How effective is Cauchy-EDA in high dimensions?

Momodou L. Sanyang
School of Computer Science
University of Birmingham
Edgbaston
Birmingham, B15 2TT, UK
Email: M.L.Sanyang@cs.bham.ac.uk

Robert J. Durrant
Department of Statistics
University of Waikato
Private Bag 3105
Hamilton 3240, NZ
Email: BobD@waikato.ac.nz

Ata Kabán
School of Computer Science
University of Birmingham
Edgbaston
Birmingham, B15 2TT, UK
Email: A.Kaban@cs.bham.ac.uk

***Abstract*—We consider the problem of high dimensional black-box optimisation via Estimation of Distribution Algorithms (EDA) and the use of heavy-tailed search distributions in this setting. Some authors have suggested that employing a heavy tailed search distribution, such as a Cauchy, may make EDA better explore a high dimensional search space. However, other authors have found Cauchy search distributions are less effective than Gaussian search distributions in high dimensional problems. In this paper, we set out to resolve this controversy. To achieve this we run extensive experiments on a battery of high-dimensional test functions, and develop some theory which shows that small search steps are always more likely to move the search distribution towards the global optimum than large ones and, in particular, large search steps in high-dimensional spaces nearly always do badly in this respect. We hypothesise that, since exploration by large steps is mostly counterproductive in high dimensions, and since the fraction of good directions decays exponentially fast with increasing dimension, instead one should focus mainly on finding the right direction in which to move the search distribution. We propose a minor change to standard Gaussian EDA which implicitly achieves this aim, and our experiments on a sequence of test functions confirm the good performance of our new approach.**

I. Introduction

Estimation of Distribution Algorithms (EDA) represent a branch of stochastic optimization heuristics that, in contrast to classical Evolutionary Algorithms, build and sample probability models of the good individuals in each generation [10]. By model building, EDA tries to learn the structure of the search space in order to guide the search towards promising areas [1]. A comprehensive overview of EDA techniques and applications may be found in [7].

EDA is known to have good properties as long as the search space is low dimensional, but it is notoriously bad in high dimensions due to excessive computational resource requirements [3], [9], [11]. In an attempt to remedy this, several authors have proposed employing heavy-tailed distributions in the sampling step of EDA instead of the more commonly used Gaussian. For instance, [20] proposes a univariate continuous EDA (UMDAc) with Lévy sampling. Furthermore, in later work by [18], Cauchy sampling has been reported to be superior to Gaussian in high dimensions. Cauchy is a very heavy tailed distribution that has no finite mean. From the conclusions of these works it appears as though the ability to make long jumps should be beneficial for high dimensional search. Though, we should note that, the study in [18], although termed 'high dimensional' by the authors, it only considered problems of up-to 32 dimensions.

On the other hand, other work [6] has found that Cauchy's long jumps virtually never lead to better solutions in high dimensional search spaces. In fact, the list of negative findings about Cauchy-based search in high dimension does not end here: In [6], the authors analyzed the volume of the level sets of the Cauchy vs. Gaussian densities, for both isotropic and anisotropic Cauchy distributions, with respect to their effectiveness when utilized in searching for optima in multimodal objective functions in an (1+1) EA. Moderate dimensions were considered, up to 20, but the results have led the authors to conclude with the conjecture that, for global optimization, heavy tails are only useful if the large variations take place mainly in a low dimensional subspace and the low dimensional space contains the better optima. Also, [14] compared BIPOP-CMA-ES having a Gaussian probabilistic model, against Cauchy EDA, and concluded that BIPOP-CMA-ES dominates the Cauchy EDA performance regardless of the particular optimization conditions. The maximum dimension considered in this study was 40. Furthermore, [15] compared Cauchy EDA against G3PCX algorithms that use Gaussian on the BBOB noiseless testbed (up-to 40 dimensions), and reported that G3PCX won in 6 out of 10 cases tested.

Low dimensional studies in turn (up-to 3 dimensions) are pretty consistent to find Cauchy superior to Gaussian when the population is relatively far from the optimum – see for instance [5], [13], [16], [22]. But in high dimensions we see a controversy in the existing literature. One issue is that the mentioned previous comparisons were done with different algorithms so it is hard to distill a global picture. Secondly, evidence about the merits of Cauchy vs. Gaussian based search is largely missing in the literature on problems larger that 40 dimensions. What will happen on problems with 50-1000 dimensions?

In this paper we set out to resolve the above controversy, and we conduct a thorough investigation into the performance of multivariate Cauchy EDA in high dimensions up to 1000 dimensional problems in comparison with its Gaussian counterpart. We shall use a scalable variant of EDA called EDA with Model Complexity Control (EDA-MCC) [3] for our purpose, and create a Cauchy sampling variant of it.

II. Presentation of the algorithm used in this work

We chose EDA-MCC [3] as the algorithmic tool for our experiments, because it is scalable and applicable to both low and high dimensional problems, and it was previously demonstrated to work well up to 500 dimensions. This allows us to vary the problem size and observe the trends in performance comparatively for Gaussian and Cauchy search distributions. Among alternatives that could be used, the random projection ensemble based EDAs [9], [17] were specifically designed for high dimensional problems. Since testing in low dimensional regimes (e.g. 2 to 20) would defeat the purpose of the random projection technique, this would limit our experiments.

Algorithm 1 The Pseudocode of a generic EDA

(1) Set $t \leftarrow 0$.
(2) Set $P \leftarrow$ Generate N points randomly to give an initial population.
Do
(3) Evaluate fitness for all N points in P
(4) Select the best individuals P^{sel} from P
(5) Calculate the sample statistics $\hat{\theta}$ of P^{sel}
(6) Sample new population P^{new} from the distribution with parameters $\hat{\theta}$
(7) $P \leftarrow P^{new}$
Until Termination criteria are met

In its original form, EDA-MCC employs a multivariate Gaussian search distribution, which for scalability purposes is modeled / approximated as a product distribution on non-overlapping subspaces. These are created by randomly partitioning the search variables that have correlations into disjoint groups. The variables that only have correlations smaller than the threshold in absolute value are modeled as univariate product distributions.

