A New EDA by a Gradient-Driven Density

Ignacio Segovia Domínguez, Arturo Hernández Aguirre, and S. Ivvan Valdez

Center for Research in Mathematics, Guanajuato, México
{ijsegoviad,artha,ivvan}@cimat.mx

Abstract. This paper introduces the Gradient-driven Density Function (∇_dD) approach, and its application to Estimation of Distribution Algorithms (EDAs). In order to compute the ∇_dD, we also introduce the Expected Gradient Estimate (EGE), which is an estimation of the gradient, based on information from other individuals. Whilst EGE delivers an estimation of the gradient vector at the position of any individual, the ∇_dD delivers a statistical model (e.g. the normal distribution) that allows the sampling of new individuals around the direction of the estimated gradient. Hence, in the proposed EDA, the gradient information is inherited to the new population. The computation of the EGE vector does not need additional function evaluations. It is worth noting that this paper focuses in black-box optimization. The proposed EDA is tested with a benchmark of 10 problems. The statistical tests show a competitive performance of the proposal.

Keywords: Gradient estimation, Estimation of Distribution Algorithm.

1 Introduction

Several Evolutionary Algorithms search the global optimum by simulations from statistical models; e.g. EDAs, ES, etc. The evolutionary computation community has been making a large effort to add new information into statistical models in order to improve the search process. There are several approaches to add search directions into statistical models [3] [2]. In this context, some popular algorithms have demonstrated the feasibility of this idea (e.g. CMA-ES, NES, etc.). *This paper introduces contributions in this trend by building density functions based on gradient estimates: Gradient-driven densities.* The proposal developed here only use the function evaluations gathered from the population to build gradient estimates on fixed individuals. Hence, the algorithm does not require any extra evaluation of function. The first-order information is an important source of promising directions to improve any individual. For that reason, a Gradient-driven Density Function (∇_dD) is introduced. Any simulation from ∇_dD might produce samples around the gradient estimation. Hence, the search process focuses in promising orientations. These novel ideas are merged to create a new EDA. As a consequence of the gradient estimation, the proposed EDA generates new individuals towards promising regions. The organization of the paper is as follows. Section 2 introduces the Expected Gradient Estimation method. Section 3 develops the Gradient-driven Density framework. Section 4 presents the

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 352–361, 2014.

EDA based on Gradient-driven Density Functions. Section 5 is devoted to test the proposed EDA against others algorithms from the state of the art. Finally, Section 6 provides some concluding remarks.

2 The Gradient Estimation

The gradient vector $\nabla \mathcal{F}(\boldsymbol{x})$ models the local greatest rate of increase by specifying a direction and magnitude at $\boldsymbol{x}$. Since the information about the problem comes from scattered samples on the search space, a neighborhood for an individual might be choosen. For that reason, this paper considers that any gradient estimate for individual $\boldsymbol{x}^{(i)}$ requires itself and its neighborhood, i.e. a set of individuals $\mathcal{N}_{\boldsymbol{x}^{(i)}} = \{\boldsymbol{x}^{(i_1)}, \boldsymbol{x}^{(i_2)}, ..., \boldsymbol{x}^{(i_k)}, ..., \boldsymbol{x}^{(i_{r-1})}, \boldsymbol{x}^{(i_r)}\}$ from the population or gathered from previous generations, where $i \neq i_1 \neq ... \neq i_k \neq ... \neq i_r$ and r is the neighborhood size. Furthermore, any criterion to select the neighborhood can be used. Also, notice that this method permits to compute a gradient estimate for each individual only by using known information about the problem. The fitness values of $\{\boldsymbol{x}^{(i)}, ..., \boldsymbol{x}^{(i_1)}, ..., \boldsymbol{x}^{(i_r)}\}$ provide knowledge about the problem. Hence, that information can be used to estimate the gradient vector of $\boldsymbol{x}^{(i)}$. The common approach approximates the gradient by fitting a hyperplane in $d+1$ dimensions, where d is the dimension size of $\boldsymbol{x}^{(i)}$. Therefore, the estimation of gradient might be tackled by the ordinary least square method [4]. Despite the fact that the previous technique creates an intuitive gradient approximation, in many contexts it might be inadequate (e.g. there are not enough samples to create the overdetermined system, etc). This section presents a new gradient estimation based on two mathematical concepts: *the directional derivative and the statistical expectation.*

Definition 1. *Let $\mathcal{N}_{\boldsymbol{x}^{(i)}} = \{\boldsymbol{x}^{(i_1)}, \boldsymbol{x}^{(i_2)}, ..., \boldsymbol{x}^{(i_k)}, ..., \boldsymbol{x}^{(i_{r-1})}, \boldsymbol{x}^{(i_r)}\}$ be the neighbors of individual $\boldsymbol{x}^{(i)}$, from the population. Then the Expected Gradient Estimate for $\boldsymbol{x}^{(i)}$ is defined by*

$$\widehat{\nabla \mathcal{F}}(\boldsymbol{x}^{(i)}) = \frac{1}{r} \sum_{k=1}^{r} \frac{\mathcal{F}(\boldsymbol{x}^{(i_k)}) - \mathcal{F}(\boldsymbol{x}^{(i)})}{\|\boldsymbol{x}^{(i_k)} - \boldsymbol{x}^{(i)}\|^2} (\boldsymbol{x}^{(i_k)} - \boldsymbol{x}^{(i)}) \tag{1}$$

where $i \neq i_1 \neq ... \neq i_k \neq ... \neq i_r$, r is the neighborhood size and $\mathcal{F}(\cdot)$ computes the fitness value.

