
P13	0.1538	0.2352	0.0507	0.0531	0	0.1633
P14	0.6806	0.2313	<b>0.1408</b>	<b>0.0194</b>	1	1.4634e-007
P15	0.1183	0.1278	1.1885	2.9844	0	0.8613
P16	<b>0.6865</b>	<b>2.9408</b>	2.2038	3.8294	1	1.1752e-007

**Table 3.** Statistical Comparison of UCEDA and MCEDA on Problems 01-16 with uniform initialisation and small Population size 20

	UCEDA		MCEDA		Rank sum Test	
	Mean	Std. dev.	Mean	Std. dev.	H	P
P01	0	0	0	0	0	N/A
P02	0	0	0	0	0	N/A
P03	5.2295e+03	6.0844e+03	<b>0</b>	<b>0</b>	1	9.7282e-011
P04	0	0	0	0	0	N/A
P05	0	0	0	0	0	N/A
P06	5.6064	14.1692	<b>0</b>	<b>0</b>	1	9.7282e-011
P07	0.2708	0.2092	<b>0.0858</b>	<b>0.1346</b>	1	1.4256e-004
P08	20.5476	0.5120	19.7487	4.1438	0	0.9690
P09	<b>3.5818</b>	<b>4.0211</b>	6.1954	4.4285	1	0.0289
P10	0.9154	0.9916	0.7562	0.9646	0	0.3855
P11	0.1991	0.5776	0.5856	1.1481	0	0.3731
P12	<b>1.0293e+03</b>	722.7733	2.0355e+03	1.1826e+03	1	0.0108
P13	0.3313	0.3508	<b>0.1538</b>	<b>0.2352</b>	1	1.7045e-004
P14	0.8139	0.0550	<b>0.6806</b>	<b>0.2313</b>	1	0.0085
P15	0.3948	0.1526	<b>0.1183</b>	<b>0.1278</b>	1	5.0089e-007
P16	0.4866	2.4328	0.6865	2.9408	0	0.5717

## 5 Results and Discussion

The results of our experiments are summarized in tables 1 to 3 and bold font indicates statistically significant out performance at 95% confidence level.

Tables 1 and 2 report results from experiments that compare MCEDA with MGEDA. MCEDA performed better than MGEDA in most of the 16 benchmark functions, see tables 1 and 2. From results omitted for space constraints, we also found when the population size was increased to 200, MGEDA outperformed the MCEDA, and this was also confirmed in the results of the experiments with population size 500. The reason for this is the MGEDA is better when the best individuals are close to the optimum, so when we initialized lots of them everywhere, there are chances that some of them will be close. To test this hypothesis, we devised two sets of experiments with initial population far from the optimum (Results shown in Table 1), and with uniform initialization everywhere in the search space, but a smaller population size of only 20 (Results shown in Table 2). Taken together, these results confirm our hypothesis discussed above. MCEDA has performed better than MGEDA, in both of these settings. This is because of the long jumps of Cauchy.

In table 3, we report a comparison between MCEDA and UCEDA. We can clearly see from table 3 that MCEDA performed better than UCEDA in most of the functions.

## 6 Conclusions and Future Work

In this paper, we studied the use of a multivariate Cauchy distribution in black-box continuous optimization by EDA. Our MCEDA blends together the advantages of multivariate modelling with the Cauchy sampling's ability of escaping early convergence and efficiently explore the search space. We conducted extensive experiments on 16 benchmark functions and found that MCEDA outperformed MGEDA when the population is far from the global optimum and is able to work even with small population sizes. We also demonstrated the superiority of multivariate Cauchy EDA against univariate Cauchy EDA.

Future work is to extend this study to high dimensional search spaces, possibly leveraging recent techniques of random projection for optimization [1].

## References

1. Kaban, A., Bootkrajang, J., Durrant, R.: Towards large scale continuous EDA: A random matrix theory perspective. In: Proc. of Genetic and Evolutionary Computation Conference (GECCO), Amsterdam, The Netherlands, pp. 383–390 (2013)
2. Larranaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A new tool for Evolutionary Computation. Kluwer Academic Publishers (2001)
3. Peel, D., McLachlan, G.: Robust mixture modelling using the t distribution. *Statist. Comput.* 10, 339–348 (2000)
4. Posik, P.: BBOB-Benchmarking a simple Estimation of Distribution Algorithm with Cauchy Distribution. In: GECCO 2009, Montreal, Quebec, Canada (July 2009)
5. Posik, P.: Comparison of Cauchy EDA and BIPOP-CMA-ES Algorithms on the BBOB Noiseless Testbed. In: GECCO 2010, Portland, Oregon, USA (July 2010)
6. Pošík, P.: Preventing premature convergence in a simple EDA via global step size setting. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 549–558. Springer, Heidelberg (2008)
7. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimisation (2005)
8. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computations* 3(2) (July 1999)
9. Yao, X., Liu, Y.: Fast evolution strategies. *Control and Cybernetics* 26(3), 467–496 (1997)
10. Yuan, B., Gallagher, M.: On the importance of diversity maintenance in estimation of distribution algorithms. In: Beyer, H.G., O'Reilly, U.M. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005, vol. 1, pp. 719–729. ACM Press, New York (2005)