Before proceeding further, we should mention that correlation only captures linear dependencies and will miss any nonlinear ones. In a separate study we experimented with employing Mutual Information estimates instead, but observed only marginal improvements at a considerably higher computation cost [21], most likely because very accurate estimates of the dependency structure are not so crucial in a heuristic search that aims for finding approximate solutions.

It is straightforward to modify this strategy to sample from independent multivariate Cauchy blocks instead, which we do for the purpose of our experiments. The pseudo-code of a generic EDA is given in Algorithm 1, and Algorithm 2-3a-3b summarize EDA-MCC. Our only modification is in the multivariate modeling, namely step (c) of Algorithm 3a, to allow for multivariate Cauchy sampling in the subspaces. We implemented the multivariate Cauchy sampling by making use of the Gaussian scale-mixture representation of the Cauchy density [12], and sampling this generatively:

$$\text{Cauchy}_x(\mu, \Sigma) = \int_{u>0} N_x(\mu, \Sigma/u)\, \text{Ga}_u(1/2, 1/2)\, du \quad (1)$$

Algorithm 2 EDA-MCC

Inputs: θ, c, mc, $sampling$
(1) Set $t \leftarrow 0$.
(2) Set $P \leftarrow$ Generate N points uniformly randomly in the search box to give an initial population.
Do
(3) Evaluate the fitness of all N points in P
(4) $P^{sel} \leftarrow$ Select the fittest $m < N$ individuals from P using truncation selection.
(5) Split the search variables in 2 groups:
(a) Estimate the $d \times d$ correlation matrix C from a random subset of size $mc \leq m$ of P^{sel}.
(b) Split $\{1, ..., d\} = T_u \bigcup T_s$ as follows:
$T_u \leftarrow \{i : \forall j \neq i, C(i,j) < \theta\}$
$T_s \leftarrow \{1, ..., d\} - T_u$
(6) $W_u \leftarrow P^{sel}_{|T_u}$ //P^{sel} restricted to variables in T_u
$W_s \leftarrow P^{sel}_{|T_s}$ //P^{sel} restricted to variables in T_s
$P^{new}_{|T_s} \leftarrow$ call SM($W_s, c, sampling$)
$P^{new}_{|T_u} \leftarrow$ call WI(W_u)
(7) $P \leftarrow P^{new}$
Until Termination criteria are met
Output: P

Algorithm 3a Subspace Modeling of strongly correlated variables

function SM
Inputs: W_s, c, smp
$L \leftarrow$ dimensionality of W_s
Randomly partition the L variables of W_s into L/c non-intersecting subsets, W_{s_1},...,$W_{s_{L/c}}$
for i = 1 to L/c
(a) $\mu_i \leftarrow$ sample mean from W_{s_i}
(b) $\mathbf{\Sigma}_i \leftarrow$ sample covariance ($c \times c$) from W_{s_i}
(c) If $smp =$ 'Gaussian', $\mathbf{s}_1^{(i)}, ..., \mathbf{s}_N^{(i)} \overset{iid}{\sim} N(\mu_i, \mathbf{\Sigma}_i)$
Else $smp =$ 'Cauchy' with μ_i as location parameter & $\mathbf{\Sigma}_i$ as dispersion parameter:
$\mathbf{s}_1^{(i)}, ..., \mathbf{s}_N^{(i)} \overset{iid}{\sim}$ Cauchy$(\mu_i, \mathbf{\Sigma}_i)$
(d) $S_{|(i-1)\cdot c+1:i\cdot c} \leftarrow [\mathbf{s}_1^{(i)}, ..., \mathbf{s}_N^{(i)}]$
endfor
Output: S
end function

where u may be regarded as an hidden variable, and $\text{Ga}(\cdot)$ is the Gamma density.

III. Experiments

We set out to resolve the controversy about the comparative merits of multivariate Gaussian vs. Cauchy search distributions in high dimensions. Towards this end, we conducted experiments on 7 benchmark functions taken from the CEC05 competition [19] – these are listed in Table I – and we varied the problem dimensionality from 20 up to 1000. Among the functions tested, 4 are unimodal, and 3 multi-modal. All

Algorithm 3b Univariate Modeling of weakly correlated variables

function WI
Inputs: W_u
$L \leftarrow$ dimensionality of W_u
for i = 1 to L
(a) Estimate $\mu_i \leftarrow$ sample_mean($W_{u|i}$)
(b) Estimate $\sigma_i^2 =$ sample_variance($W_{u|i}$)
(c) Draw $u_1^i, ..., u_N^i \overset{iid}{\sim} N(\mu_i, \sigma_i^2)$
(d) $U_{|i} \leftarrow (u_1^i, ..., u_N^i)$
endfor
Output: U
end function

TABLE I: Scalable test functions from the CEC'05 collection.

Problem	Name
P01	Shifted Sphere Function
P02	Shifted Schwefel's Problem 1.2
P03	Shifted Rotated High Conditioned Elliptic Function
P04	Shifted Schwefel's Problem 1.2 with Noise in Fitness
P05	Shifted Rosenbrock's Function
P06	Shifted Rastrigin's Function
P07	Expanded Extented Griewank Function plus Rosenbrock

the global optima are within some given box constrains. All problems are minimization. More details on the functions may be found in [19].

A. Roadmap and parameter settings

Our first experiments were conducted on the Shifted Rosenbrock Function to replicate the findings of [18] in the settings considered there (i.e varying dimensions up to 32). The purpose of this experiment was to see if the version of EDA we are using is consistent with their findings. Once confirmed, we further looked at the Shifted Rosenbrock Function in higher dimensions to get a more complete picture. As we shall see, the conclusion turns out to be very different in the higher dimensional regime.

We then conducted experiments on a good number of benchmark problems to test if the above finding is observed more generally. The following set of dimensions (problem sizes) were used to conduct our experiments, {20, 30, 40, 50, 100, 200, 300, 400, 500, 1000} for all problems.