In order to justify equation (1), let us assume that $\boldsymbol{x}^{(i_k)}$ exists on the line defined by the true gradient $\nabla \mathcal{F}(\boldsymbol{x}^{(i)})$. This means

$$\frac{\boldsymbol{x}^{(i_k)} - \boldsymbol{x}^{(i)}}{\|\boldsymbol{x}^{(i_k)} - \boldsymbol{x}^{(i)}\|} = \pm \frac{\nabla \mathcal{F}(\boldsymbol{x}^{(i)})}{\|\nabla \mathcal{F}(\boldsymbol{x}^{(i)})\|}. \tag{2}$$

Since the orientation depends on the sign, each case will be examined separately. Let $\boldsymbol{u}_+$ and $\boldsymbol{u}_-$ be two normalized vectors as follows

$$\boldsymbol{u}_+ = \frac{\nabla \mathcal{F}(\boldsymbol{x}^{(i)})}{h}, \quad \boldsymbol{u}_- = -\frac{\nabla \mathcal{F}(\boldsymbol{x}^{(i)})}{h} \tag{3}$$

where $h = \|\nabla\mathcal{F}(\boldsymbol{x}^{(i)})\|$; please note $\boldsymbol{u}_+$ has the same direction as the true gradient, opposite to $\boldsymbol{u}_-$. From the well-known directional derivative definition and its properties observe that

$$\left(\lim_{h\to 0}\frac{\mathcal{F}(\boldsymbol{x}^{(i)}+h\boldsymbol{u}_+)-\mathcal{F}(\boldsymbol{x}^{(i)})}{h}\right)\boldsymbol{u}_+ = \|\nabla\mathcal{F}(\boldsymbol{x}^{(i)})\|\frac{\nabla\mathcal{F}(\boldsymbol{x}^{(i)})}{\|\nabla\mathcal{F}\left(\boldsymbol{x}^{(i)}\right)\|} = \nabla\mathcal{F}(\boldsymbol{x}^{(i)})$$

$$\left(\lim_{h\to 0}\frac{\mathcal{F}(\boldsymbol{x}^{(i)}+h\boldsymbol{u}_-)-\mathcal{F}(\boldsymbol{x}^{(i)})}{h}\right)\boldsymbol{u}_- = -\|\nabla\mathcal{F}(\boldsymbol{x}^{(i)})\|\frac{-\nabla\mathcal{F}(\boldsymbol{x}^{(i)})}{\|\nabla\mathcal{F}\left(\boldsymbol{x}^{(i)}\right)\|} = \nabla\mathcal{F}(\boldsymbol{x}^{(i)})$$

This is important because similar connections can be found just by considering two individuals, mainly due to the assumption in equation (2); for instance

$$\left(\lim_{l\to 0}\frac{\mathcal{F}(\boldsymbol{x}^{(i)}+l\cdot\boldsymbol{u})-\mathcal{F}(\boldsymbol{x}^{(i)})}{l}\right)\boldsymbol{u} = \nabla\mathcal{F}(\boldsymbol{x}^{(i)}) \tag{4}$$

$$\boldsymbol{u} = \frac{\boldsymbol{x}^{(i_k)}-\boldsymbol{x}^{(i)}}{\|\boldsymbol{x}^{(i_k)}-\boldsymbol{x}^{(i)}\|}, \quad l = \|\boldsymbol{x}^{(i_k)}-\boldsymbol{x}^{(i)}\| \tag{5}$$

Equation (4) presents a different manner to remake the gradient function. Notice that although the derivative is unavailable, an approximation by finite differences can be considered. It leads us to introduce a gradient estimate for $\boldsymbol{x}^{(i)}$, just given *one* neighbor, as follows

$$\boldsymbol{g}^{(i_k)} = \left(\frac{\mathcal{F}(\boldsymbol{x}^{(i)}+l\boldsymbol{u})-\mathcal{F}(\boldsymbol{x}^{(i)})}{l}\right)\boldsymbol{u} = \frac{\mathcal{F}(\boldsymbol{x}^{(i_k)})-\mathcal{F}(\boldsymbol{x}^{(i)})}{\|\boldsymbol{x}^{(i_k)}-\boldsymbol{x}^{(i)}\|^2}(\boldsymbol{x}^{(i_k)}-\boldsymbol{x}^{(i)}) \tag{6}$$