All experiments were ran with three different population sizes {300, 1000, 2000} in order to make sure that the observed behavior is not a byproduct of a particular choice of population size. A budget of $10000 \times d$ function evaluations was set in all experiments, where d is the dimension of the problem. This was the recommended budget size in [19] for the CEC'05 competition.

The following tunable parameters were set in accordance with the recommendations in [3]: The threshold θ to decide if a search variable has weak or strong correlations is set to 0.3, the number of selected individuals (m) is set to half of the population size, and the sample size used to estimate correlations (mc) is set to 100. However, we did not go by the recommendation of [3] in setting the maximum group size, c. The reason will be explained shortly. Instead, we set $c = \min(\lceil d/5 \rceil), \lceil N/15 \rceil)$, where N is the population size. The performance criterion is the difference (gap) between the fitness of the best individual found and the true global optimum. Each experiment was run 25 times (with random independent restarts) and we report the average and standard deviation of these differences.

1) A note on setting the max group size, c in EDA-MCC: We believe the following must be a typo on page 811 in [3], for their 500-dimensional experiments, where the block size is claimed to be set to $c = 100$ and the number of selected individuals is $m = 100$. In our experience this setting does not work, and indeed this setting would mean to estimate 100×100 covariance blocks from only 100 points which leads to a singular covariance estimate (its rank is at most 99 due to the degree of freedom lost by estimating the mean). Thus one needs to either reduce the block size c or to increase the population size N: Since the latter is undesirable we took $c = \min(\lceil d/5 \rceil, \lceil N/15 \rceil$. With our setting, now we have $c \times c = \min(\lceil d/5 \rceil, \lceil N/15 \rceil) \times \min(\lceil d/5 \rceil, \lceil N/15 \rceil)$ covariance blocks to estimate from $m = \lceil N/2 \rceil$ points.

IV. Results and Discussion

A. Results on shifted Rosenbrock: Confirming the findings of [18], and developing a more complete picture

Following [18], we start by running experiments on the shifted Rosenbrock function up to 32 dimensions. As we already mentioned, [18] reported superior performance when employing the Cauchy search distribution as opposed to the Gaussian when tested in this dimensionality range. Although they use a different optimization algorithm and different parameter setting than ours, we were able to confirm their finding. Table II presents our results obtained with the population size $N = 2000$, along with a statistical analysis. We see that the Cauchy search distribution performs significantly better than the Gaussian up to 100 dimensions in this case.

TABLE II: Ranksum Statistical test for performance comparison between Gaussian and Cauchy on Shifted Rosenbrock function with Budget = $10000 \times d$ and Population size = 2000.

Dimension	Cauchy		Gaussian		Ranksum Test	
	mean	std	mean	std	H	P-Value
20	**20.3619**	**21.5225**	23.8297	31.5721	1	3.12E-24
30	**1.15E+04**	**4.95E+04**	5.30E+04	9.35E+04	1	2.13E-21
40	**8.03E+03**	**6.32E+04**	5.18E+04	5.40E+04	1	3.45E-30
50	**338.7745**	**976.4567**	5.66E+04	7.39E+04	1	1.54E-33
100	**8.04E+03**	**4.76E+04**	1.55E+05	1.65E+05	1	3.56E-32
200	5.44E+10	5.85E+09	**2.31E+05**	**1.93E+05**	1	2.56E-34
300	4.53E+11	2.01E+10	**3.88E+05**	**2.79E+05**	1	2.56E-34
400	9.18E+11	4.34E+10	**7.72E+05**	**4.38E+05**	1	2.56E-34
500	1.35E+12	4.89E+10	**1.14E+06**	**6.06E+05**	1	2.56E-34
1000	3.77E+12	1.29E+11	**5.03E+06**	**1.66E+06**	1	7.07E-18

However, we also see from Table II that the extrapolation suggested in [18] to higher dimensional problems than those tested by the authors, actually fails. Instead, we see a crossing point at around $d = 100$, after which exactly the opposite

conclusion becomes true: The Gaussian search distribution performs significantly better than the Cauchy at problem dimensions larger than $d = 100$, up to $d = 1000$.

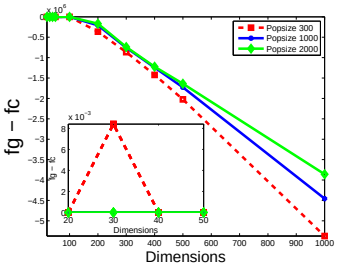
We found the above conclusion consistently (up to slight shifts of the crossing point) when choosing other population sizes as well. This will be apparent in the next subsection where summary plots of results obtained with three different population sizes will be presented. Moreover, as we shall see, the finding that Gaussian performs better than Cauchy in high (beyond 100) dimensional problems is also observed for all benchmark problems tested.

B. Results of an extensive empirical study

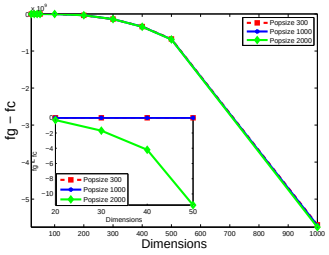
Having found an interesting pattern of comparative behavior in the previous section on the shifted Rosenbrock function, we then performed similar comparative experiments on all functions from Table I in order to see if our finding holds more generally. Figure 1 presents all these results in a compact format. Here we display the differences between the fitness value achieved with Gaussian (fg) and with Cauchy (fc) search distributions respectively. By fitness value we mean the average of the best fitness in the last generation, as averaged over 25 independent runs. Whenever this difference ($fg - fc$), is positive it means that Cauchy outperformed Gaussian (recall, we do minimization so smaller fitness is better), and vice-versa – whenever $fg - fc$ is negative then Gaussian outperformed Cauchy. The 7 plots correspond to the 7 benchmark problems tested, and each curve on these plots corresponds to a particular choice of population size. Since the fitness differences are much larger when d is large, we also show a zoomed version of the lower dimensional regime in order to better see the details.

From Figure 1 we see that the comparative behavior of the two search distributions in the high dimensional regime, as observed in the previous section, consistently holds up on all functions tested, and with all population sizes tested. That is, the differences in the fitness values ($fg - fc$) are positive in the dimension range 20-50 in most cases, meaning that Cauchy tends to be better in this regime. But, as the dimension exceeds 50 or 100, the differences become negative and remain negative, indicating that Gaussian is now better than Cauchy. We can also see from figure 1 that the results with smaller population size yield the largest contrast between the performances of these two search distributions.

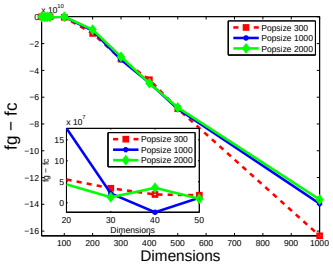
We therefore conclude on the basis of these results that Cauchy may be better than Gaussian in low dimensional problems, but Gaussian is superior in high dimensional problems. Statistical tests (omitted for space constraints) confirmed that these differences are statistically significant.



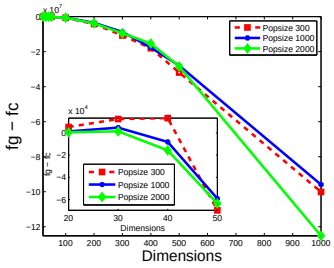
(a) Shifted Sphere Function



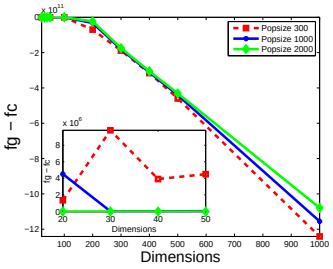
(b) Shifted Schwefel's Problem 1.2



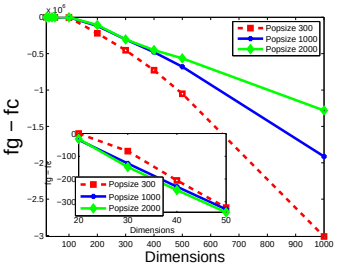
(c) Shifted rotated Elliptic Function



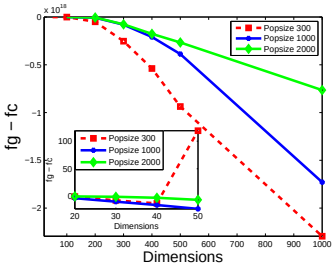
(d) Shifted Schwefel's Problem 1.2 With Noise in fitness



(e) Shifted Rosenbrock Function



(f) Shifted Rastrigin Function



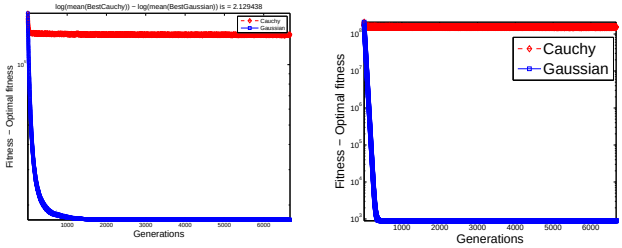
(g) Exp. Ext. Griewank Fn + Rosenbrock

Fig. 1: Differences between the average (from 25 repeated runs) of the best fitness values achieved by the Cauchy (fc) and by the Gaussian (fg) EDAs, as the dimension is varied, for seven test problems. The smaller plots superimposed represent zoomed versions of the same results in the range of 20-50 dimensions.

C. Further results when the optimum is shifted much further away

Since Cauchy sampling in optimisation is expected to have an advantage over Gaussian when long jumps are beneficial, we also tried to modify the test problems by shifting the global optimum and increasing the search box sizes from $[-10^2\ \ 10^2]$ up to $[-10^7\ \ 10^7]$, to see if Cauchy's long jumps will pay off. We found this is not the case, and Cauchy search makes very slow progress in all cases tested. Example results are given in Figure 2. These experiments conclude that Cauchy long jumps does not help in high dimensions, which agrees with

the findings in [6]. That is, the chances for a long jump to turn out lucky vanish with increasing dimension, and in the next section we show that in fact this issue is unavoidable.



(a) Rosenbrock; Shift = 10000 (b) Sphere; Shift = 300000

Fig. 2: Comparisons of Gaussian vs. Cauchy search distributions on problems with highly shifted optima and increased sizes of the search box.

V. Understanding the reasons for our experimental findings

Here we show why large search steps are, in general, more likely to perform worse than smaller ones and explain the role that the problem dimensionality plays in this issue.

We start by considering a search distribution that selects

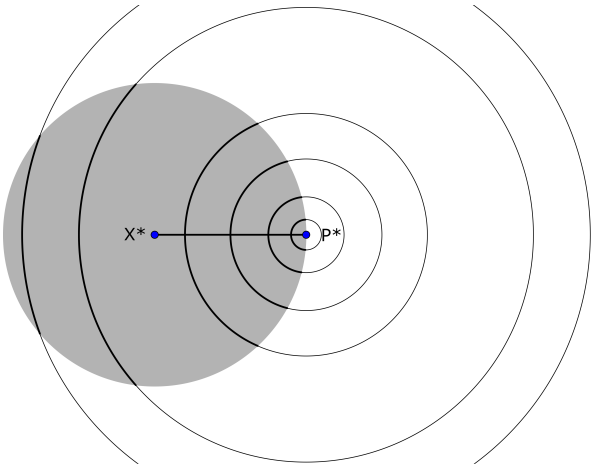


Fig. 3: Proof by picture – the probability that $\|x^\star - p'\| < \|x^\star - p^\star\|$ is monotonically decreasing in the step size of the search.