However, there is no chance to ensure that $\boldsymbol{x}^{(i_k)}$ is on the line defined by the true gradient, because the neighbors come from an unknown hidden random process. Hence, the difference between fitness values is also a random variable. Therefore, each estimate $\boldsymbol{g}^{(i_k)}$ arises from a random process. Please assume that $\mathcal{P}$ is the hidden uncertainty model which describes the behavior of outcomes $\boldsymbol{g}^{(i_k)}$. So, any instance of random variable $\boldsymbol{g}^{(i)} \sim \mathcal{P}$ is an outcome $\boldsymbol{g}^{(i_k)}$. A representative vector for the hidden model can be computed by the statistical expectation. Moreover, the $E(\boldsymbol{g}^{(i)})$ can be approximated as follows

$$E(\boldsymbol{g}^{(i)}) = \int_{\mathbb{R}^d}\boldsymbol{g}^{(i)}\mathcal{P}d\boldsymbol{g}^{(i)} \approx \frac{1}{r}\sum_{k=1}^{r}\boldsymbol{g}^{(i_k)} \tag{7}$$

which is the *Expected Gradient Estimate*, equation (1).

To the best of our knowledge, the EGE developed above has not been addressed in literature. However, further theoretical study is necessary to verify its relationship with other approaches [1]. In order to empirically contrast the approximated orientations of EGE versus the usual approximation by hyperplane, a fixed population and a gradient estimation on each individual will be considered. An ideal population, at first generation, must cover the search domain

evenly; thus in this experiment the population is built by the Halton quasi-random sequence, from Matlab® with default options. Also, please consider the Sphere problem, the neighborhood size $r = d + 1$ and the r closer individuals to $\boldsymbol{x}^{(i)}$ (neighborhood, according to the Euclidean distance). Below there is an angle-comparison between $\widehat{\nabla \mathcal{F}}(\boldsymbol{x}^{(i)})$ and $\nabla \mathcal{F}(\boldsymbol{x}^{(i)})$. So, the measurement vector of angles $\boldsymbol{\alpha} = \{\alpha^{(1)}, ..., \alpha^{(i)}, ..., \alpha^{(N)}\}$ includes an angle value for each individual, where N is the population size. Figure 1 shows the histograms of orientation by

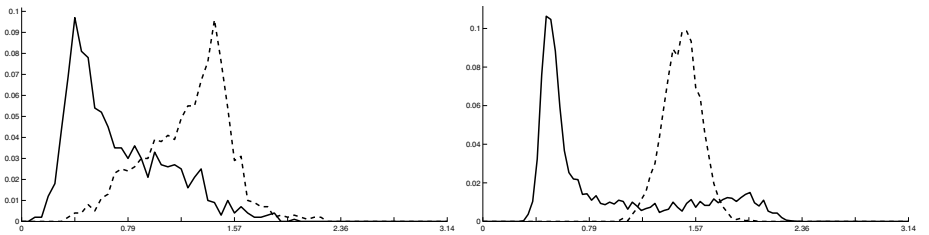

Fig. 1. Histograms of orientation in Sphere problem, Expected Gradient Estimation (EGE, solid line) versus hyperplane approach by ordinary least square (HLS, dashed line). (a) 10 dimensions: HLS has a median value of 1.30488 and EGE has a median value of 0.627966. (b) 30 dimensions: HLS has a median value of 1.48743 and EGE has a median value of 0.617965. EGE shows better performance.

setting $r = d + 1$, $N = 10d$ and $x_k \in [-600, 300]$; where d is the dimension size. Notice that the EGE has more chances to compute better oriented vectors than the hyperplane approach for higher dimensions. In addition, the median values support the graphic observation. Here, the neighborhood size $r = d + 1$ was chosen from literature [4]. In summary, the results suggest that the EGE might outperform previous gradient approximations used in evolutionary computation.

3 The Gradient-Driven Density

Each generation has three different data sets: the individuals $\{\boldsymbol{x}^{(i)}\}$, the function evaluations $\{\mathcal{F}(\boldsymbol{x}^{(i)})\}$ and the estimations of the gradient estimates $\{\widehat{\nabla \mathcal{F}}(\boldsymbol{x}^{(i)})\}$. These provide distinct information about the function and the algorithms' behavior. Several stochastic optimization approaches (e.g. ES, EDA) aim to build a multivariate density function on optimum locations; or at least in better regions than the current ones. This section will begin with the same goal for a fixed individual from the population. Therefore, there might exist a multivariate density function $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(i)}$ based on its gradient estimate $\widehat{\nabla} \mathcal{F}(\boldsymbol{x}^{(i)})$, which is able to simulate better individuals than the present $\boldsymbol{x}^{(i)}$. Due to the fact that only two vectors will be used here, there is no chance to ensure that all simulations improve the current fitness value $\mathcal{F}(\boldsymbol{x}^{(i)})$. However, the density modeling

can be modified to take advantage of the gradient estimate by developing a new estimation of parameters, updating the original parameters, improving the simulation, etc. For this reason, definition 2 introduces the ∇_dD from the individual perspective.

Definition 2. *Let* $\boldsymbol{z}$ *be an individual in the domain space and* $G(\boldsymbol{z})$ *a function which computes its gradient estimate. The density function* $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ *is a Gradient-driven Density (*$\nabla_d D$*) for individual* $\boldsymbol{z}$ *if the following two conditions are satisfied:*

1) The multivariate density $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ *is a unimodal function,*
2) $\frac{G(\boldsymbol{z})}{\|G(\boldsymbol{z})\|} = \frac{\nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})}{\|\nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})\|}$.

The conditions set up above allow a wide range of future proposals. The first condition permits a single mass of probability towards promising regions. The random search must be led by $G(\boldsymbol{z})$ because it is orienting towards promising regions. In fact, the second condition just allows density functions which $\nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})$ has the same direction as the gradient estimate $G(\boldsymbol{z})$. There are many ways to build a ∇_d Density. Below, a suitable technique based on multivariate calculus and the angular discrepancy are introduced .

Definition 3. *Let* $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ *be a multivariate unimodal density. In order to ensure that* $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ *is a* ∇_d *Density, a parameter estimation on* $\boldsymbol{\theta}$ *must be performed. The minimum angle estimation solves this by,*

$$\widehat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta}} \frac{G(\boldsymbol{z})^t \nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})}{\|G(\boldsymbol{z})\| \, \|\nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})\|} \tag{8}$$

Notice that the *minimum angle estimation* solves the parameter estimation of $\boldsymbol{\theta}$ by maximizing the dot product of two normalized vectors. It is a natural way because the angle between two vectors is related to the dot product. In addition, even if finding the solution of (8) is not possible, a good approximation can be discovered. The rest of this section uses the previous definition to build ∇_d densities. Please assume that $\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta})$ is a *multivariate normal density* and $\nabla \mathrm{p}(\boldsymbol{z}, \boldsymbol{\theta})$ its gradient function. Also, let

$$\boldsymbol{z}_G = \frac{G(\boldsymbol{z})}{\|G(\boldsymbol{z})\|} \tag{9}$$

be the normalized gradient estimate of individual $\boldsymbol{z}$. The estimation method must calculate the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ which satisfy definition 2. Then, the statistical parameters can be found by solving

$$< \boldsymbol{\mu}^{new}, \boldsymbol{\Sigma}^{new} > = \max_{<\boldsymbol{\mu}, \boldsymbol{\Sigma}>} \frac{\boldsymbol{z}_G^t \left[\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \boldsymbol{z}) \right]}{\| \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \boldsymbol{z}) \|}. \tag{10}$$

Let us address this problem separately for each parameter. By taking the derivative with respect to $\boldsymbol{\mu}$ and setting it equal to zero, we found the equation

$$\|\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z})\|^2\boldsymbol{\Sigma}^{-t}\boldsymbol{z}_G - \boldsymbol{z}_G^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z})\boldsymbol{\Sigma}^{-t}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z}) = 0. \quad (11)$$

In a similar way, by taking the derivative with respect to $\boldsymbol{\Sigma}^{-1}$ and setting it equal to zero, we found the equation

$$\|\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z})\|^2\boldsymbol{z}_G(\boldsymbol{\mu}-\boldsymbol{z})^t - \boldsymbol{z}_G^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z})(\boldsymbol{\mu}-\boldsymbol{z})^t = 0. \quad (12)$$

Notice that both are nonlinear matrix equations! In addition, the problem (12) is a constraint equation, since $\boldsymbol{\Sigma}$ needs to be a symmetric positive semidefinite matrix. So, it leads us to solve two more complex optimization problems than the original one. However, a few interesting facts arise by inspecting equations (10)-(12), when $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z}) = \boldsymbol{z}_G$: Equation (10) reaches its maximum value, i.e. 1; and Equations (11) and (12) are fulfilled. Furthermore, notice that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are closely related. In fact, there are an infinite number of symmetric semipositive definite matrices $\boldsymbol{\Sigma}$ able to fulfill $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z}) = \boldsymbol{z}_G$. This certainly means that the nonlinear system has an infinity number of solutions. However, straightforward solutions can be found by these observations. By assuming the matrix $\boldsymbol{\Sigma}$ is fixed and solving for the mean vector in $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{z}) = \boldsymbol{z}_G$ a new formula is found:

$$\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}^{new}-\boldsymbol{z}) = \boldsymbol{z}_G \qquad \therefore \qquad \boldsymbol{\mu}^{new} = \boldsymbol{z} + \boldsymbol{\Sigma}\boldsymbol{z}_G. \quad (13)$$

Given a fixed covariance matrix, its related mean vector can be computed by (13). Furthermore, there is a unique mean vector for a given $\boldsymbol{\Sigma}$. On the contrary, given a fixed mean vector, there is a number of infinite possible covariance matrices. Definition 4 introduces a ∇_d Density based on the previous analysis.