new candidate solutions from the uniform distribution on a sphere of fixed radius, r, about a current population member – why this captures the essential behaviour of Gaussian high-dimensional search will be explained shortly – and we look at the effect of varying r. More precisely we consider the probability of the event that a new candidate solution is closer to the global (or any particular local) optimum than the current population member. See Figure 3 – the point $x^\star$ is the global optimum in the search space, the point $p^\star$ is the centre (mean) of the current population, and the shaded circle represents the ball of radius $R := \|x^\star - p^\star\|$ centred on $x^\star$. Clearly a new candidate solution p' is closer to the global optimum than $p^\star$ if and only if it lies within this ball, that is when $\|x^\star - p'\| < R$. In Figure 3 we see this intersection in bold for several choices of r – in 2 dimensions this intersection is an arc, in 3 it is a spherical cap, and in 4 or more dimensions it is a hyperspherical cap. Now, what is the probability of the event $\|x^\star - p'\| < R$? Denote by S_r^{d-1} the sphere about $p^\star$ of radius r in $\mathbb{R}^d$: When p' is drawn from the uniform distribution on S_r^{d-1}, this probability is the proportion of the surface of the whole sphere comprising the intersection, namely the quotient of the surface area of the hyperspherical cap to the sphere S_r^{d-1}. For a fixed value of $\|x^\star - p^\star\|$, and for any problem dimensionality $d \geq 2$, this probability is monotonically decreasing in r for $r \in (0, 2R)$[1] and, of course, it is zero for values of $r > 2R$ in *any* dimension. Thus if the search direction from a current solution is chosen uniformly at random then, irrespective of any other consideration, larger step sizes are always more likely to take us further from the global optimum than smaller step sizes. How fast does this probability decay as a function of the step size or of the dimensionality? Define the angle of the hyperspherical cap at $p^\star$ to be $2\theta_r$, and note that the proportion of the sphere of radius r covered by this cap is the same as the proportion of the unit sphere covered by a cap on the unit sphere also with angle $2\theta_r$. Therefore $\Pr\{\|x^\star - p'\| < R\} \leq \exp(-\frac{d^2}{2}\cos^2\theta_r)$ where the RHS follows from Lemma 2.2 of [2] which upper bounds this latter quantity. By simple trigonometry one finds that $\cos\theta_r = r/2R$, and thus we obtain the following theorem:

Theorem 1 (Most Search Steps are Bad)**.** *Let $x^\star$, $p^\star$ be two fixed points in $\mathbb{R}^d$ with the Euclidean distance between them $R := \|x^\star - p^\star\|$. Let $p' = p^\star + z$ where z is sampled from the uniform distribution on the hypersphere of radius r. Then:*

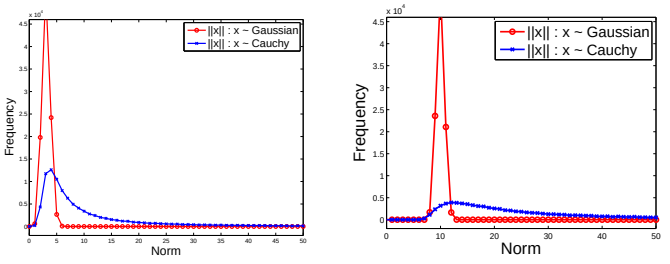
$$Pr\left\{\|x^\star - p'\| > \|x^\star - p^\star\|\right\} > 1 - \exp\left(-\frac{d^2 r^2}{8R^2}\right) \qquad (2)$$

This means that, for any fixed setting of R, the probability of sampling a point closer to the global optimum than the current reference point decays exponentially quickly in both the search radius (step size) r, and the dimensionality d. It also means that, for any choice of relative step size r/R, the proportion of good directions (i.e. directions that get us closer to the optimum than the reference point) decays exponentially quickly in the problem dimension. Therefore, if the step direction is random, large steps in high-dimensional search spaces are far less likely to take us closer to the global optimum than small steps, and thus for high-dimensional search we would expect that with very high probability heavy-tailed distributions such as the Cauchy will perform poorly. This suggests that exploration by large steps is mostly counterproductive in high dimensions and instead one should focus mainly on finding the right direction in which to move the search distribution.

Now we discuss some possible reasons why a Gaussian search distribution does better. From high dimensional prob-

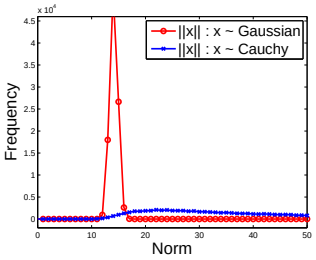
[1] In dimension 1 this probability is exactly 0.5 for a step of size $r \in (0, 2R)$.

ability theory it is known that high dimensional probability distributions may look very different from their low dimensional versions, and may therefore behave in a counter-intuitive manner. We conjecture the good performance of the Gaussian search may be due to its good concentration property, which the Cauchy distribution lacks. This property means that in high dimensions most of the points sampled from the distribution lie within a *thin shell* at approximately equal distance from the center of the distribution - in other words although in high dimensions we will not generate new points very close to the mean, neither will we generate points very far from the mean either. Figure 4 demonstrates this empirically. We sampled 100,000 points from a 10, 100, 200 and 1000-dimensional standard Gaussian and plotted the histogram of Euclidean distances from the origin (centre of the distribution). We see from the figure that all of these distances are close to approximately $\sqrt{d}$ ($\sqrt{10} = 3.16, \sqrt{100} = 10, \sqrt{200} = 14.14, \sqrt{1000} = 31.66$). So, as the dimensionality increases we have most of the points within a shell that gets thinner and thinner relative to the average distance from the centre.

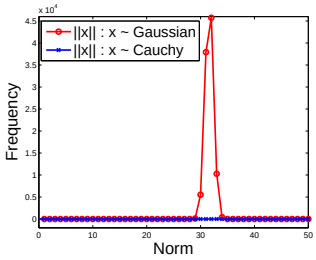


(a) D = 10, c = 2

(b) D = 100, c = 20



(c) D = 200 , c = 20



(d) D = 1000 , c = 20

Fig. 4: Comparison of the histograms of Gaussian vs. Cauchy norms as d increases. The values of the parameter c chosen here (i.e. the dimension of independent multivariate Cauchy components) correspond to a population size of 300 (although we observed no qualitiative difference for other choices). We used 100,000 sample points to create these histograms.