Definition 4. *Let $\boldsymbol{\Sigma}_0$ be a fixed covariance matrix. Then the ∇_d Normal ($\nabla_d\mathcal{N}$) has parameters*

$$\boldsymbol{\mu}_g = \boldsymbol{z} + \boldsymbol{\Sigma}_0\boldsymbol{z}_G \qquad \textit{and} \qquad \boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}_0 \quad (14)$$

Notice that, simulations from the proposed densities will produce samples in a similar direction as the gradient vector (or gradient estimate). The next section applies the developed *Gradient-driven densities* in evolutionary computation.

4 The Gradient-Driven Density in EDAs

The Estimation of Distribution Algorithm (EDA) aims to simulate new individuals on regions near optimum locations, preferably close to the global optimum. Interesting optimization methods might be developed by considering the ∇_dD technique into EDAs. The EDA fits a target statistical model. A common one is the multivariate normal function [5]. This section introduces an EDA, based on

this density function, by computing the expectation and variance of a multivariate Gaussian mixture model. Please consider a mixture of two models: the *empirical normal density* and a *Gradient-driven Density*. The first one promotes the exploitation whilst the second one allows predictive samples on possible promising regions (exploration). In order to build a simpler model, the mixture of densities is approximated by a unique Multivariate Gaussian model [6]. The target density for the proposed EDA is built by $\mathcal{N}(\boldsymbol{\mu}^{new}, \boldsymbol{\Sigma}^{new})$, where:

$$\begin{aligned}\boldsymbol{\mu}^{new} &= E(E(\boldsymbol{X}|\vartheta)) = (1-\beta)\widehat{\boldsymbol{\mu}} + \beta\boldsymbol{\mu}_g, \\ \boldsymbol{\Sigma}^{new} &= Var(E(\boldsymbol{X}|\vartheta)) + E(Var(\boldsymbol{X}|\vartheta)) = (1-\beta)\widehat{\boldsymbol{\Sigma}} + \beta\boldsymbol{\Sigma}_g \\ &\quad + (1-\beta)(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}^{new})(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}^{new})^t + \beta(\boldsymbol{\mu}_g - \boldsymbol{\mu}^{new})(\boldsymbol{\mu}_g - \boldsymbol{\mu}^{new})^t\end{aligned} \tag{15}$$

and $\beta \in [0, 1]$ is the associated weight to the ∇_dD. Also, β controls the amount of credibility on each model. Please note that $\beta = 0$ produces the empirical density and $\beta = 1$ yields the other one. In addition, since the simulation method might build samples outside the search domain, a re-insertion technique is added, line 10 of algorithm 2. Let $\gamma_k = l_k^{upper} - l_k^{lower}$ be the domain length in dimension k, where l_k^{upper} and l_k^{lower} are the upper bound and lower bound in dimension k. For each dimension, the new sample $\boldsymbol{y}^{(i)} = (y_1^{(i)}, \cdots, y_k^{(i)}, \cdots, y_D^{(i)})$ is tested/replaced by

- if $y_k^{(i)} > l_k^{upper}$ then $a = (y_k^{(i)} - l_k^{upper})/\gamma_k$ and $y_k^{(i)} = l_k^{upper} - \gamma_k(a - \lfloor a \rfloor)$
- if $y_k^{(i)} < l_k^{lower}$ then $a = (l_k^{lower} - y_k^{(i)})/\gamma_k$ and $y_k^{(i)} = l_k^{lower} + \gamma_k(a - \lfloor a \rfloor)$

which ensure any new individual is inside the domain. The algorithm 2 describes the proposed EDA led by a Gradient-driven Density (EDA-LGD). Because of the importance of the gradient estimate for the ∇_dD, this proposal just computes the gradient of the best individual using the historical best individuals from previous generations. So, if at generation (t) a new best individual $\boldsymbol{x}^{best}$ is found, then $\boldsymbol{x}^{best}$ replaces the worst individual in P_{best} and the next gradient estimate is over $\boldsymbol{x}^{best}$ with the neighborhood $\{P_{best} \setminus \boldsymbol{x}^{best}\}$. Then two populations are saved: the usual population Pob_t at each generation (t) and the historically best individuals P_{best}; in algorithm 2 the first one has N individuals meanwhile the second one has $d + 2$ individuals.