We then repeated the same experiment with 10, 100, 200 and 1000-dimensional Cauchy norms where 70% of the components of the points were sampled from independent c-dimensional multivariate standard Cauchy distributions and the remaining 30% from independent standard Gaussian – this mimics a typical SM & WI split from our Cauchy-EDA-MCC simulations. We superimposed these histograms on the same plots with the Gaussian norms in Figure 4. From Figure 4 it is very apparent that the Gaussian norms are all clamped in a narrow range, whereas the Cauchy norms are increasingly spread out. This will have implications on the implicit searching strategy associated with these two distributions, as we shall discuss in the remainder of this section.

Take the Gaussian case first. More formally, for a generic non-degenerate[2] $d \times d$ covariance matrix Σ, let $X \sim N(0, \Sigma)$. Then the expected norm can be approximated as follows:

$$E[||X||] \leq \sqrt{E[||X||^2]} = \sqrt{\mathrm{Tr}(\Sigma)} \tag{3}$$

using Jensen's inequality. Indeed, applying the linearity of expectation, we have $E[||X||^2] = E[\sum_{i=1}^{d} X_i^2] = \sum_{i=1}^{d} E[X_i^2] = \sum_{i=1}^{d} (\Sigma_{ii}) = \mathrm{Tr}(\Sigma)$.

Note that in the case $\Sigma = I$ we have $\sqrt{\mathrm{Tr}(\Sigma)} = \sqrt{d}$. This is why we saw the averages of Gaussian norms at approximately $\sqrt{d}$ in Figure 4. Furthermore, the following lemma shows that with high probability $\|X\|$ is close to $\sqrt{\mathrm{Tr}(\Sigma)}$ (in absolute difference relative to the spectral norm of Σ).

Lemma 1. *Let $X \in R^d$ where X has entries drawn from a multivariate Gaussian with mean zero and Σ covariance. Then, $\forall \epsilon \in (0, 1)$,*

$$Pr\left\{ \left| \|X\| - \sqrt{Tr(\Sigma)} \right| \geq \epsilon \sqrt{\lambda_{\max}(\Sigma)} \right\} \leq 2 \exp\left[-\frac{\epsilon^2}{2} \right] \tag{4}$$

This probability inequality was mentioned in [8] without proof. In the Appendix we derive it from Lemma 1 of [4].

Now, Lemma 1 implies that in Gaussian EDA search, a large fraction of the new generation lies in a thin shell at the same distance from the center of the population – therefore selection of the fittest points essentially selects the promising *directions*. These two elements – using all of the available resources to select directions, and then ensuring a steady move of size just below $\sqrt{Tr(\Sigma)}$ from the center of the population from one generation to the next – provide Gaussian EDA a well focused strategy that is beneficial and resource-efficient. Moreover, as we approach a local optimum $Tr(\Sigma)$ will decay, so in fact Gaussian EDA automatically tunes the search granularity over successive generations.

By contrast, the Cauchy density does not have good concentration properties. This is very apparent from the numerical experiment in Figure 4. While we see a reasonably high density region in the case of $d = 10$, as d increases, the heavy tails of the distribution in all directions dissolve any high density region. Therefore, Cauchy based search has no ability to prioritize selecting good directions.

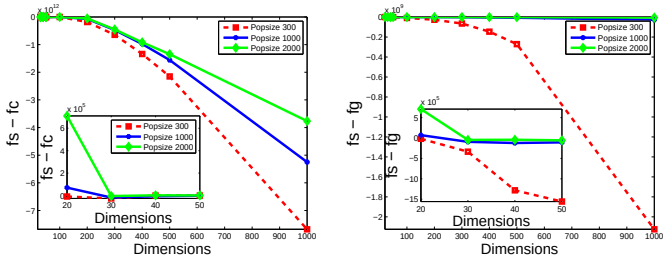
In the sequel we shall put the above explanation to a test: We shall create a new search distribution for EDA that takes to the extreme the clever implicit searching strategy of Gaussian EDA that we just uncovered. If our reasoning above is correct, then the new search distribution might perform even better in high dimensions.

[2]Note that the model complexity control on the covariance estimates in EDA-MCC ensures that the covariance estimates are indeed non-degenerate – of course, provided that we set the parameters c and m wisely (as discussed in an earlier section).

VI. EDA WITH UNIFORM SEARCH DISTRIBUTION ON A HYPERSPHERE

Rather than searching in a thin shell at some constant distance from the center of the population, let us search precisely on the hypersphere with the same radius. Based on our analysis in the previous section, from eqs. (3)-(4), we define the search distribution as a uniform distribution on the sphere of radius $\sqrt{\mathrm{Tr}(\Sigma)}$, where, as before, Σ is the covariance estimated from the selected individuals. This way, when the high fitness individuals are selected they represent exactly the high fitness directions at granularity equal to the radius. The subsequent generation then makes a steady move towards the average of the selected directions, just like it was the case for Gaussian based search.

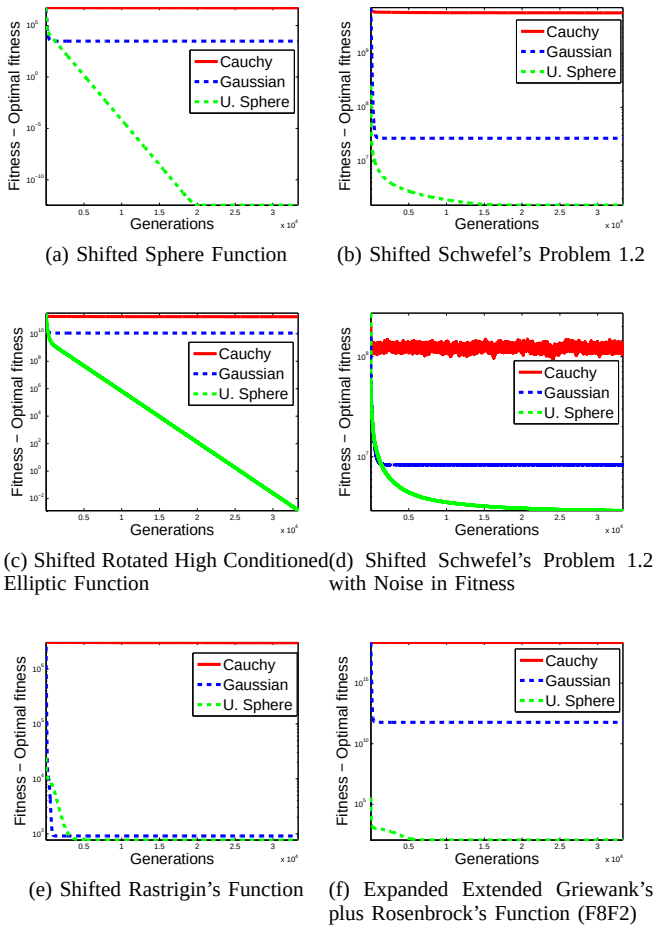
We tested and validated the performance of this new EDA variant in an extensive series of experiments, comparatively with both the Gaussian and the Cauchy EDA variants discussed earlier. We first present detailed results on the search process for the Shifted Rosenbrock function in Figure 5, with three different population sizes, each tested on four different dimensions of the problem, from low to high. As conjectured, we can see that the uniform sphere based search strategy becomes increasingly efficient in high dimensions and outperforms both Cauchy and Gaussian based EDA search as the dimensionality of the problem increases. We confirmed using ranksum tests that these differences are statistically significant. This is because in an exponentially increasing search space, when only having a linearly increasing budget it becomes more and more important to prioritize the task of selecting good directions. We also see that this effect is very robust and not influenced by the particular choice of population size. All plots represent average of best fitness as computed from 25 independent runs. The total budget was set to $10^4 \cdot d$, where d is the dimension of the problems.



(a) Uniform Sphere vs. Cauchy (b) Uniform Sphere vs. Gaussian

Fig. 5: Differences between the average (from 25 repeated runs) of the best fitness values achieved by the Gaussian (fg) and by the Uniform on Sphere (fs) EDAs, as the dimension is varied, for the Shifted Rosenbrock function. The smaller plots superimposed represent zoomed versions of the same results in the range of 20-50 dimensions.

Finally, in Figure 6 we demonstrate the results of large scale experiments in 1000-dimensions on the remaining 6 benchmark function listed in Table I. Here we used a population size of $N = 300$. Again we see that UniformSphere-EDA consistently and significantly outperforms the other two EDA variants. From these results, and recalling our rationale for



(a) Shifted Sphere Function (b) Shifted Schwefel's Problem 1.2

(c) Shifted Rotated High Conditioned Elliptic Function (d) Shifted Schwefel's Problem 1.2 with Noise in Fitness

(e) Shifted Rastrigin's Function (f) Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)

Fig. 6: Comparison of Gaussian EDA-MCC, Cauchy EDA-MCC and UniformSphere-EDA on 1000-dimensional problems. The population size was 300, and each curve is the average of the best fitness values from 25 independent runs. The budget of function evaluations was $10^4 \cdot d$, where d is the dimension of the problem.

creating this new EDA version, we conclude that our study resolved the controversy about the merits of Gaussian against Cauchy EDA search in high dimensional problems, and as a byproduct our new EDA variant also gives us new insights about how to approach high dimensional EDA search.

VII. CONCLUSIONS

In this paper, we conducted a large empirical study to benchmark the performance of Cauchy and Gaussian search distributions in EDA using a scalable black-box EDA optimizer. Our empirical results suggest that Cauchy search distributions perform particularly badly in high-dimensional spaces. To explain this phenomenon we developed theory that explains why large search steps are inefficient in high dimensional search spaces, and we showed that this inefficiency is

unavoidable in practice. We argued that a Gaussian search distribution has an in-built prioritizing strategy that implicitly focuses resources within a generation on selecting good search directions: This strategy is a by-product of the concentration property of Gaussian norms in high dimensions. On the other hand, Cauchy norms lack good concentration properties and make a high proportion of (very) large steps, and this results in an increasingly inefficient search strategy when the problem dimension increases. Based on our theoretical insights and understanding of high dimensional domains, we proposed a minor modification to the standard Gaussian EDA which enforces search within a generation to all take place at a fixed radius of the current population centre. Initial experiments on a battery of test problems indicate that this simple change improves high dimensional search markedly – fuller evaluation of the promise of this approach remains for future work.

REFERENCES

[1] Yuan. B and Gallagher. M. On the importance of diversity maintenance in estimation of distribution algorithms. In *In H.G. Beyer and U.M OReilly, editors, Proceedings of the Genetic and Evolutionary Computation Conference GECCO 05, Volume 1, Pages 719-729, New York, NY, USA.*, 2005.

[2] K. Ball. *An Elementary Introduction to Modern Convex Geometry. In: Flavors of Geometry*. MSRI Publications, 1997.

[3] W. Dong, T Chen, P. Tino, and X. Yao. Scaling up estimation of distribution algorithm for continuous optimisation. *IEEE Transaction of Evolutionary Computation. Vol 17, Issue 6.*, 2013.

[4] R. J. Durrant and A. Kabán. Error bounds for kernel fisher linear discriminant in gaussian hilbert space. In *Proceedings of the 15th international Conference on Artificial Intelligence and Statistics(AISTATS)*, 2012.

[5] N. Hansen. Benchmarking a bi-population cma-es on the bbob-2009 function testbed. In *GECCO, (Companion), pages 23892396.*, 2009.

[6] N. Hansen, F. Gemperle, A. Auger, and P. Koumoutsakos. When do heavy-tail distributions help? In *PPSN*, 2006.

[7] M. Hausschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. Technical report, Missouri Estimation of Distribution Algorithms Laboratory, St. Louis., 2011.

[8] A. Kabán. Non-asymptotic analysis of compressive fisher discriminants in terms of the effective dimension. In *7th Asian Conference on Machine Learning (ACML 2015), Journal of Machine Learning Research-Proceedings Track*, pages 17–32, 2016.

[9] A. Kabán, J. Bootkrajang, and R.J. Durrant. Towards large scale continuous eda: A random matrix theory perspective. *Evolutionary Computation, MIT Press*, 2015.

[10] P. Larranaga and J.A Lozano. *Estimation of Distribution Algorithms: A new tool for Evolutionary Computation.* Kluwer Academic Publishers (2001), 2001.