5 Experiment

This section contrasts the proposed EDA against two known Evolutionary Algorithms based on multivariate densities: CMA-ES [3] and xNES [2]. Each algorithm runs in 10 benchmark problems, see Table 1. In order to make a fair comparison, the code was downloaded from authors homepage and 50 runs were performed. Also, the initial center of densities was chosen randomly in the search domain with an initial variance according to the domain (1/3 of this). The three algorithms only have two stopping conditions: maximum number of evaluations of function is reached ($10^4 \times d$), or target error smaller than 10^{-8},

$t \leftarrow 0$, $\beta \leftarrow 0.5$, $N \leftarrow \lceil 4 * (1 + d^{0.7}) \rceil$, $M \leftarrow 2 * \lfloor \log(d) \rfloor + 1$, $r \leftarrow d + 1$
$Pob_t \leftarrow \mathcal{U}(Domain)$, compute $\mathcal{F}(\boldsymbol{x}^{(i)})$, find the $\boldsymbol{x}^{best}$ ▷ First population
$P_{best} \leftarrow$ Best $r + 2$ individuals from Pob_t ▷ Historical best population
while (Stop condition is not reached) **do**
  ∘ Gradient estimate $G(\boldsymbol{x}^{best}) = \widehat{\nabla \mathcal{F}}(\boldsymbol{x}^{best})$ with neighborhood $\{P_{best} \setminus \boldsymbol{x}^{best}\}$
  ∘ Normalized vector $\boldsymbol{x}_G^{best}$ by (9) or negative for minimization
  ∘ Empirical estimation of $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$. Initial covariance $\boldsymbol{\Sigma}_0 = diag(diag(\widehat{\boldsymbol{\Sigma}}))$
  ∘ Parameters $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ by definition 4. Parameters $\boldsymbol{\mu}^{new}$ and $\boldsymbol{\Sigma}^{new}$ by (15)
  ∘ $\mathcal{S} \leftarrow$ Simulate M individuals from $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}^{new}, \boldsymbol{\Sigma}^{new})$
  ∘ $\mathcal{S} \leftarrow Reinsertion(\mathcal{S})$ ▷ if-outside-domain
  ∘ Fitness values $\mathcal{F}(\mathcal{S})$
  ∘ $Pob_{t+1} \leftarrow$ Best individuals among $\{Pob_t, \mathcal{S}\}$
  ∘ Find the $\boldsymbol{x}_{t+1}^{best}$ of Pob_{t+1}
  if $\boldsymbol{x}_{t+1}^{best}$ has better fitness value than $\boldsymbol{x}^{best}$ **then**
    $\boldsymbol{x}^{best} \leftarrow \boldsymbol{x}_{t+1}^{best}$ and $\boldsymbol{x}_{t+1}^{best}$ replaces the worst individual in P_{best}
  end if
  ∘ $M_{sur} \leftarrow$ Number of survivors from $\mathcal{S}$ into Pob_{t+1}
  if $\frac{M_{sur}}{M} > 1/2$ **then**
    $\beta \leftarrow \beta + 0.05$; if $\beta > 1$ then $\beta = 1$ ▷ Exploration
  else
    $\beta \leftarrow \beta - 0.05$; if $\beta < 0$ then $\beta = 0$ ▷ Exploitation
  end if
  $t \leftarrow t + 1$
end while

Fig. 2. Pseudocode of the EDA led by a Gradient-driven Density (EDA-LGD)

Table 1. Benchmark problems [2] [5]. The minimum fitness value of all problems is 0, except for $\mathcal{F}_4$, $\mathcal{F}_6$ and $\mathcal{F}_{10}$ where $\mathcal{F}_4^* = 2$, $\mathcal{F}_6^* = -10$ and $\mathcal{F}_{10}^* = -0.1d$.

Name	Alias	Domain	Name	Alias	Domain
Sphere	$\mathcal{F}_1$	$x_i \in [-600, 300]$	Different Powers	$\mathcal{F}_2$	$x_i \in [-20, 10]$
Brown	$\mathcal{F}_3$	$x_i \in [-1, 4]$	Mishra 2	$\mathcal{F}_4$	$x_i \in [0, 1]$
Ellipsoid	$\mathcal{F}_5$	$x_i \in [-20, 10]$	Parabolic Ridge	$\mathcal{F}_6$	$x_i \in [-20, 10]$
Rosenbrock	$\mathcal{F}_7$	$x_i \in [-20, 10]$	Ackley	$\mathcal{F}_8$	$x_i \in [-20, 10]$
Griewangk	$\mathcal{F}_9$	$x_i \in [-600, 300]$	Negative Cosine Mixture	$\mathcal{F}_{10}$	$x_i \in [-1, 0.5]$

i.e. $(\mathcal{F} - \mathcal{F}^*) < 10^{-8}$. Figure 3 contrasts the error $\mathcal{F} - \mathcal{F}^*$ reached for each algorithm. Also, this Figure shows a comparison between two algorithms in the second and third columns. For each problem there are three measures: 1) the first row is the percentage of success rate, 2) the second row is the mean and standard deviation of reached fitness values, 3) the third row is the mean and standard deviation of needed evaluations of function. The mean values highlighted with boldface, i.e. the winner algorithm, are supported by a statistical test. The last column presents the results of two nonparametric bootstrap tests. Here, the hypotheses are based on the mean value μ. The hypotheses $(H_0 : \mu_1 \geq \mu_2, H_1 : \mu_1 < \mu_2)$ yields the p-value ρ_1 and $(H_0 : \mu_2 \geq \mu_1, H_1 : \mu_2 < \mu_1)$ produces