[11] M. N. Omidvar and X. Li. A comparative study of CMA-ES on large scale global optimisation. In *AI 2010: Advances in Artificial Intelligence*, pages 303–312, 2011.

[12] D. Peel and G. McLachlan. Robust mixture modelling using the t distribution. *Statist. Comput.*, 2000.

[13] P. Posik. Bbob-benchmarking a simple estimation-of-distribution algorithm with cauchy distribution. In *Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference, GECCO, pages 2309-2314. New York, NY, USA, 2009. ACM.*, 2009.

[14] P. Posik. Comparison of cauchy eda and bipop-cma-es algorithms on the bbob noiseless testbed. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation.*, pages 1697–1702, 2010.

[15] P. Posik. Comparison of cauchy eda and g3pcx algorithms on the bbob noiseless testbed. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO).*, 2010.

[16] M. L. Sanyang and A. Kabán. Multivariate cauchy eda optimisation. In *IDEAL 2014, LNCS 8669, pp. 449456*, 2014.

[17] M. L. Sanyang and A. Kabán. Heavy tails with parameter adaptation in random projection based continuous eda. In *2015 IEEE Congress on Evolutionary Computation (CEC), page 2074-2081. IEEE,*, 2015.

[18] T. Schaul, T. Glasmachers, and J. Schmidhuber. High dimensions and heavy tails for natural evolution strategies. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 845–852, New York, NY, USA, 2011. ACM.

[19] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, and S. Tiwari., editors. *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real Parameter Optimisation*, 2005.

[20] Y. Wang and B. Li. A restart univariate estimation of distribution algorithm: Sampling under mixed gaussian and lévy probability distribution. In *Proc. IEEE Congress on Evolutionary Computation (CEC 2008)*, pages 3917–3924. IEEE, 2008.

[21] Q. Xu. Large scale eda optimisation. Msc Project Report, University of Birmingham, 2016.

[22] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation*, VOL. 3,, 1999.

APPENDIX – PROOF OF LEMMA 1

Proof. The following bounds [4] hold for the Gaussian square norm, with the two sides holding with different probabilities. Here we massage this into a bound on the Gaussian norm and make the two sides hold with the same probability. From [4]:

$$Pr\left\{ \|X\| \geq \sqrt{(1+\epsilon)Tr(\Sigma)} \right\} \leq \exp\left(-\frac{Tr(\Sigma)(\sqrt{1+\epsilon}-1)^2}{2\lambda_{\max}(\Sigma)} \right) \quad (5)$$

$$Pr\left\{ \|X\| \leq \sqrt{(1-\epsilon)Tr(\Sigma)} \right\} \leq \exp\left(-\frac{Tr(\Sigma)(\sqrt{1-\epsilon}-1)^2}{2\lambda_{\max}(\Sigma)} \right) \quad (6)$$

Now from the LHS of equation 5 and our target, we set:

$$\sqrt{Tr(\Sigma)+\epsilon Tr(\Sigma)} = \sqrt{Tr(\Sigma)} + \tau\sqrt{\lambda_{\max}(\Sigma)}$$

Solving for ϵ, and replacing it into the RHS of eq.5 gives:

$$\exp\left[-\frac{Tr(\Sigma)\left(\sqrt{1+\tau[2\sqrt{\frac{\lambda_{\max}(\Sigma)}{Tr(\Sigma)}}+\frac{\tau\lambda_{\max}(\Sigma)}{Tr(\Sigma)}]}-1 \right)^2}{2\lambda_{\max}(\Sigma)} \right]$$

$$= \exp\left[-\frac{Tr(\Sigma)}{2}\left(\sqrt{\frac{1}{\lambda_{\max}(\Sigma)}+\frac{2\tau}{\sqrt{Tr(\Sigma)\lambda_{\max}(\Sigma)}}+\frac{\tau^2}{Tr(\Sigma)}}-\frac{1}{\sqrt{\lambda_{\max}(\Sigma)}} \right)^2 \right]$$

Taking LCM of the term under the square root, we have

$$= \exp\left[-\frac{Tr(\Sigma)}{2}\left(\sqrt{\frac{Tr(\Sigma)+2\tau\sqrt{Tr(\Sigma)\lambda_{\max}(\Sigma)}+\tau^2\lambda_{\max}(\Sigma)}{Tr(\Sigma)\lambda_{\max}(\Sigma)}}-\frac{1}{\sqrt{\lambda_{\max}(\Sigma)}} \right)^2 \right]$$

$$= \exp\left[-\frac{Tr(\Sigma)}{2}\left(\sqrt{\frac{(\sqrt{Tr(\Sigma)}+\tau\sqrt{\lambda_{\max}(\Sigma)})^2}{Tr(\Sigma)\lambda_{\max}(\Sigma)}}-\frac{1}{\sqrt{\lambda_{\max}(\Sigma)}} \right)^2 \right]$$

$$= \exp\left[-\frac{Tr(\Sigma)}{2}\left(\frac{\sqrt{Tr(\Sigma)}+\tau\sqrt{\lambda_{\max}(\Sigma)}}{\sqrt{Tr(\Sigma)\lambda_{\max}(\Sigma)}}-\frac{1}{\sqrt{\lambda_{\max}(\Sigma)}} \right)^2 \right]$$

Now taking LCM of the term inside the square, and simplifying, we get:

$$= \exp\left[-\frac{Tr(\Sigma)}{2}\left(\frac{\tau\sqrt{\lambda_{\max}(\Sigma)}}{\sqrt{Tr(\Sigma)\lambda_{\max}(\Sigma)}} \right)^2 \right] = \exp\left[-\frac{\tau^2}{2} \right]$$

after cancellations. Rename τ by ϵ, and this completes the proof for one side of Lemma 1. The other side is analogous, and yields:

$$Pr\left\{ \|X\| - \sqrt{Tr(\Sigma)} \leq -\epsilon\sqrt{\lambda_{\max}(\Sigma)} \right\} \leq \exp\left[-\frac{\epsilon^2}{2} \right]$$

□