$\mathcal{F}$	EDA-LGD	CMA-ES	ρ_1 vs ρ_2
$\mathcal{F}_1$	**100.00**	**100.00**	
	8.7e-9±1.2e-9	**5.5e-9±1.2e-9**	1.0,**1e-4**
	4.9e+4±1.7e+4	**3.8e+3±1.4e+2**	1.0,**1e-4**
$\mathcal{F}_2$	**100.00**	**100.00**	
	8.0e-9±1.6e-9	9.4e-9±5.9e-10	**1e-4**,1.0
	5.5e+3±5.6e+2	9.0e+3±7.2e+2	**1e-4**,1.0
$\mathcal{F}_3$	**100.00**	**100.00**	
	8.6e-9±1.1e-9	**5.1e-9±1.3e-9**	1.0,**1e-4**
	5.1e+3±2.1e+2	**3.0e+3±1.4e+2**	1.0,**1e-4**
$\mathcal{F}_4$	**100.00**	4.00	
	9.4e-9±5.0e-10	8.9e-2±8.9e-2	**1e-4**,1.0
	2.9e+3±1.2e+2	1.9e+5±1.6e+4	**1e-4**,1.0
$\mathcal{F}_5$	**100.00**	**100.00**	
	8.6e-9±1.2e-9	**5.4e-9±1.2e-9**	1.0,**1e-4**
	9.8e+3±7.9e+2	1.8e+4±3.0e+2	**1e-4**,1.0
$\mathcal{F}_6$	**100.00**	**100.00**	
	9.0e-9±9.0e-10	**7.8e-9±1.0e-9**	1.0,**1e-4**
	7.6e+3±3.9e+2	9.3e+3±1.5e+3	**1e-4**,1.0
$\mathcal{F}_7$	28.00	**88.00**	
	5.5e-1±1.4e+0	4.7e-1±1.3e+0	0.6,0.3
	1.9e+5±8.7e+3	**4.2e+4±5.8e+4**	1.0,**1e-4**
$\mathcal{F}_8$	**82.00**	56.00	
	2.4e-1±5.6e-1	1.5e+0±2.2e+0	**3e-4**,1.0
	4.4e+4±7.3e+4	9.1e+4±9.7e+4	**4e-3**,0.9
$\mathcal{F}_9$	2.00	**76.00**	
	1.6e+0±2.5e+0	**2.5e-3±4.9e-3**	1.0,**1e-4**
	1.9e+5±2.3e+4	**5.1e+4±8.4e+4**	1.0,**1e-4**
$\mathcal{F}_{10}$	**12.00**	6.00	
	2.7e-1±1.9e-1	3.0e-1±1.5e-1	0.1,0.8
	1.7e+5±6.4e+4	1.8e+5±4.7e+4	0.2,0.8

(a)

$\mathcal{F}$	EDA-LGD	xNES	ρ_1 vs ρ_2
$\mathcal{F}_1$	**100.00**	**100.00**	
	8.7e-9±1.2e-9	8.9e-9±8.7e-10	0.2,0.7
	4.9e+4±1.7e+4	**2.8e+4±2.9e+2**	1.0,**1e-4**
$\mathcal{F}_2$	**100.00**	**100.00**	
	8.0e-9±1.6e-9	7.4e-9±2.0e-9	0.9,6e-2
	5.5e+3±5.6e+2	1.6e+4±8.4e+2	**1e-4**,1.0
$\mathcal{F}_3$	**100.00**	**100.00**	
	8.6e-9±1.1e-9	8.6e-9±1.1e-9	0.5,0.4
	5.1e+3±2.1e+2	2.4e+4±3.3e+2	**1e-4**,1.0
$\mathcal{F}_4$	**100.00**	94.00	
	9.4e-9±5.0e-10	2.4e+2±1.7e+3	7e-2,0.9
	2.9e+3±1.2e+2	8.3e+4±3.1e+4	**1e-4**,1.0
$\mathcal{F}_5$	**100.00**	38.00	
	8.6e-9±1.2e-9	1.7e-3±7.9e-3	5e-2,0.9
	9.8e+3±7.9e+2	1.5e+5±5.8e+4	**1e-4**,1.0
$\mathcal{F}_6$	**100.00**	96.00	
	9.0e-9±9.0e-10	9.3e-9±6.7e-10	**2e-2**,0.9
	7.6e+3±3.9e+2	5.2e+4±3.3e+4	**1e-4**,1.0
$\mathcal{F}_7$	28.00	**100.00**	
	5.5e-1±1.4e+0	**8.6e-9±1.2e-9**	0.9,**2e-3**
	1.9e+5±8.7e+3	**4.4e+4±2.3e+3**	1.0,**1e-4**
$\mathcal{F}_8$	82.00	**100.00**	
	2.4e-1±5.6e-1	**9.3e-9±5.5e-10**	0.9,**1e-3**
	4.4e+4±7.3e+4	4.3e+4±4.3e+2	0.5,0.4
$\mathcal{F}_9$	2.00	**96.00**	
	1.6e+0±2.5e+0	**3.9e-4±2.0e-3**	1.0,**2e-4**
	1.9e+5±2.3e+4	**3.2e+4±3.4e+4**	1.0,**1e-4**
$\mathcal{F}_{10}$	12.00	**68.00**	
	2.7e-1±1.9e-1	1.5e-1±5.0e-1	0.9,6e-2
	1.7e+5±6.4e+4	**8.7e+4±7.8e+4**	1.0,**1e-4**

(b)

Fig. 3. Percentage of success rate, reached fitness values and needed number of evaluations (mean and standard deviation) for each algorithm in dimension 20. The last column shows two nonparametric bootstrap tests. If ρ_1 is boldface the winner is EDA-LGD, if ρ_2 is boldface the winner is either CMA-ES or xNES, otherwise there is no winner.

the p-value ρ_2. So, if ρ_1 is boldface the winner is EDA-LGD, if ρ_2 is boldface the winner is either CMA-ES or xNES, otherwise there is no winner. The null hypothesis is rejected with significance level $\alpha = 0.05$ **Comments (CMAES):** The problems $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_5$ and $\mathcal{F}_6$ do not seem difficult for EDA-LGD nor CMA-ES, since both algorithms reach the perfect success rate. On the contrary, the rest of the problems have a more difficult landscape. According to the bootstrap test, there is statistical evidence to conclude that in 5 out of 10 problems the proposed EDA requires fewer function evaluations than the CMA-ES. **Comments (xNES):** According to the bootstrap test, there is statistical evidence to conclude that in 5 out of 10 problems the proposed EDA requieres fewer function evaluations than the xNES. Also, there appears to be a pattern related to the landscape. For instance, note xNES has better results for problems $\mathcal{F}_7 - \mathcal{F}_{10}$, but EDA-LGD has better results for problems $\mathcal{F}_2 - \mathcal{F}_6$. This kind of pattern must be further studied in future work.

6 Conclusion

This paper presents a new EDA based on the Gradient-driven densities (∇_dD). In order to build the proposed EDA (EDA-LGD) two main contributions were developed: the Expected Gradient Estimate (EGE) and the ∇_dD. Also, a technique has been proposed to compute a gradient estimate for any individual only by using the actual knowledge about the problem. Hence, the estimation of the gradient does not need extra evaluations of function. The ∇_dD are statistical models built by taking into account a gradient estimate. This new framework can create a density function for any individual. Consequently, any simulation from those densities has a random gradient component. Here, Gradient-driven densities based on the Multivariate Normal have been constructed. However, the developed framework allows for the assumption of other statistical models. The ideas discussed above motivated a new EDA: EDA-LGD. It is based on the Gradient-driven Independent Normal, the EGE and the hierarchical latent variable model. Moreover, it was tested in 10 benchmark problems; where the EDA-LGD shows competitive performance against CMA-ES and xNES. In summary, the EDA-LGD is an interesting approach because of the performance of the algorithm and its mathematical foundation. Since the ∇_dD will produce samples in a similar direction as the gradient estimation, this density can be regarded as a predictive model. Thus, the Gradient-Driven density allows for exploration of the search domain whilst the empirical density intends fast convergence (exploitation). Finally, notice that the main contributions developed here can be extended to other evolutionary algorithms.

References

1. Flaxman, A.D., Kalai, A.T., McMahan, H.B.: Online convex optimization in the bandit setting: Gradient descent without a gradient. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, pp. 385–394. Society for Industrial and Applied Mathematics, Philadelphia (2005)
2. Glasmachers, T., Schaul, T., Sun, Y., Wierstra, D., Schmidhuber, J.: Exponential natural evolution strategies. In: Genetic and Evolutionary Computation Conference (2010)
3. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a new evolutionary computation. STUDFUZZ, vol. 192, pp. 75–102. Springer, Heidelberg (2006)
4. Hazen, M., Gupta, M.R.: Gradient estimation in global optimization algorithms. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1841–1848 (2009)
5. Larrañaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer, Boston (2002)
6. Schnatter, F.S.: Finite mixture and Markov switching models. Springer, Heidelberg (2006)